# BACHELOR OF TECHNOLOGY

# INFORMATION TECHNOLOGY

## SYLLABI BOOK
## (2021 - 2025)

Department of Information Technology
Faculty of Technology
Dharmsinh Desai University
Nadiad – 387 001, Gujarat, India.
www.ddu.ac.in

# DHARMSINH DESAI UNIVERSITY
# FACULTY OF TECHNOLOGY
## COURSE STRUCTURE FOR B.TECH. (INFORMATION TECHNOLOGY) W.E.F. 2021-22

## B.Tech Semester –I

| Subject | Teaching Scheme | | | Total | Credit | Exam Scheme (Marks) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | L | T | P | | | Th. | Int. | TW | Prac. | Total |
| Mathematics – I | 3 | 1 | 0 | 4 | 4 | 60 | 40 | - | - | 100 |
| Basic Electrical Engineering | 3 | 1 | 2 | 6 | 5 | 60 | 40 | 50 | - | 150 |
| Programming for Problem Solving - I | 4 | 0 | 3 | 7 | 5.5 | 60 | 40 | 50 | - | 150 |
| Engineering Graphics & Design | 1 | 0 | 4 | 5 | 3 | - | - | 100 | - | 100 |
| Software Workshop | 0 | 0 | 2 | 2 | 1 | - | - | 50 | - | 50 |
| | 11 | 2 | 11 | 24 | 18.5 | 180 | 120 | 250 | 0 | 550 |

## B.Tech Semester –II

| Subject | Teaching Scheme | | | Total | Credit | Exam Scheme (Marks) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | L | T | P | | | Th. | Int. | TW | Prac. | Total |
| Mathematics – II | 3 | 1 | 0 | 4 | 4 | 60 | 40 | - | - | 100 |
| Programming for Problem Solving - II | 4 | 0 | 3 | 7 | 5.5 | 60 | 40 | 50 | - | 150 |
| Physics | 3 | 1 | 2 | 6 | 5 | 60 | 40 | 50 | - | 150 |
| Hardware Workshop | 0 | 0 | 4 | 4 | 2 | - | - | 100 | - | 100 |
| English | 2 | 0 | 2 | 4 | 3 | 40 | - | 50 | - | 90 |
| Environmental Studies | 2 | 0 | 0 | 2 | 0 | 40 | - | - | - | 40 |
| | 14 | 2 | 11 | 27 | 19.5 | 260 | 120 | 250 | 0 | 630 |

## B.Tech Semester –III

| Subject | Teaching Scheme | | | Total | Credit | Exam Scheme (Marks) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | L | T | P | | | Th. | Int. | TW | Prac. | Total |
| Probability Theory and Statistics | 3 | 1 | 0 | 4 | 4 | 60 | 40 | - | - | 100 |
| Communication Systems | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| Design of Digital Circuits | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| Effective Technical Communication | 3 | 0 | 0 | 3 | 3 | 50 | - | 50 | - | 100 |
| Object Oriented Programming using Java | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| Data Structures and Algorithms | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| | 22 | 1 | 8 | 31 | 27 | 350 | 200 | 150 | 100 | 800 |

## B.Tech Semester –IV

| Subject | Teaching Scheme | | | Total | Credit | Exam Scheme (Marks) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | L | T | P | | | Th. | Int. | TW | Prac. | Total |
| Universal Human Values – II | 3 | 0 | 0 | 3 | 3 | 60 | - | - | - | 60 |
| Discrete Mathematics | 3 | 1 | 0 | 4 | 4 | 60 | 40 | - | - | 100 |
| Computer and Communication Networks | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| Database Management System | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| Design & Analysis of Algorithm | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| Microprocessor Programming and Interfacing | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| | 22 | 1 | 8 | 31 | 27 | 360 | 200 | 100 | 100 | 760 |

## B.Tech Semester –V

| Subject | Teaching Scheme | | | Total | Credit | Exam Scheme (Marks) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | L | T | P | | | Th. | Int. | TW | Prac. | Total |
| Computer Organisation | 3 | 0 | 0 | 3 | 3 | 60 | - | - | - | 60 |
| Theory of Automata & Formal Language | 4 | 0 | 0 | 4 | 4 | 60 | 40 | - | - | 100 |
| App Development | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| Professional Elective Course - 1 | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| Professional Elective Course - 2 | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| Open Elective - 1 (E-Commerce And E-Security) | 3 | 0 | 0 | 3 | 3 | 60 | - | - | - | 60 |
| | 22 | 0 | 6 | 28 | 25 | 360 | 160 | 75 | 75 | 670 |

## B.Tech Semester –VI

| Subject | Teaching Scheme | | | Total | Credit | Exam Scheme (Marks) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | L | T | P | | | Th. | Int. | TW | Prac. | Total |
| Open Elective - 2 (Data Analysis & Information Extraction) | 3 | 0 | 0 | 3 | 3 | 60 | - | - | - | 60 |
| Language Translator | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| Applied Operating System | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| Data Analytics using Python | 0 | 0 | 2 | 2 | 1 | - | - | 25 | 25 | 50 |
| Professional Elective Course - 3 | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| Project – I | 0 | 0 | 2 | 2 | 1 | - | - | 25 | 25 | 50 |
| | 15 | 0 | 10 | 25 | 20 | 240 | 120 | 125 | 125 | 610 |

## B.Tech Semester –VII

| Subject | Teaching Scheme | | | Total | Credit | Exam Scheme (Marks) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | L | T | P | | | Th. | Int. | TW | Prac. | Total |
| Constitution of India | 1 | 0 | 0 | 1 | 0 | - | - | - | - | - |
| Professional Elective Course - 4 | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| DevOps | 0 | 0 | 2 | 2 | 1 | - | - | 25 | 25 | 50 |
| Open Elective - 3 (Artificial Intelligence) | 3 | 0 | 0 | 3 | 3 | 60 | - | - | - | 60 |
| Professional Elective Course - 5 | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| Professional Elective Course - 6 | 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |
| Project – II | 0 | 0 | 2 | 2 | 1 | - | - | 25 | 25 | 50 |
| | 16 | 0 | 10 | 26 | 20 | 240 | 120 | 125 | 125 | 610 |

## B.Tech Semester –VIII

| Subject | Teaching Scheme | | | Total | Credit | Exam Scheme (Marks) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | L | T | P | | | Th. | Int. | TW | Prac. | Total |
| Industrial Internship | 0 | 6 | 24 | 30 | 18 | 0 | 0 | 150 | 350 | 500 |
| | 0 | 6 | 24 | 30 | 18 | 0 | 0 | 150 | 350 | 500 |

## List of Professional Elective Courses (PEC)

| | | |
|---|---|---|
| 1 | Advanced Java Technology | Professional Elective Course - 1 |
| | Internet of Things | |
| 2 | Software Engineering | Professional Elective Course - 2 |
| | Advanced Algorithms | |
| 3 | Full Stack Development | Professional Elective Course - 3 |
| | Cloud Computing | |
| 4 | Machine Learning & Deep Learning | Professional Elective Course - 4 |
| | Digital Image Processing | |
| 5 | Distributed Computing | Professional Elective Course - 5 |
| | Advanced Operating System | |
| 6 | Web Technology | Professional Elective Course - 6 |
| | Speech & Natural Language Processing | |

## B. TECH. SEMESTER – I (IT)
## SUBJECT: MATHEMATICS – I

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 3 | 1 | - | 4 | 4 | 60 | 40 | - | - | 100 |

Reference Code:BSC102

### A. COURSE OBJECTIVES

The objective of this course is to familiarize the prospective engineers with techniques in calculus, matrices, vector spaces and multivariable calculus

### B. DETAILED SYLLABUS

**Unit** **Topic(s)**

**[1] CALCULUS**

Evolutes and involutes, Evaluation of definite and improper integrals; Beta and Gamma functions and their properties, Applications of definite integrals to evaluate surface areas and volumes of revolutions. Rolle's Theorem, Mean value theorems, Taylor's and Maclaurin's theorems with remainders; Indeterminate forms and L'Hospital's rule, Maxima and minima.

**[2] MATRICES**

Matrices, Vectors: addition and scalar multiplication, matrix multiplication; Rank of a matrix, Linear systems of equations, Determinants, Cramer's Rule, Inverse of a matrix, Gauss Elimination and Gauss Jordan method.

**[3] VECTOR SPACES**

Eigenvalues, Eigenvectors, Symmetric, Skew-symmetric, and Orthogonal Matrices, Linear Independence of vectors, Diagonalization.

**[4] MULTIVARIABLE CALCULUS (Differentiation)**

Limit, Continuity and Partial derivatives, Directional derivatives, Total derivative; Tangent plane and normal line; Maxima, minima and saddle points; Method of Lagrange multipliers; Vector Differential Calculus; Gradient, curl and divergence.

### C. RECOMMENDED TEXT/ REFERENCE BOOK


1. B. S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 40th Edition, 2007.
2. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002.
3. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
4. D. Poole, Linear Algebra: A Modern Introduction, 2nd Edition, Brooks/Cole, 2005

5. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
6. Ramana B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
7. N.P. Bali and Manish Goyal, A textbook of Engineering Mathematics, Laxi Publications, Reprint, 2010.
8. V. Krishnamurthy, V.P. Mainra and J.L. Arora, An introduction to Linear Algebra, Affiliated East–West press, Reprint 2005.

## B. TECH. SEMESTER – I (IT)

## SUBJECT: BASIC ELECTRICAL ENGINEERING

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 3 | 1 | 2 | 6 | 5 | 60 | 40 | 50* | - | 150 |

Reference Code ESC104

*TW Marks include Viva based on TW

## A. COURSE OBJECTIVES

The course imparts an in-depth understanding of the fundamental concepts with an objective to expose the students to the various types of electrical, electronic and magnetic circuits and their applications. This course is designed to provide knowledge of fundamentals and various laws in electromagnetic and magnetic circuits, and electrostatics.

## B. DETAILED SYLLABUS

**Unit** **Topic(s)**

**[1] DC CIRCUITS**

Electrical circuit elements (R, L, and C), the impact of temperature, voltage, and current sources, Kirchhoff current and voltage laws, analysis of simple circuits with dc excitation. Superposition, Thevenin and Norton Theorems; Time-domain analysis of first-order RL and RC circuits.

**[2] AC CIRCUITS**

Representation of sinusoidal waveforms, peak and RMS values, phasor representation, real power, reactive power, apparent power, power factor; Analysis of single-phase ac circuits consisting of R, L, C, RL, RC, RLC combinations (series and parallel), resonance; Three-phase balanced circuits, voltage and current relations in star and delta connections.

**[3] ELECTRO-MAGNETIC INDUCTION**

Introduction, Magnetic effect of electric current, Current carrying conductor in the magnetic field, Law of electromagnetic induction, Induced emf, Self-Inductance (L), Mutual Inductance (M), and Coupling coefficient between two magnetically coupled circuits (K), Inductances in series and parallel.

**[4] MAGNETIC CIRCUITS**

Introduction, Definition of Magnetic quantities, Magnetic circuit, Leakage flux, Fringing effect, Comparison between magnetic and electric circuits.

**[5] TRANSFORMERS**

Magnetic materials, BH characteristics, ideal and practical transformer, equivalent circuit, losses in transformers, regulation, and efficiency; Auto-transformer and three-phase transformer connections.

### [6] ELECTRICAL MACHINES

Generation of rotating magnetic fields, Construction and working of a three-phase induction motor, Significance of torque-slip characteristic; Loss components and efficiency, starting and speed control of induction motor; Single-phase induction motor. Construction, working, torque-speed characteristic and speed control of separately excited DC motor; Construction and working of synchronous generators, Construction, Principles, and working theory and Types of DC Motors & Generators, 1-Ph & 3-Ph Induction Motor, AC Generator.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Basic Electrical, Electronics, and Computer Engineering, R. Muthu Subramanian, S. Salvahanan, K. A. Muraleedharan, 2nd Edition, Tata McGraw Hill.
2. Electronics Principles, Albert Paul Malvino, 6th Edition, Tata McGraw Hill
3. Electrical Technology (Vol: II), B. L. Theraja, A. K. Theraja, 23rd Edition, R. Chand & Company
4. Basic Electrical Engineering, D.P. Kothari, I. J. Nagrath, 3rd Edition, Tata McGraw Hill
5. Introduction to VLSI Circuit & Systems, John P. Uyemura, 1st Edition, John Willey & Sons Inc.
6. Basic Electrical Engineering, D.C. Kulshreshtha, 1st Edition, Tata McGraw Hill
7. Electrical and Electronics Technology, E. Hughes, 10th Edition, Pearson
8. Electrical Engineering Fundamentals, V.D. Toro, 2nd Edition, Prentice Hall India
9. Fundamentals of Electrical Engineering, L.S. Bobrow, , Oxford University Press

## B. TECH. SEMESTER – I (IT)

## SUBJECT: PROGRAMMING FOR PROBLEM SOLVING - I

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | -- | 3 | 7 | 5.5 | 60 | 40 | 50* | -- | 150 |

Reference Code ECS201

*TW Marks include Viva based on TW

## A. COURSE OBJECTIVES

- To teach fundamental programming concepts such as tokens,data types,storage class.
- To explain how to use decision making, branching, looping concepts for given problems.
- To impart knowledge of modular programming using functions.
- To discuss concepts of derived and user-defined data types using arrays,pointers,structures and unions.
- To demonstrate skills to write,debug,and execute c programs through laboratory practices.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] OVERVIEW OF C

Basic structure of C program, Compiling and running C program.

### [2] CONSTANTS, VARIABLES AND DATA TYPES

Types of constants, Basic data types, Identifier, Variable, Enum, Symbolic constant, Typedef, Keywords, Overflow and Underflow.

### [3] OPERATORS AND EXPRESSIONS

Arithmetic, relational, logical, Assignment, bitwise, and sizeof operator, Operator precedence and associativity, Expression evaluation.

### [4] MANAGING INPUT AND OUTPUT OPERATIONS

getchar and putchar functions, formatted I/O using printf and scanf.

### [5] DECISION MAKING AND BRANCHING

if and if...else statement, Nested and ladder if...else , Conditional operator, switch statement, goto statement with warning.

### [6] DECISION MAKING AND LOOPING

while, do...while, for loops, nested loops, break and continue statements.

### [7] ARRAYS AND STRINGS

Introduction to arrays, Declaration, initialization and access of one-dimensional and two-dimensional arrays, Introduction to multi-dimensional and variable length arrays, Declaration and initialization of strings, Printing and scanning strings to/from standard I/O, String handling functions, list of strings.

### [8] USER-DEFINED FUNCTIONS

Function prototype and function declaration, function definition, Function call, actual and formal parameters/arguments, Return type and return statement, Nested function call, recursion, Scope, visibility, and lifetime of variables.

### [9] STRUCTURES AND UNIONS

Defining structure, declaring and initializing structure variables, typedef, Accessing structure members, Copying and comparing structure variables, Nested structures, arrays and structures, Structures and functions, unions.

### [10] POINTERS

Introduction, accessing address of a variable, Declaration and initialization of pointer variables, Accessing variable using pointer, chain of pointers, Scale factor and pointer expressions, Pointers and arrays, Pointer to array Vs array of pointers, Passing arrays and strings to the function, Array of pointers, pointers and functions, pointers and structures, const pointer vs pointer to const.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Balagurusamy, Programming in ANSI C,8th Ed., Tata McGraw Hil,2019.
2. Byron Gottfried, Programming with C, 3rd Ed., McGraw Hill Education,2017.
3. Kernighan and Ritchie, The C Programming Language, 2nd Ed., PHI Learning,2015.
4. Peter Van Der Linden, Expert C Programming: Deep C Secrets,1st Ed.,Pearson Education, 1994.
5. Yashvant Kanetkar, Let Us C,12th Ed., BPB Publication,2015.
6. Ashok N. Kamthane, Programming in C, 2nd Ed., Pearson Education,2011.

## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember and comprehend C language syntax and programming paradigms. | Remember |
| CO2 | Be able to understand and describe C programming language syntax and concepts. | Understand |
| CO3 | Be able to use and apply C Programming concepts to solve algorithmic and logical problems. | Apply |
| CO4 | Be able to analyse the given problem and to formulate appropriate C language solution based on definitive language concept(s). | Analyze |
| CO5 | Be able to weigh a flowchart or a diagram for a given problem and evaluate given C programs using decision making, branching, looping, user defined function, array, structure, pointers, etc. | Evaluate |
| CO6 | Be able to design and construct a solution for given problem statements or application requirements by assembling modular solutions. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 2 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 2 | 0 | 1 | 3 | 2 | 3 |
| CO2 | 3 | 2 | 1 | 1 | 1 | 0 | 2 | 0 | 1 | 1 | 0 | 1 | 3 | 2 | 3 |
| CO3 | 2 | 3 | 1 | 1 | 1 | 2 | 1 | 0 | 2 | 1 | 0 | 1 | 3 | 2 | 3 |
| CO4 | 2 | 3 | 3 | 2 | 3 | 2 | 2 | 0 | 1 | 2 | 0 | 1 | 3 | 2 | 3 |
| CO5 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 1 | 2 | 0 | 1 | 3 | 2 | 3 |
| CO6 | 2 | 1 | 2 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 3 | 2 | 3 |
| Avg. | 2.17 | 2 | 1.67 | 1.33 | 1.5 | 1 | 1.5 | 0 | 1.17 | 1.5 | 0 | 1 | 3 | 2 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Overview of C, gcc compiler and basic program of C.
2. Implement the programs using Operators and Expressions.
3. Implement the programs using Decision making and Branching.
4. Implement the programs using Decision making and Looping.
5. Implement the programs using Arrays.
6. Implement the programs using Character Array.
7. Implement the programs using Functions.
8. Implement programs using Recursive function & Arrays as arguments to function.
9. Implement the programs using Structures.
10. Implement the programs using Union.
11. Implement the programs using Pointers.

## B. TECH. SEMESTER – I (IT)

## SUBJECT: ENGINEERING GRAPHICS & DESIGN

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 1 | - | 4 | 5 | 3 | - | - | 100* | - | 100 |

Reference Code ESC106

*TW Marks includes Viva based on TW

## A. COURSE OBJECTIVES

The objectives of this course are:

- To Understand the drawing importance in Engineering.
- To Describe the 3-Dimensional object in a different 2-Dimensional view.
- To Develop skills in Reading and Interpretation of Engineering Drawings.
- To enhance drawing skills through hands-on training in a CAD lab using engineering software.

## B. DETAILED SYLLABUS

| Unit | Topic(s) |
|---|---|
| [1] | **INTRODUCTION TO ENGINEERING DRAWING**<br>Principles of Engineering Graphics and their significance, usage of Drawing instruments, lettering, Conic sections including the Rectangular Hyperbola (General method only); Cycloid, Epicycloids, Hypocycloid and Involutes; Scales –Plain, Diagonal and Venire Scales; |
| [2] | **ORTHOGRAPHIC PROJECTIONS**<br>Principles of Orthographic Projections-Conventions -Projections of Points and lines inclined to both planes; Projections of planes inclined Planes-Auxiliary Planes; |
| [3] | **PROJECTIONS OF REGULAR SOLIDS**<br>Planes-Auxiliary Views; Draw simple annotation, dimensioning and scale. Floor plans that include: windows, doors, and fixtures such as WC, bath, sink, shower, etc. |
| [4] | **SECTIONS AND SECTIONAL VIEWS OF RIGHT ANGULAR SOLIDS**<br>Prism, Cylinder, Pyramid, Cone – Auxiliary Views; Development of surfaces of Right Regular Solids - Prism, Pyramid, Cylinder and Cone; Draw the sectional orthographic views of geometrical solid s, objects from industry and dwellings (foundation to slab only) |
| [5] | **ISOMETRIC PROJECTIONS**<br>Principles of Isometric projection –Isometric Scale, Isometric Views, Conventions; Isometric Views of lines, Planes, Simple and compound Solids; Conversion of Isometric Views to Orthographic Views and Vice versa, Conventions; |

**[6] OVERVIEW OF COMPUTER GRAPHICS**

Listing the computer technologies that impact on graphical communication, Demonstrating knowledge of the theory of CAD software

**[7] CUSTOMIZATION AND CAD DRAWING**

Set up of the drawing page and the printer, including scale settings, Setting up of units and drawing limits; ISO and ANSI standards for coordinate dimensioning and tolerance; Orthographic constraints, Snap to objects manually and automatically; Producing drawings by using various coordinate input entry methods to draw straight lines, Applying various ways of drawing circles;

**[8] ANNOTATIONS, LAYERING, AND OTHER FUNCTIONS**

Applying dimensions to objects, applying annotations to drawings; Setting up and use of Layers, layers to create drawings, Create, edit and use customized layers; Changing line lengths through modifying existing lines (extend/lengthen); Printing documents to paper using the print command; orthographic projection techniques; Drawing sectional views of composite right regular geometric solids and project the true shape of the sectioned surface; Drawing annotation, Computer-aided design (CAD) software modelling of parts and assemblies. Parametric and non -parametric solid, surface, and wireframe models. Part editing and two-dimensional documentation of models. Planar projection theory includes sketching of perspective, isometric, multi-view, auxiliary, and section views. Spatial visualization exercises. Dimensioning guidelines, tolerance techniques; dimensioning and scale multi-views of dwelling;

## C. RECOMMENDED TEXT/REFERENCE BOOKS


1. Bhatt N.D., Panchal V.M. & Ingle P.R., (2014), Engineering Drawing, Charotar Publishing House
2. Shah, M.B. & Rana B.C. (2008), Engineering Drawing and Computer Graphics, Pearson Education
3. Agrawal B. & Agrawal C. M. (2012), Engineering Graphics, TMH Publication
4. Narayana, K.L. & P Kannaiah (2008), Textbook on Engineering Drawing, Scitech Publishers) (Corresponding set of) CAD Software Theory and User Manuals

## B. TECH. SEMESTER – I (IT)

## SUBJECT: SOFTWARE WORKSHOP

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| - | - | 2 | 2 | 1 | - | - | 50* | - | 50 |

Reference Code ESC107

*TW Marks include Viva based on TW

## A. COURSE OBJECTIVES

- To explain the significance of Operating System and Linux Architecture
- To teach how to interact with Operating System with commands.
- To teach how to write, debug and test shell scripts
- To explain how to use utilities of the Unix operating system using the command line interface by various Unix commands.
- To teach how to perform file operation and change file permission in Linux OS.
- To demonstrate how to generate documentation using Latex.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] OPERATING SYSTEM

Introduction to Operating System and Linux Architecture

### [2] SOFTWARE

Installation of open source/freeware software using package manager for programming/simulation.

### [3] SHELL COMMANDS

Linux usage, commands & shell scripting. Command structure and general-purpose utility

### [4] FILE HANDLING

Basic of file handling. The file system, Handling ordinary files, File attributes and permission, file system details

### [5] SHELL SCRIPTING

Basic Shell commands, Looping and Branching and Various program using Shell Scripting

### [6] SHELL UTILITIES

Find command and shell, simple filters, advance filters.

### [7] EDITORS

VI editor for basic text editing, LATEX for scientific documents and report writing

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Sumitabha Das, Unix: Concepts and Applications, 4th Edition, Tata McGraw Hill, 2010
2. Firuza Karmali Aibara, A Short Introduction to Latex: A Book for Beginners, First Edition, Independent Publisher, 2019


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember and use shell scripting, Unix commands and Latex commands. | Remember |
| CO2 | Be able to understand and use significance of Operating System and its component, Linux architecture, and scripting language features. | Understand |
| CO3 | Be able to apply different commands and their options to use Unix utilities, to perform software installation using package manager and to perform operation related to file and directories. | Apply |
| CO4 | Be able to analyse the given problem, choose and combine appropriate Unix commands to get desire output and be able to write, debug and test shell script in Vi editor. | Analyze |
| CO5 | Be able to evaluate one or more options and Linux commands are available to solve the given problem or program requirements for shell script and be able to clearly decide a choice under the given context. | Evaluate |
| CO6 | Be able to create and modify professional documents and report having equation, table, images, lists and index in Latex. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 |
| **CO2** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 |
| **CO3** | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 3 | 2 | 2 |
| **CO4** | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 3 | 2 | 2 |
| **CO5** | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 3 | 2 | 2 |
| **CO6** | 0 | 1 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | 2 | 2 | 2 |
| **Avg.** | 0.17 | 0.67 | 0.67 | 0.5 | 0.5 | 0 | 0 | 0 | 0 | 0.83 | 0 | 0.67 | 2.5 | 2 | 2 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Overview of Linux Operating System and work with terminal.
2. Installation of open source/freeware software using package manager for programming.
3. Command Usage and General-Purpose Utilities of Linux OS.
4. File system and Handling ordinary files & directories.
5. Basic File Attributes and more File Attributes, chmod, chown, chgrp.
6. Working with the vi Editor.
7. Working with Shell script for basic problems.
8. Shell script with while loop.
9. Shell script with for loop and Branching.
10. Simple filters and Advance filters on Linux Terminal.
11. Study of LATEX for scientific documents and report writing.

## B. TECH. SEMESTER – II (IT)

## SUBJECT: MATHEMATICS-II

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 3 | 1 | - | 4 | 4 | 60 | 50 | - | - | 100 |

Reference Code BSC301

## A. COURSE OBJECTIVES

The objective of this course is to familiarize the prospective engineers with techniques in Differential Equations, and numerical methods and Laplace transform.

## B. DETAILED SYLLABUS

| Unit | Topic(s) |
|---|---|
| [1] | **FIRST ORDER ORDINARY DIFFERENTIAL EQUATIONS AND INTRODUCTION TO HIGHER ORDER DIFFERENTIAL EQUATIONS**<br>Exact, linear and Bernoulli's equations; Euler's equations; Equations not of first degree: equations solvable for p, equations solvable for y, equations solvable for x and Clairaut's type, second-order linear differential equations with variable coefficients. Method of variation of parameters, Cauchy-Euler equation. |
| [2] | **NUMERICAL METHODS**<br>Ordinary differential equations: Taylor's series, Euler and modified Euler's methods, Runge- Kutta method of fourth-order for solving first-order equations; Solution of algebraic and transcendental equations: Newton Raphson's Method, Numerical integration: Trapezoidal rule and Simpson's 1/3rd and 3/8 rules. |
| [3] | **MULTIVARIABLE CALCULUS (INTEGRATION)**<br>Multiple Integration: Double integrals (Cartesian), Change of the order of integration in double integrals, Change of variables (Cartesian to polar); Applications: areas and volumes; Triple integrals (Cartesian), Scalar line integrals, Vector line integrals, Scalar surface integrals, Vector surface integrals, theorems of Green, Gauss and Stoke's. |
| [4] | **LAPLACE TRANSFORM**<br>Laplace Transform, Properties of Laplace Transform, Laplace transform of periodic functions, Finding inverse Laplace transform by different methods, Convolution theorem; Evaluation of integrals by Laplace transform, Solving ODE by Laplace Transform method. |

## C. RECOMMENDED TEXT/REFERENCE BOOKS


1. B. S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 40th Edition, 2007.
2. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002.
3. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
4. W. E. Boyce and R. C. Di Prima, Elementary Differential Equations and Boundary Value Problems, 9th Edn., Wiley India, 2009.
5. S. L. Ross, Differential Equations, 3rd Ed., Wiley India, 1984.
6. A. Coddington, An Introduction to Ordinary Differential Equations, Prentice Hall India, 1995.
7. E. L. Ince, Ordinary Differential Equations, Dover Publications, 1958.
8. J. W. Brown and R. V. Churchill, Complex Variables and Applications, 7th Ed., Mc- Graw Hill, 2004.
9. N.P. Bali and Manish Goyal, A textbook of Engineering Mathematics, Laxmi Publications, Reprint, 2008

## B. TECH. SEMESTER – II (IT)

## SUBJECT: PROGRAMMING FOR PROBLEM SOLVING - II

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | – | 3 | 7 | 5.5 | 60 | 40 | 50 | – | 150 |

Reference Code ECS201

## A. COURSE OBJECTIVES

- To provide a detailed overview of fundamental concepts of object-oriented programming like abstraction, inheritance, polymorphism etc.
- To teach programmatic implementation of these concepts using c++ language.
- To discuss and demonstrate advantages of object-oriented programming over procedural programming.
- To discuss correlation of these concepts with subjects like software engineering and object-oriented design & analysis.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] BASICS OF C++

Overview, Program structure, keywords, identifiers, constants, data types, Symbolic constants, declaration of variables, operators, namespaces, control structures, Dynamic memory – C style - malloc, calloc, realloc and free Vs C++ style, New and delete keywords, reference and pointer

### [2] FUNCTIONS IN C++

Main function (variations in signature), function prototype, inline functions, Call and return by reference, default parameters, function overloading.

### [3] INTRODUCTION TO OBJECT ORIENTED PROGRAMMING

Procedural Vs Object Oriented Programming, Principles of OOP, Benefits and applications of OOP

### [4] CLASSES AND OBJECTS – ENCAPSULATION AND ABSTRACTION

Introduction, private and public members, defining member functions, static members, Objects as function arguments and return type, Friend functions, const member functions, Constructors and their types, Destructor, Operator overloading, type conversion.

### [5] INTRODUCTION TO C++ STRING CLASS

### [6] INHERITANCE

Introduction, types of inheritance – single, multiple, multilevel, hierarchical, and hybrid inheritance, Protected members, overriding, virtual base class

### [7] POLYMORPHISM

Introduction, Pointers and Objects, this pointer, pointer to derived classes, virtual and pure virtual functions, dynamic binding

### [8] INPUT/OUTPUT

Introduction to streams, standard I/O stream objects, Stream classes, unformatted and formatted I/O, manipulators

### [9] EXCEPTION HANDLING

Basics of exception handling, Try-catch-throw, rethrowing exceptions, user defined exceptions

### [10] TEMPLATES

Basics of class templates and function templates.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. E Balagurusamy, Object-Oriented programming with C++, Seventh Edition, TMH publication, 2008.
2. Bjarne Stroustrup, The C++ Programming Language, Fourth Edition, Addison -Wesley publication, 1997.
3. Robert Lafore, Object-Oriented Programming in C++, Fourth Edition, SAMS publication, 1997.
4. Andrew Koenig and Barbara E Moo, Accelerated C++: Practical Programming by Example, First Edition, Addison-Wesley publication, 2000.
5. Steven Holzner, C++ Black Book, First edition, Paraglyph Press, 2001.
6. Herbert Schildt, C++: The Complete Reference, Fourth Edition, McGraw Hill Education, 2002.

## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember and comprehend C++ language syntax and programming paradigms. | Remember |
| CO2 | Be able to use and understand language syntax and concepts for C++ Programming along with templates for class and function. | Understand |
| CO3 | Be able to use and apply Object Oriented Programming(OOP) concepts to solve algorithmic and logical problems. | Apply |
| CO4 | Be able to analyse the given problem and to formulate appropriate C++ language solutions based on OOP Principle(s). | Analyze |
| CO5 | Be able to weigh and evaluate C++ solution(s) for given problem(s) using various OOP concepts such as Encapsulation, Abstraction, Inheritance, Polymorphism, Exception Handling, etc. | Evaluate |
| CO6 | Be able to design and construct a C++ solution for given problem statements or application requirements by assembling modular solutions. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 2 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 2 | 0 | 1 | 3 | 2 | 3 |
| CO2 | 3 | 2 | 1 | 1 | 1 | 0 | 2 | 0 | 1 | 1 | 0 | 1 | 3 | 2 | 3 |
| CO3 | 2 | 3 | 1 | 1 | 1 | 2 | 1 | 0 | 2 | 1 | 0 | 1 | 3 | 2 | 3 |
| CO4 | 2 | 3 | 3 | 2 | 3 | 2 | 2 | 0 | 1 | 2 | 0 | 1 | 3 | 2 | 3 |
| CO5 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 1 | 2 | 0 | 1 | 3 | 2 | 3 |
| CO6 | 2 | 1 | 2 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 3 | 2 | 3 |
| Avg. | 2.17 | 2 | 1.67 | 1.33 | 1.5 | 1 | 1.5 | 0 | 1.17 | 1.5 | 0 | 1 | 3 | 2 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Overview of C++ and basic program of c++.
2. Implement the programs using concept of functions in c++.
3. Implement the programs using concept of basic class and objects.
4. Implement the programs using concept of constructors and destructors.
5. Implement the programs using concept of operator overloading and string classes.
6. Implement the programs using concept of inheritance.
7. Implement the programs using concept of polymorphism.
8. Implement the programs using concept of I/O operations.
9. Implement the programs using concept of exception handling.
10. Implement the programs using concept of templets.
11. Overview of File I/O concepts and program(s) using file stream classes.

## B. TECH. SEMESTER – II (IT)

## SUBJECT: PHYSICS

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 3 | 1 | 2 | 6 | 5 | 60 | 40 | 50* | - | 150 |

Reference Code BSC101

*TW Marks include Viva based on TW

## A. COURSE OBJECTIVES

The course provides an in-depth understanding of the concepts associated with Semiconductor, Optoelectronics, Communication, Oscillators, and Basic Switching devices. It also serves the basic design ideas around rectification and amplification. The course focuses on modulation techniques and their components. The overall aspects of basic physics application in electronics with a practical approach are covered in this subject. This course also includes analog modulation & demodulation techniques (AM, FM, and PM) and digital modulation (ASK, FSK and PSK).

## B. DETAILED SYLLABUS

**Unit** **Topic(s)**

### [1] SEMICONDUCTORS

Intrinsic and extrinsic semiconductors, Carrier generation and recombination, Carrier transport: diffusion and drift, p-n junction, Metal-semiconductor junction (Ohmic and Schottky), Semiconductor materials of interest for optoelectronic & other devices.

### [2] DIODE

Introduction to P-N junction Diode and V-I characteristics, Half wave and Full-wave rectifiers, capacitor filter; Zener diode and its characteristics, Zener diode as a voltage regulator, Special purpose diodes.

### [3] LIGHT-SEMICONDUCTOR INTERACTION

Radiative transitions and optical absorption, LED and LASER, Photo detectors.

### [4] ACTIVE COMPONENTS AND APPLICATIONS

BJT: Structure and input-output characteristics of a BJT, The Unbiased Transistor, Transistor Currents, Biased Transistor, a single stage voltage divider biasing, Emitter Bias, The CE Connections, The Base Curve, Collector curve, Transistor approximation Variation in current Gain, The Load Line, The Operating point, Recognizing Saturation, BJT as a switch & Amplifiers, LED Drivers.

### [5] OSCILLATORS

General form of the oscillator, Sinusoidal oscillator, phase shift oscillator, and Crystal Oscillator.

### [6] MOSFET

MOS physics and mode of operations, nFET current-voltage relationship, MOS pass characteristics, and CMOS inverter, Dynamic RAM (DRAM) 1T bit-cell.

### [7] FIBER OPTICS

Fiber Optics and Optoelectronics, Historical Developments, A Fiber‑Optic Communication System, Advantages of Fiber‑Optic Systems, Ray Propagation in Optical Fibers, Fundamental Laws of Optics, Ray Propagation in Step‑Index Fibers, Ray Propagation in Graded‑Index Fibers.

### [8] COMMUNICATION SYSTEMS

Communication system components, Analog modulation- AM, FM, PM; Digital modulation- ASK, FSK, PSK.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Electronics Principles, Albert Paul Malvino, 6th Edition, Tata McGraw Hill
2. David Griffiths, Introduction to Electrodynamics
3. S. M. Sze, Semiconductor Devices: Physics and Technology, Wiley (2008).
4. R.P Khare, Fiber Optics and Optoelectronics, Oxford University Press
5. Sanjay Sharma, Communication Systems: Analog and Digital
6. Halliday and Resnick, Physics
7. W. Saslow, Electricity, magnetism and light
8. Singh, Semiconductor Optoelectronics: Physics and Technology, McGraw-Hill Inc. (1995).
9. B. E. A. Saleh and M. C. Teich, Fundamentals of Photonics, John Wiley & Sons, Inc., (2007).
10. Yariv and P. Yeh, Photonics: Optical Electronics in Modern Communications, Oxford University Press, New York (2007).
11. P. Bhattacharya, Semiconductor Optoelectronic Devices, Prentice Hall of India (1997)
12. Behrouz A. Forouzan, Data communication, and Networking.
13. B. P lathi, Modern Digital and Analog Communication Systems, 3rd edition.

## B. TECH. SEMESTER – II (IT)

## SUBJECT: HARDWARE WORKSHOP

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| - | - | 4 | 4 | 2 | - | - | 100 | - | 100 |

Reference Code ESC201

## A. COURSE OBJECTIVES

- To teach the use of different electronic components and computer hardware.
- To explain the use of different peripheral devices, architecture of different IoT boards like Arduino Uno, ESP32 and Raspberry Pi.
- To impart knowledge of Internet & Internet Protocols.
- To demonstrate website development using HTML and CSS, website deployment and IoT board coding using Arduino IDE.
- To demonstrate Circuit design, IoT boards configuration with different components like LED, Temperature and Humidity sensors, Motion sensors, etc.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] ELECTRONIC COMPONENTS

Study of Digital Multi-meter, Power Supply, Function Generator, Cathode Ray Oscilloscope, Digital Oscilloscope, Study the Measurement of Phase Difference in single phase circuit, Study of Various Electrical and Electronics component like LED, LDR, Photo-diode, MOSFET, MCB and Relay.

### [2] COMPUTER HARDWARE

Introduction to a personal computer and its basic peripherals, installation of Operating System, Software and the required device drivers. Students are suggested to perform similar tasks on the laptop scenario wherever possible.

### [3] PERIPHERALS

Programming of Computer Ports & Interfacing of Electronic Components, Cables and Connectors like RJ45, RS232 and CRO probe.

### [4] INTERNET

Introduction to Internet & World Wide Web modules, making a PC Internet ready. Introduction to Internet and TCP/IP, Ethernet Connection, WiFi connection, configure TCP/IP (IP, Gateway, DNS, and Proxy), and use of ping command. Information sharing and data transfer over Local Area Network and Internet

### [5] WEB INFRASTRUCTURE

Basic Components of Web Sites, Front end & back-end tools and technology. HTML & CSS, Developing, Configuring and deploying a website.

### [6] IOT BOARDS AND CIRCUIT SIMULATION

Introduction to IOT boards like Arduino, Raspberry Pie etc. Interfacing, Circuit designing and PCB designing.

### [7] MINI PROJECT

Student will develop a mini project related to the topics listed above

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Dr. Madhuri A Joshi, Electronic Components and Materials Principles, Third Edition, Shroff Publishers & Distributors PVT. LTD., 2004
2. Jyotika Deshmukh, A Textbook of Computer Hardware and Networking, D J Publications
3. Jennifer Robbins, Learning Web Design, 4th edition, O'Reilly Media, 2012


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember the use of various electronic component, concepts of circuit design architecture of various IoT boards and HTML, CSS. | Remember |
| CO2 | Be able to understand the significance and application of Electronic Component, Computer Hardware Peripherals, working of a Computer System and Internet. | Understand |
| CO3 | Be able to install operating system, software and device driver and also be able to troubleshoot related problems using system tools and be able to establish local area network. | Apply |
| CO4 | Be able to analyse the given problem and identify appropriate IoT components such as Arduino, Node MCU, Raspberry PI, and various sensors and other circuit components. | Analyze |
| CO5 | Be able to select the suitable IoT board and components for given problem solution. | Evaluate |
| CO6 | Be able to develop an IoT or web based project in a team using IoT hardware and Web Infrastructure also be able to prepare a project report and present the work. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 2 |
| **CO2** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 2 |
| **CO3** | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 2 |
| **CO4** | 1 | 2 | 1 | 2 | 3 | 0 | 1 | 0 | 2 | 1 | 0 | 1 | 3 | 2 | 2 |
| **CO5** | 1 | 2 | 1 | 2 | 3 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 3 | 2 | 3 |
| **CO6** | 1 | 1 | 1 | 2 | 3 | 0 | 0 | 0 | 3 | 1 | 0 | 1 | 3 | 2 | 3 |
| **Avg.** | 0.67 | 1 | 0.5 | 1 | 1.67 | 0 | 0.17 | 0 | 1 | 0.5 | 0 | 0.33 | 3 | 2 | 2.33 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Study and application of basic power source, measuring instruments and electronics components.
2. Study and application of function generator and CRO.
3. Understand the use of computer peripherals and hardware, installation of Operating System, device driver and other software.
4. Interfacing Electronic Components and understanding basics of Circuit Designing and PCB designing.
5. Understand the basics of Network topology and cable interface.
6. Introduction to Internet, Internet Protocols and World Wide Web modules
7. Establish a network for data transfer over Local Area Network and network printer.
8. Understand Basic Components of Web Sites, Front end & back-end tools and technology.
9. Develop a basic website using HTML, CSS and Wordpress.
10. Configure and deploy the website over the server.
11. Study and application of Internet of Things, IoT boards like Arduino, Raspberry Pie and configure them.
12. Interfacing and connecting multiple IoT components together.
13. Mobile phone security management and permission understanding.
14. Develop Mini Project using above mentioned topics.

## B. TECH. SEMESTER – II (IT)

## SUBJECT: ENGLISH

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 2 | - | 2 | 4 | 3 | 40 | - | 50* | - | 90 |

Reference Code HSMC201

*TW Marks include Viva based on TW

## A. COURSE OBJECTIVES

The objective of the course is to provide basic knowledge of the English language to students coming from different backgrounds. The course aims to teach English Grammar and Communications skills which will be useful to engineers.

## B. DETAILED SYLLABUS

**Unit** **Topic(s)**

**[1] VOCABULARY BUILDING**

The concept of Word Formation, Root words from foreign languages and their use in English, Acquaintance with prefixes and suffixes from foreign languages in English to form derivatives, Synonyms, antonyms, and standard abbreviations.

**[2] BASIC WRITING SKILLS**

Sentence Structures, use of phrases and clauses in sentences, Importance of proper punctuation, creating coherence, organizing principles of paragraphs in documents, Techniques for writing precisely.

**[3] IDENTIFYING COMMON ERRORS IN WRITING**

Subject-verb agreement, Noun-pronoun agreement, Misplaced modifiers, Articles, Prepositions, Redundancies, Clichés.

**[4] NATURE AND STYLE OF SENSIBLE WRITING**

Describing, Defining, Classifying, providing examples or evidence, Writing introduction and conclusion.

**[5] WRITING PRACTICES**

Comprehension, Précis Writing, Essay Writing

**[6] ORAL COMMUNICATION**

Listening Comprehension, Pronunciation, Intonation, Stress and Rhythm, Common, Everyday Situations: Conversations and Dialogues, Communication at Workplace, Interviews, Formal Presentations (This unit involves interactive practice sessions in Language Lab).

## C. RECOMMENDED TEXT/REFERENCE BOOK


1. Practical English Usage. Michael Swan. OUP. 1995.
2. Remedial English Grammar. F.T. Wood. Macmillan.2007
3. On Writing Well. William Zinsser. Harper Resource Book. 2001
4. Study Writing. Liz Hamp-Lyons and Ben Heasley. Cambridge University Press. 2006.
5. Communication Skills. Sanjay Kumar and PushpLata. Oxford University Press. 2011.
6. Exercises in Spoken English. Parts. I-III. CIEFL, Hyderabad. Oxford University Press

## B. TECH. SEMESTER – II (IT)

## SUBJECT: ENVIRONMENTAL STUDIES

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 2 | - | - | 2 | 0 | 40 | - | - | - | 40 |

Reference Code MC-II

### A. COURSE OBJECTIVES

The objective of this course is to bring awareness about sustainable development is a key to the future of mankind. Understanding, analyzing, and proposing solutions to the contemporary environmental issues and problems of pollution, population explosion, solid waste disposal, environmental degradation, economic productivity, global warming, ozone layer depletion, and loss of biodiversity.

### B. DETAILED SYLLABUS

**Unit** **Topic(s)**

**[1]** **THE MULTIDISCIPLINARY NATURE OF ENVIRONMENTAL STUDIES**

Definition, scope, and importance & need for public awareness.

**[2]** **NATURAL RESOURCES**

Renewable and non-renewable resource: Natural resources and associated problems; Forest resources: Use and over-exploitation, deforestation, case studies, Timber extraction, mining, dams, and their effects on forests and tribal people; Water resources: Use and over-utilization of surface and ground water, floods, drought, conflicts, over water, dams benefit and problems; Mineral resources: Use and exploitation, environmental effects of extracting and using mineral resources, case studies; Food resources: World food problems, changes caused by agriculture and overgrazing, effects of modern agriculture, fertilizer-pesticide problems, water logging, salinity, case studies; Energy resources: Growing energy needs, renewable and non-renewable energy sources, use of alternate energy sources, case studies; Land resources: Land as a resource, land degradation, man induced landslides, soil erosion and desertification; Role of an individual in conservation of natural resources; Equitable use of resources of sustainable lifestyles

**[3]** **ECOSYSTEMS**

Concept of an ecosystem, Structure and function of an ecosystem, producers, consumers and decomposers, Energy flow in the ecosystem, Ecological succession, Food chains, food webs and ecological pyramids; Introduction, types, characteristic features, structure and function of the following ecosystem: Forest ecosystem, Grassland ecosystem, Desert ecosystem and Aquatic ecosystem (ponds, streams, lakes, rivers, oceans, estuaries)

**[4]** **BIODIVERSITY AND ITS CONSERVATION**

Introduction definition: Genetic, species and ecosystem diversity; Bio-geographical classification of India; Value of biodiversity: Consumptive use, productive use, social, ethical, aesthetic and option values; Biodiversity at global, national and local levels. India as a mega-diversity nation, Hot-spots of biodiversity, threats to biodiversity, habitat loss, poaching of wildlife, man-wildlife conflicts, Endangered and endemic species of India; Conservation of biodiversity: In-situ and ex-situ conservation of biodiversity

**[5]** **ENVIRONMENTAL POLLUTION**

Definition, Causes, effects and control measures of Air pollution, Water pollution, Soil pollution, Marine pollution, Noise pollution, Thermal pollution, nuclear hazards; Solid waste management, causes, effects and control measures of urban and industrial wastes;

Role of an individual in prevention of pollution, Pollution case studies; Disaster management: floods, earthquake, cyclone, and landslides.

**[6] SOCIAL ISSUES AND THE ENVIRONMENT**

From unsustainable to sustainable development, urban problems related to energy Water conservation, rainwater harvesting, watershed management; Resettlement and rehabilitation of people: its problems and concerns, Case studies; Environmental ethics: Issues and possible solutions; Climate change: Global warming, acid rain, ozone layer depletion, nuclear accidents, and the holocaust. Case studies, Wasteland reclamation, Consumerism and waste products; Environment Protection Act: Air (Prevention and Control of Pollution) Act, Water (Prevention & Control of Pollution) Act, Wildlife Protection Act, Forest Conservation Act; Issues involved in the enforcement of environmental legislation Public awareness

**[7] HUMAN POPULATION AND THE ENVIRONMENT**

Population growth, variation among nations, population explosion, Family Welfare Program, Environment, and human health, human rights, Value education HIV/AIDS, Women and Child Welfare; Role of Information Technology in Environmental and human health Case studies.

**[8] FIELD WORK**

Visit to a local area to document environmental assets (river/forest/grassland/hill/mountain); Visit a local polluted site – Urban/Rural/Industrial/Agricultural; Study common plants, insects, birds. Study of simple ecosystems – pond, river, hill, slopes, etc.

## C. RECOMMENDED TEXT/ REFERENCE BOOK


1. Erach Bharucha Textbook of Environmental Studies; Second Edition, Universities Press: Hyderabad, 2013
2. Poonia, M. P.; Sharma, S. C. Environmental studies; Khanna Publishing House: New Delhi, 2017
3. Rajagopalan, R. Environmental Studies; Oxford University Press: India, 2015
4. Varandani, N. S. Basics of Environmental studies; Lambert Academic Publishing: Germany, 2013.
5. Basak, A. Environmental Studies; Dorling Kindersley: India, 2009.
6. Dhameja, S. K. Environmental studies; S. K. Kataria and Sons: New Delhi, 2007.
7. Rao, C. S. Environmental Pollution Control Engineering; Wiley publishers: New Delhi, 2006.
8. Brunner, R. C. Hazardous Waste Incineration; McGraw Hill: Michigan, 1989.
9. Clark, R. S. Marine Pollution; Clanderson Press Oxford: Bath, 2001.
10. Trivedy, R. K. Handbook of Environmental Laws, Acts, Guidelines, Compliances & standards; B. S. publications: Hyderabad, 2005.
11. Jadhav, H.; Bhosale, V. M. Environmental Protection and Laws; Himalaya Pub. House: Delhi, 1995.
12. Agarwal, K. C. Environmental Biology; Nidi Publ.: Bikaner, 2001.
13. Bharucha, E. The Biodiversity of India; Mapin Publishing: Ahmedabad, India, 2002.
14. Cunningham, W.P.; Cooper; Gorhani, T. H. E.; Hepworth, M.T., Environmental Encyclopedia;Jaico Publ. House: Mumbai, 2001.
15. De, A. K. Environmental Chemistry; Wiley Eastern: New Delhi, 2006.
16. Gleick, H. P. Water in crisis, Pacific Institute for Studies in Dev., Environment & Security; Stockholm Env. Institute Oxford Univ. Press: New York, 1993.
17. Hawkins, R.E., Encyclopedia of Indian Natural History; Bombay Natural History Society: Bombay, 1987.

18. Heywood, V. H.; Waston, R. T. Global Biodiversity Assessment; Cambridge Univ. Press: Cambridge, 1995.
19. Mckinney, M.L.; School, R.M. Environmental Science systems & Solutions; Web enhanced edition: USA, 1996.
20. Miller, T.G. Jr.; Spoolman, S. E. Environmental Science; Cengage learning: Wadsworth, 2014.
21. Odum, E.P. Fundamentals of Ecology; W.B. Saunders: USA, 1971.
22. Rao, M. N.; Datta, A.K. Waste Water treatment; Oxford & IBH Publ.: New Delhi, 1987.
23. Sharma, B. K., Environmental Chemistry; Goel Publ. House: Meerut, 2001.
24. Townsend, C., Harper, J.; Michael, B. Essentials of Ecology; Blackwell: Oxford, 2008.
25. Trivedi, R. K., Handbook of Environmental Laws, Rules Guidelines, Compliances and Standards, Vol I and II; B. S. Publications, Hyderabad, 2010.
26. Trivedi, R. K.; Goel, P. K. Introduction to air pollution; ABD Publishers: Jaipur, 2003.
27. Wanger, K. D., Environmental Management; W.B. Saunders Co. Philadelphia, USA, 1998.

## B. TECH. SEMESTER – III (IT)

## SUBJECT: PROBABILITY THEORY AND STATISTICS

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 3 | 1 | - | 4 | 4 | 60 | 40 | - | - | 100 |

Reference Code BSC3XX

## A. COURSE OBJECTIVES

The objectives of teaching this course are:

- To provide an understanding of the basic concepts of probability, conditional probability, and independent events.
- To focus on the random variable, mathematical expectation and different types of distributions, sampling theory, and estimation theory.
- To demonstrate and teach the design of statistical hypothesis about the real-world problem and conduct appropriate tests for drawing valid inferences about the population characteristics
- To explain the significance of hypothesis testing for any research work

## B. DETAILED SYLLABUS

**Unit** **Topic(s)**

**[1]** **BASIC PROBABILITY**

Probability spaces, conditional probability, independence; Discrete random variables, Independent random variables, the multinomial distribution, Poisson approximation to the binomial distribution, infinite sequences of Bernoulli trials, sums of independent random variables; Expectation of Discrete Random Variables, Moments, Variance of a sum, Correlation coefficient, Chebyshev's Inequality.

**[2]** **CONTINUOUS PROBABILITY DISTRIBUTIONS**

Continuous random variables and their properties, distribution functions and densities, normal, exponential, and gamma densities.

**[3]** **BIVARIATE DISTRIBUTIONS**

Bivariate distributions and their properties, distribution of sums and quotients, conditional densities, Bayes' rule.

**[4]** **BASIC STATISTICS**

Measures of Central tendency: Moments, skewness, and Kurtosis - Probability distributions: Binomial, Poisson, and Normal - evaluation of statistical parameters for these three distributions, Correlation, and regression – Rank correlation.

**[5]** **APPLIED STATISTICS**

Curve fitting by the method of least squares- fitting of straight lines, second-degree parabolas, and more general curves; Test of significance: Large sample test for a single proportion, a difference of proportions, a single mean, difference of means, and difference of standard deviations.

**[6]** **SMALL SAMPLES**

Test for a single mean, a difference of means and correlation coefficients, test for ratio of variances - Chi-square test for goodness of fit and independence of attributes.

## C. RECOMMENDED TEXT/REFERENCE BOOK


1. Introduction to Probability and Statistics for Engineers and Scientists, Sheldon M. Ross, 4th edition1.
2. Head First Statistics, Dawn Griffiths, O'Reilly
3. Introduction to Probability Theory, P. G. Hoel, S. C. Port, and C. J. Stone, Universal Book Stall, 2003 (Reprint).
4. A First Course in Probability, S. Ross, 6th Ed., Pearson Education India, 2002.
5. An Introduction to Probability Theory and its Applications, W. Feller, Vol. 1, 3rd Ed., Wiley, 1968.

## B. TECH. SEMESTER – III (IT)

## SUBJECT: COMMUNICATION SYSTEMS

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | - | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code ESC3XX

### A. COURSE OBJECTIVES

To impart an in-depth understanding of the major concepts, techniques, and performance criteria used in the analysis of various signal operations (time domain and frequency domain), this course covers the Fourier analysis of the signals, provides knowledge of various blocks that constitute an analog and digital communication system and their interrelation and correlates the concepts of Information Theory with reference to analog & digital communication systems. This course also includes the analog modulation & demodulation techniques (AM, FM, and PM) and digital modulation & demodulation techniques (PCM, DPCM, and DM).

### B. DETAILED SYLLABUS

**Unit** **Topic(s)**

**[1] WAVEFORM SPECTRA**

Introduction, Sinusoidal Waveform, General Periodic Waveforms Trigonometric Fourier Series for a Periodic Waveforms, Fourier Coefficients, Spectrum for the Trigonometric Fourier Series, Rectangular Waves, Sawtooth Waveform, Pulse Train, Some General Properties of Periodic Waveforms, Exponential Fourier Series, Approximate Formulas for the Fourier Coefficient, Energy Signals for Fourier Transform, Filtering of Signals, Power Signals, Bandwidth Requirements for Analog Information Signals.

**[2] DIGITAL LINE WAVEFORMS**

Symbols, Bits, Bits and Bauds, Functional notations for Pulses, Line codes and Waveforms, M-ary Encoding, Inter Symbol Interference.

**[3] AMPLITUDE MODULATION**

Introduction, Amplitude Modulation, Amplitude Modulated Transmitters, AM Receivers.

**[4] SINGLE SIDEBAND MODULATION**

Introduction, Single Sideband Principles, The Balanced Modulator SSB Generation, SSB Reception, Modified SSB Systems.

**[5] ANGLE MODULATION**

Introduction, Frequency Modulation, Phase Modulation, Equivalence between FM and PM, Angle Modulator Circuits, Angle Modulation Detectors.

**[6] PULSE MODULATION**

Pulse Amplitude Modulation, Pulse Code Modulation, Pulse Frequency Modulation, Pulse Time Modulation, Pulse Position Modulation, Pulse Width Modulation.

**[7] DIGITAL COMMUNICATION**

Synchronization, Asynchronous Transmission, Probability of Bit Error in Baseband Transmission, Matched Filters, Optimum Terminal Filters, Bit Timing Recovery, Eye Diagram, Digital Carrier System, Carrier Recovery Circuit, DPSK, Hard and Soft Decision, Error Control Coding.

**[8] INTRODUCTION TO INFORMATION THEORY**

Measure of Information, Source Encoding.

## C. RECOMMENDED TEXT/REFERENCE BOOK


1. Modern Digital and Analog Communication System, B. P. Lathi, 2nd Edition, Oxford Publication
2. Communication Systems, Simon Haykin, 3rd Edition, John Wiley & sons.
3. Electronic Communication System-Fundamental through Advance, Tomas W., 3rd Edition, Wisley.
4. Communication System Analog & Digital, R. P. Singh, Tata McGraw-Hill.

## B. TECH. SEMESTER – III (IT)

## SUBJECT: DESIGN OF DIGITAL CIRCUITS

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | - | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code IE302

## A. COURSE OBJECTIVES

- To teach fundamental knowledge of digital logic and its application which helps to design digital circuits.
- To familiarize about fundamental principles of digital design using combinational and sequential logic to analyze and design the digital circuits.
- To demonstrate skills to construct, analyze, verify, and troubleshoot digital circuits using appropriate techniques and test equipment.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

**[1] BINARY SYSTEMS**

Introduction to Digital Computers and Digital Systems, Binary Numbers, Number Base, Conversion, Octal and Hexadecimal Numbers, Complements, binary Codes, Binary Storage and Registers, Binary Logic, Integrated Circuits.

**[2] BOOLEAN ALGEBRA AND LOGIC GATES**

Basic Definitions, Axiomatic Definition of Boolean algebra, Basic Theorems and Properties of Boolean algebra, Boolean Functions, Canonical and Standard Forms, Other Logic Operations, Digital Logic Gates, IC Digital Logic Families.

**[3] SIMPLIFICATION OF BOOLEAN FUNCTIONS**

The Map Method, Two and Three Variable Maps, Four-Variable Map, Five and Six Variable Maps, Product of Sum simplification, NAND and NOR Implementations, Don't-Care Conditions, The Tabulation Method, Determination of Prime- Implicants, selection of Prime implicants.

**[4] COMBINATIONAL LOGIC**

Introduction, Design Procedure, Adders and Subtractors, Code Conversion, Analysis Procedure, Multilevel NAND Circuits, Multilevel NOR Circuits, Exclusive OR and Equivalence Functions.

**[5] COMBINATIONAL LOGIC WITH MSI AND LSI**

Introduction, Binary Parallel Adder, Decimal Adder, Magnitude Comparator, Decoders, Multiplexers, Read-Only Memory (ROM), Programmable Logic Array (PLA).

**[6] SEQUENTIAL LOGIC**

Introduction, Flip-Flops, Triggering of Flip-Flops, Analysis of Clocked Sequential Circuits, State Reduction and Assignment, Flip-Flop excitation Tables, Design Procedure, Design of Counters, Design with State Equations.

**[7] REGISTERS, COUNTERS AND THE MEMORY UNIT**

Introduction, Registers, Shift Registers, Ripple Counters, Synchronous Counters, Timing Sequences, The Memory Unit, Examples of Random-Access Memories.

**[8] DIGITAL INTEGRATED CIRCUITS**

Introduction, Bipolar Transistor Characteristics, RTL and DTL Circuits, Integrated-Injection Logic, Transistor-Transistor Logic, Emitter Coupled logic, Metal-Oxide Semiconductor, Complementary MOS.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. M. Morris Mano, Digital Logic and Computer Design, Second Edition, Prentice-Hall of India, 1999
2. A. Anand Kumar, Fundamentals of Digital Circuits, Eastern Economy Edition, Prentice-Hall of India, 2006
3. Jacob Millman & Arvin Grabel, Microelectronics, Second Edition McGraw Hill International Edition, 1999


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember and use the basic concepts of Digital system and circuits, the structure of various number systems, and the working and design process of different combinational and sequential circuits. | Remember |
| CO2 | Be able to understand and use various techniques for minimization and circuit designing. | Understand |
| CO3 | Be able to apply knowledge of mathematics to solve the given problem, Also be able to apply different minimization techniques to simplify the hardware requirements for designing the digital circuits. | Apply |
| CO4 | Be able to analyze the given problem and be able to choose appropriate digital circuit components for designing the solution. | Analyze |
| CO5 | Be able to evaluate basic requirements for a circuit design and propose a cost effective solution. Be able to evaluate a specific design solution for the circuit design among the all available solutions. | Evaluate |
| CO6 | Be able to design, implement, and test a Digital Circuit solution for a real life problem using tools and various types of digital circuit components, sequential or combinational. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 |
| **CO2** | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 |
| **CO3** | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 |
| **CO4** | 2 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 |
| **CO5** | 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 |
| **CO6** | 2 | 1 | 1 | 1 | 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 | 2 | 3 |
| **Avg.** | 2.17 | 1 | 0.5 | 0.33 | 0.5 | 0 | 0 | 0 | 0 | 0.17 | 0 | 0 | 3 | 2 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. (A) To verify basic logic gates.
   (B) To verify the universal gates NAND and NOR
2. To implement half adder and full adder circuits To implement circuit that converts binary to gray and gray to binary
3. To implement circuit that converts binary to gray and gray to binary
4. (A) To implement a 4-bit comparator.
   (B) Using 4-bit comparator implements 8-bit comparator
5. (A) To implement 3X8 decoder
   (B) Using 3X8 decoder implements 4X16 decoder
6. To implement 8X1 Multiplexer
7. Introduction to Verilog. Write a program to implement half adder and full adder in verilog.
8. To verify various flip-flops like D, T, JK
9. To implement 3-bit and 4-bit binary counters
10. To implement a BCD counter.
11. To write and read data in RAM using IC 6264.

## B. TECH. SEMESTER – III (IT)

## SUBJECT: EFFECTIVE TECHNICAL COMMUNICATION

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 3 | - | - | 3 | 3 | 50 | - | 50 | - | 100 |

Reference Code HSMC3XX

## A. COURSE OBJECTIVES

The objectives of teaching this course are:

- To discuss and explain technical writing and professional communication along with the importance of Ethics, etiquette, and Values.
- To prepare students for effective public speaking, group discussion, and interviews.
- To explain how to study and validate various information sources such as websites, business documents, and professional journals.
- To teach how to carry out self development and self assessment.
- To prepare students to produce effective technical documents.

## B. DETAILED SYLLABUS

**Unit** **Topic(s)**

**[1] INFORMATION DESIGN AND DEVELOPMENT**

Different kinds of technical documents, Information development life cycle, Organization structures, factors affecting information and document design, Strategies for organization, Information design and writing for print and for online media.

**[2] TECHNICAL WRITING, GRAMMAR, AND EDITING**

Technical writing process, forms of discourse, Writing drafts and revising Collaborative writing, creating indexes, technical writing style, and language; Basics of grammar, study of advanced grammar, editing strategies to achieve appropriate technical style; Introduction to advanced technical communication, Usability, Hunan factors, Managing technical communication projects, time estimation, Single sourcing, Localization

**[3] SELF-DEVELOPMENT AND SELF-ASSESSMENT**

Self-assessment, Awareness, Perception, and Attitudes, Values and belief, Personal goal setting, career planning, Self-esteem. Managing Time; Personal memory, Rapid reading, Taking notes; Complex problem solving; Creativity

**[4] COMMUNICATION AND TECHNICAL WRITING**

Public speaking, Group discussion, Oral; presentation, Interviews, Graphic presentation, Presentation aids, Personality Development. Writing reports, project proposals, brochures, newsletters, technical articles, manuals, official notes, business letters, memos, progress reports, minutes of meetings, and event report.

**[5] ETHICS**

Business ethics, Etiquettes in social and office settings, Email etiquettes, Telephone Etiquettes, Engineering ethics, Managing time, Role and responsibility of engineer, Work culture in jobs, Personal memory, Rapid reading, Taking notes, Complex problem solving, Creativity.

## C. RECOMMENDED TEXT/REFERENCE BOOK


1. David F. Beer and David McMurrey, Guide to writing as an Engineer, John Willey. New York, 2004
2. Diane Hacker, Pocket Style Manual, Bedford Publication, New York, 2003. (ISBN 0312406843)
3. Shiv Khera, You Can Win, Macmillan Books, New York, 2003.

## B. TECH. SEMESTER – III (IT)

## SUBJECT: OBJECT ORIENTED PROGRAMMING USING JAVA

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | - | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PCC-CS503

## A. COURSE OBJECTIVES

The objectives of teaching this course are:

- To teach fundamental knowledge of object-oriented programming principles including defining classes, polymorphism, inheritance, encapsulation, abstraction, and interface, and explain with examples how to represent the solution of a given problem in object oriented representation.
- To explain how to use library API such as String, Arrays, StringBuffer, StringTokenizer, and Math in framing solutions to problems.
- To impart knowledge of package, exception, and multithreading and show their usage in practical problems.
- To discuss concepts of making user interactive programs using GUI programming, Stream API, and Network Programming and demonstrate their use in solving problems.
- To explain JUnit testing concept and apply it to java programming applications by making different test cases to check the authenticity of programs.
- To demonstrate skills to write, debug, and execute java programs and enable them to create Java solutions for given problem statements.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] INTRODUCTION TO PROGRAMMING LANGUAGE – JAVA

Java programming: History of Java Primitive data types, variables, constants, scope and lifetime of variables, Comments; Operators, operator hierarchy, expressions; Type conversion and casting.

### [2] CONTROL STRUCTURE AND METHODS

Control flow statements and loops, Loops – for, while, do-while; Console input and output, formatting output; Constructors and methods, Overloading of methods and constructors, recursion, Parameter passing, static fields and methods, access control, this reference; Garbage collection.

### [3] OBJECT ORIENTED PROGRAMMING PRINCIPLES

OOP Concepts, Classes, and objects, Data abstraction, encapsulation, inheritance, Polymorphism. Procedural and object-oriented programming paradigm; Object-Oriented Programming Using Java, Inheritance: Inheritance types, super and

subclasses, member access rules, super keyword, preventing inheritance: final classes and methods, the object class and its methods; Polymorphism: dynamic binding, method overriding, abstract classes and methods; Interface: Interfaces vs Abstract classes, defining an interface, implement interfaces, accessing implementations through interface references, extending interface.

## [4] ARRAYS, STRING, AND PACKAGES

Arrays: 1,2, N-dimension array, jagged array, arrays of objects; String: Exploring concepts of String, StringBuffer, and StringTokenizer classes; Packages: Defining, creating, and accessing a package, understanding CLASSPATH, importing packages, Exploring "Java.Util".

## [5] EXCEPTION HANDLING

Exception Handling: Benefits of exception handling, the classification of exceptions; Exception hierarchy: Throwable, checked exceptions, and unchecked exceptions; Usage of try, catch, throw, throws, and finally; Re-throwing exceptions, exception specification, built in exceptions; Creating and using own/user-defined exception subclasses.

## [6] MULTITHREADING

Multithreading: Multithreading and Multitasking; Thread Life Cycle, thread states, creating threads: Thread Priorities, Thread Groups, Daemon Threads, interrupting threads, thread priorities; Synchronizing threads, inter-thread communication.

## [7] GETTING STARTED WITH GRAPHIC PROGRAMMING

The AWT class hierarchy, Containers: Frame, Dialog, Panel; Events: Event Sources, Event Classes, Event Listeners, Delegation Event Model; Handling Action, Mouse, Window, Keyboard Events; Adapter Classes

## [8] CREATING USER INTERFACE AND ADVANCED GRAPHICS

The AWT Class Hierarchy; User Interface Components: Labels, Button, Canvas, Scrollbars, Text Components, Check Box, Checkbox Groups, Choices, Lists; Panels: Scrollpane, Dialogs, Menubar, Graphics, Layout Manager; Layout Manager Types: Border, Grid, Flow, Card, Grid Bag, No layout, etc.

## [9] INPUT AND OUTPUT

Input/Output classes. File management using file class Streams: Byte streams, character stream; Text input/output, binary input/output; Random access file operations.

### [10] NETWORK PROGRAMMING

Networking concepts: Introduction to TCP and UDP protocol; Socket programming classes: Socket, ServerSocket, InetAddress, URL, URL Connection; Client-server and multi-threaded application.

### [11] JAVA UNIT TESTING – JUNIT

JUnit: Types of Testing, Test Driven Development, Assert class, Test cases.

## C. RECOMMENDED TEXT/REFERENCE BOOKS


1. Y. Daniel Liang, An Introduction to Java programming, 6th edition, Publisher: PHI, 2001.
2. Herbert Schildt, The Complete Reference Java, 5th edition, Publisher: Tata McGraw-Hill, 2006.


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember object oriented programming features and concepts, Java programming language concepts and API interfaces, classes, and methods | Remember |
| CO2 | Be able to understand and use various language concepts, API, and language features. | Understand |
| CO3 | Be able to apply the concepts, features, API, and JUnit testing of Java to solve the given problem on console based application or GUI based application. | Apply |
| CO4 | Be able to analyze the given problem or program requirements with reference to available concepts, features, and API of Java and solve the problem. | Analyze |
| CO5 | Be able to evaluate one or more options to solve the given problem or program requirements and be able to clearly decide a choice under the given context. | Evaluate |
| CO6 | Be able to design and create an application/solution and different test cases for given problem statements or application requirements for console based application and GUI based application. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 2 | 3 |
| CO2 | 3 | 2 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 3 | 2 | 3 |
| CO3 | 2 | 3 | 1 | 1 | 1 | 2 | 1 | 0 | 0 | 1 | 0 | 1 | 3 | 2 | 3 |
| CO4 | 2 | 3 | 3 | 2 | 3 | 2 | 2 | 0 | 0 | 2 | 0 | 1 | 3 | 2 | 3 |
| CO5 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 2 | 0 | 1 | 3 | 2 | 3 |
| CO6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 |
| Avg. | 1.83 | 1.83 | 1.17 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0.83 | 3 | 2 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Write the programs using the concept of nested for loops and recursion.
2. Write the programs using the concept of command line argument.
3. Write the programs using the concept of arrays and StringBuffer class.
4. Write the programs using the concept of Generic class, Inheritance, Interface and Package.
5. Write a program that uses the concept of unit testing using JUnit.
6. Write the program which creates the Frame and implements MouseListener.
7. Implementing a GUI based calculator application and drawing different figures on a Canvas.
8. Write an application to simulate traffic lights and a calculator using GridbagLayout.
9. Write a program that uses the concept of Exception Handling.
10. Write the programs that use the concept of Threads.
11. Write a program that uses the concept of File I/O. Write a program that uses the concept of socket programming.

## B. TECH. SEMESTER – III (IT)

## SUBJECT: DATA STRUCTURE AND ALGORITHMS

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | – | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PCC3XX

## A. COURSE OBJECTIVES

- To teach different data structures and their operations.
- To teach and demonstrate the selection of efficient data structure for improving the efficiency of the system.
- To explain concepts that are useful to students to understand subjects like Database Management System and Design and Analysis of Algorithms.
- To impart the knowledge of real-world applications of the data structures.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] BASIC CONCEPTS

System Life Cycle; Object-Oriented Design: Algorithm Decomposition versus OO Decomposition, Fundamental Definitions, and Concepts of OO programming; Data Abstraction and Encapsulations; Algorithm Specification: Introduction, Recursive Algorithms.

### [2] ARRAYS

Abstract Data Types and the C++ Class; The Array as an Abstract Data Type; The Polynomial Abstract Data: Polynomial Representation, Polynomial Addition, Polynomial Multiplication, Disadvantages of Representing Polynomials by Arrays.

### [3] STACK AND QUEUE

The Stack Abstract Data Type; The Queue Abstract Data Type; Evaluation of Expressions: Expressions, Postfix Notation, Infix to Postfix, Multiple Stacks, and Queues.

### [4] LINKED LISTS

Singly Linked Lists; Representing Lists in C++: Defining a List Node in C++, Designing a List in C++, Pointer Manipulation in C++, List Manipulation Operations, Linked List Operations, Circular Lists, Linked Stacks and Queues; Polynomials: Polynomial Representation, Adding Polynomials; Doubly Linked Lists; Generalized Lists: Representation of Generalized Lists, Recursive Algorithms for Lists, Reference Counts, Shared and Recursive Lists.

### [5] TREES

Introduction: Terminology, Representation of Trees; Binary Trees: The Abstract Data Type, Properties of Binary Trees, Binary Tree Representations; Binary Tree Traversal and Tree Iterators: Introduction, Inorder Traversal, Preorder Traversal, Postorder Traversal, Iterative Inorder Traversal, Level-Order Traversal; Additional Binary Tree Operations: Copying Binary Trees, Testing Equality, The Satisfiability Problem; Threaded Binary Trees: Threads, Inorder Traversal of a Threaded Binary Tree, Inserting a Node into a Threaded Binary Tree. Heaps: Definitions, Insertion, and Deletion Of Max Heaps. Binary Search Trees: Definition, Searching a Binary Search Tree, Insertion and Deletion and Joining into a Binary Search Tree, Height of a Binary Search Tree.

### [6] GRAPHS

The Graph Abstract Data Type: Introduction, Definitions, Graph Representations. Elementary Graph Operations: Depth First Search, Breadth-First Search, Connected Components, Spanning Trees, Biconnected Components, Shortest Paths and Transitive Closure: All-Pairs Shortest Paths.

### [7] SORTING

Insertion Sort. Quick Sort. Merge Sort: Merging, Iterative Merge Sort, Recursive Merge Sort. Heap Sort. List and Table Sorts. Summary of Internal Sorting.

### [8] HASHING

The Symbol Table Abstract Data Type, Static Hashing, Hash Tables, Hashing Functions, Overflow Handling.

### [9] ADVANCED SEARCH STRUCTURES

AVL Trees, 2-3 Trees, 2-3-4 Trees, Red-Black Trees, B-Trees, Splay Trees, Digital Search Trees, Tries.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Fundamentals of Data Structures using C++ by: Horowitz, Sahni, Galgotia Pub. 1998 ed
2. Data Structures& Algorithms, by: Aho, Ullman, Addision Wesley
3. An Introduction to Data Strucures with applications, by: Tremblay, Sorenson, McGraw Hill.
4. The art of Computer Programming Vol. I & III, by: Kunth, Addision Wesley.
5. Data Structures using C and C++, by: YedidyahLangsam, Tenenbaum

## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember and use different data structure concepts and algorithms. | Remember |
| CO2 | Be able to understand and use various data structures and algorithms. | Understand |
| CO3 | Be able to apply the concepts of different data structures to solve the given problem using an appropriate algorithm. | Apply |
| CO4 | Be able to analyze the given problem and be able to select appropriate data structure like array,stack,queue,linked list,tree,graph etc. to solve the problem to improve efficiency. | Analyze |
| CO5 | Be able to evaluate one or more options to solve the given problem or program requirements and be able to clearly decide a choice under the given context. | Evaluate |
| CO6 | Be able to design a solution for a given problem statement and to prepare an efficient algorithm using appropriate data structure. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 2 | 3 |
| CO2 | 1 | 2 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 3 | 2 | 3 |
| CO3 | 2 | 3 | 1 | 1 | 1 | 2 | 0 | 0 | 0 | 1 | 0 | 1 | 3 | 2 | 3 |
| CO4 | 1 | 3 | 3 | 3 | 3 | 2 | 0 | 0 | 0 | 2 | 0 | 1 | 3 | 2 | 3 |
| CO5 | 1 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 2 | 0 | 1 | 3 | 2 | 3 |
| CO6 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 3 | 2 | 3 |
| Avg. | 1 | 1.83 | 1.33 | 1.33 | 1 | 1 | 0.17 | 0 | 0.5 | 1 | 0 | 0.83 | 3 | 2 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Implement the Polynomial representation using an Array and perform addition of two polynomials using Array.
2. Implement the Application of Stack : Conversion from Infix expression to Postfix.
3. Make the basic operations of circular Queue.
4. Implement the Polynomial representation using a Linked List and perform addition of two polynomials using Linked List.
5. Implement the Doubly Linked List.
6. Implement the Binary Tree and perform tree traversals.
7. Find the Shortest Path using Dijkstra's Algorithm.
8. Implement the Sorting using Quick Sort method.
9. Implement the Sorting using Merge Sort method.
10. Implement the Static Hashing using any one method.

## B. TECH. SEMESTER – IV (IT)

## SUBJECT: UNIVERSAL HUMAN VALUES - II

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 3 | - | - | 3 | 3 | 60 | - | - | - | 60 |

Reference Code HSMC-

## A. COURSE OBJECTIVES

**The objectives of teaching this course are:**

- To help the students appreciate the essential complementarity between 'VALUES' and SKILLS' to ensure sustained happiness and prosperity which are the core aspirations of all human beings.
- To facilitate the development of a Holistic perspective among students towards life and profession as well as towards happiness and prosperity based on a correct understanding of the human reality and the rest of existence. Such a holistic perspective forms the basis of Universal Human Values and movement toward value-based living in a natural way.
- To highlight plausible implications of such a Holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behavior, and mutually enriching interaction with Nature.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] COURSE INTRODUCTION

Need, Basic Guidelines, Content, and Process for Value Education Self Exploration–what is it? - it's content and process; 'Natural Acceptance' and Experiential Validation- as the mechanism for self exploration, Continuous Happiness, and Prosperity- A look at basic Human Aspirations, Right understanding, Relationship and Physical Facilities- the basic requirements for fulfillment of aspirations of every human being with their correct priority, Understanding Happiness and Prosperity correctly- A critical appraisal of the current scenario, Method to fulfill the above human aspirations: understanding and living in harmony at various levels

### [2] UNDERSTANDING HARMONY IN THE HUMAN BEING

Harmony in Myself! Understanding human being as a co-existence of the sentient 'I' and the material 'Body', Understanding the needs of Self ('I') and 'Body' - Sukh and Suvidha, Understanding the Body as an instrument of 'I' (I being the doer, seer, and enjoyer), Understanding the characteristics and activities of 'I' and harmony in 'I', Understanding the harmony of I with the Body: Sanyam and Swasthya; correct

appraisal of Physical needs, the meaning of Prosperity in detail, Programs to ensure Sanyam and Swasthya

### [3] UNDERSTANDING HARMONY IN THE FAMILY AND SOCIETY

Harmony in Human-Human Relationship Understanding Harmony in the family – the basic unit of human interaction, Understanding values in human to human relationship; the meaning of Nyaya and program for its fulfillment to ensure Ubhay-tripti; Trust (Vishwas) and Respect (Samman) as the foundational values of relationship, Understanding the meaning of Vishwas; Difference between intention and competence, Understanding the meaning of Samman, Difference between respect and differentiation; the other salient values in a relationship, Understanding the harmony in the society (society being an extension of the family): Samadhan, Samridhi, Abhay, Sah-astitva as comprehensive Human Goals, Visualizing a universal harmonious order in societyUndivided Society (Akhand Samaj), Universal Order (Sarvabhaum Vyawastha )- from family to world family.

### [4] UNDERSTANDING HARMONY IN THE NATURE AND EXISTENCE

Whole existence as Coexistence: Understanding the harmony in the Nature, Interconnectedness, and mutual fulfillment among the four orders of nature recyclability and self-regulation in nature, Understanding Existence as Coexistence (Sah-astitva) of mutually interacting units in all-pervasive space, Holistic perception of harmony at all levels of existence.

### [5] IMPLICATIONS OF THE ABOVE HOLISTIC UNDERSTANDING OF HARMONY IN PROFESSIONAL ETHICS

Natural acceptance of human values, Definitiveness of Ethical Human Conduct, Basis for Humanistic Education, Humanistic Constitution and Humanistic Universal Order, Competence in Professional Ethics, Case studies of typical holistic technologies, management models and production systems, Strategy for the transition from the present state to Universal Human Order.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. R.R Gaur, R Sangal, G P Bagaria, A foundation course in Human Values and professional Ethics, Excel books, New Delhi, 2010, ISBN 978-8-174-46781-2
2. A Nagaraj, JeevanVidya: Ek Parichaya, Jeevan Vidya Prakashan, Amarkantak, 1999.
3. A.N. Tripathi, Human Values, New Age Intl. Publishers, New Delhi, 2004.
4. Mohandas Karamchand Gandhi, The Story of My Experiments with Truth
5. E. F Schumacher, Small is Beautiful
6. Cecile Andrews, Slow is Beautiful
7. J C Kumarappa, Economy of Permanence
8. PanditSunderlal, Bharat Mein Angreji Raj

## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember and use concepts of human values in day to day life. | Remember |
| CO2 | Be able to understand the difference between values and skills, happiness and excitement, feeling of prosperity and accumulation of physical facilities, the Self and the Body, Intention and Competence of an individual, ethical and unethical practices etc. | Understand |
| CO3 | Be able to apply the understanding of ethical conduct to formulate the strategy for ethical life and profession. | Apply |
| CO4 | Be able to analyze the value of harmonious relationship based on trust and respect in their life and profession | Analyze |
| CO5 | Be able to examine the role of a human being in ensuring harmony in self, family , society and entire nature / existence. | Evaluate |
| CO6 | Be able to identify a strategy to actualize a harmonious environment at their workplace or institute and evaluate the significance of value inputs in formal education and and start applying them in their life and profession | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 2 | 0 |
| CO2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 3 | 0 | 0 | 0 | 0 | 2 | 0 |
| CO3 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 3 | 0 | 0 | 0 | 0 | 2 | 0 |
| CO4 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 1 | 0 | 0 | 0 | 0 | 2 | 0 |
| CO5 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 | 1 | 0 | 0 | 0 | 0 | 2 | 0 |
| CO6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 0 | 0 | 0 | 0 | 2 | 0 |
| Avg. | 0 | 0 | 0 | 0 | 0 | 1 | 1.5 | 2.17 | 2 | 0 | 0 | 0 | 0 | 2 | 0 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## B. TECH. SEMESTER – IV (IT)

## SUBJECT:DISCRETE MATHEMATICS

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 3 | 1 | - | 4 | 4 | 60 | 40 | - | - | 100 |

Reference Code PCC4XX

## A. COURSE OBJECTIVES

The objectives of teaching this course are:

- To impart an in-depth understanding of various concepts related to Discrete Mathematics, correct terminology, and notation.
- To teach how to construct correct direct and indirect proofs, the division into cases in a proof, use of counterexamples, etc.
- To explain Sets, Functions, Relations, Groups, Graphs, Trees, and their applications using real-world examples.
- Demonstrate and teach how to apply logical reasoning to solve a variety of problems.

## B. DETAILED SYLLABUS

| Unit | Topic(s) |
|---|---|
| [1] | **SETS AND PROPOSITIONS**<br>Combination of sets, finite, uncountable infinite, and infinite sets, mathematical induction, principles of inclusion, and exclusion, propositions. |
| [2] | **PROPOSITIONAL LOGIC**<br>Propositional Logic, Applications of Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Proof Methods and Strategy. |
| [3] | **PERMUTATIONS, COMBINATIONS, DISCRETE PROBABILITIES**<br>Rules of sums and products, permutations, combinations, generation, discrete probability, conditional probability, information. |
| [4] | **RELATIONS AND FUNCTIONS**<br>Relational model of databases, properties of binary relations, equivalence relation, partitions, partial ordering, lattices, chains and antichains, functions, and pigeon-hole principle. |
| [5] | **GRAPHS**<br>Basic terminology, multi- and weighted graphs, paths, circuits, shortest path, Eulerian path, Travelling Salesman problem, factors of a graph, planar graphs. |
| [6] | **TREES**<br>Trees, rooted trees, path length, prefix codes, binary search trees, spanning trees and cut-sets, minimum spanning trees, transport networks. |
| [7] | **RECURRENCE RELATIONS**<br>Linear recurrence relations with constant coefficient, homogeneous, particular and total solutions, generating functions, sorting algorithms, and matrix multiplication. |
| [8] | **DISCRETE NUMERICAL FUNCTIONS**<br>Manipulations of numerical functions, asymptotic behavior, generating functions, combinatorial problems. |

**[9] GROUP**

Groups and sub-groups, generators, evaluation of powers, cossets, Lagrange's theorem, permutation group and Burnsides theorem, group codes, isomorphism, automorphism, homomorphism, normal subgroups, rings, integral domains and fields, ring homomorphism, polynomial rings, and cyclic codes.

**[10] LATTICES AND BOOLEAN ALGEBRAS**

Lattices and algebraic systems, the principle of duality, properties of algebraic systems, distributive lattices, Boolean algebras, uniqueness, Boolean functions and expressions, propositional calculus.

## C. RECOMMENDED TEXT/ REFERENCE BOOK


1. Discrete Mathematics Applications, Kenneth H. Rosen, 7th edition, Mc Graw Hill
2. Elements of Discrete Mathematics, by: C.L. Liu, 2nd Ed. McGraw-Hill
3. Modern Applied Algebra, by: Birkoff and Bartee, McGraw-Hill, CBS.
4. Discrete Mathematics - A Unified Approach, by: Stephen A. Wiitala. Computer Science Series, McGraw-Hill.

## B. TECH. SEMESTER – IV (IT)

## SUBJECT: COMPUTER AND COMMUNICATION NETWORKS

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | - | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PCC-CS601

## A. COURSE OBJECTIVES

**The objectives of teaching this course are:**

- To teach about network hardware, software, services, and protocols and explain the uses of computer networks in daily life.
- To explain the layered architecture of network software and compare OSI and TCP/IP model
- To demonstrate the functionalities and working of different layers in TCP/IP protocol stack.
- To demonstrate and teach the usage of various network devices, their configurations, and setup

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] INTRODUCTION

Introduction and Applications of computer Networks; Network Hardware-LAN, MAN, WAN, internetworks; Network Software, Design Issues, Interfaces & Services, Connection-Oriented & Connectionless services, Service primitives, Relationship of services to protocols.

### [2] STUDY OF REFERENCE MODELS

Introduction OSI & TCP/IP, their comparison & critiques.

### [3] THE PHYSICAL LAYER

Guided Transmission Media: magnetic media, twisted pair, baseband & broadband, coaxial cable, fiber optics; Wireless Transmission: radio, microwave, infrared, lightwave.

### [4] THE DATA LINK LAYER

DLL Design issues Error Detection & Correction; Elementary Data link Protocols: Simplex protocol, Stop and Wait, Automatic Repeat Request, Sliding Window Protocols (1-bit sliding window, Go Back N, Selective Repeat Protocols); Examples of Data link layer protocols: HDLC, PPP..

### [5] MEDIUM ACCESS SUBLAYER

Channel Allocation Problem: Static & Dynamic, Multiple Access protocols (ALOHA, CSMA/CA AND CD, Collision Free Protocols, Limited contention protocols, WDMA, FDMA, TDMA, CDMA); Wireless LAN protocols, IEEE-802.3(Ethernet), 802.4(Token Bus), 802.5(Token Ring); Bridges: From 802.x to 802.y, transparent Bridges, Spanning Tree, Source Routing Bridges, remote bridge.

### [6] THE NETWORK LAYER

Network layer Design issues; Internetworking-How networks differ, how networks can be connected, concatenated virtual circuits, connectionless internetworking, and tunneling, internetwork routing; The network layer in the internet: the IP protocol, IPv4 Header, fragmentation, IP addresses & subnets, Internet Control Protocols – ARP, RARP, ICMP, IGMP; Routing Algorithms: Static Routing, Dynamic Routing, Intra-domain: Distance Vector Routing(RIP), Link-state (OSPF), Inter-domain Routing: Path vector (BGP).

### [7] THE TRANSPORT LAYER

The Transport Service: services provided to upper layers, transport services primitives; Elements of Transport Protocols; The Internet Transport Protocols; TCP service model: TCP protocol, TCP Segment Header, TCP Connection Management, TCP Transmission Policy, TCP Congestion Policy; UDP & overview of Socket.

### [8] CONGESTION CONTROL AND QUALITY OF SERVICE

Congestion control algorithm general policies, Congestion prevention policies, Traffic shaping, Flow specifications, Congestion control in VC subnets, Congestion controls in Datagram Subnets; Load shedding, jitter control, Quality of services-requirements, Techniques to achieve a good quality of services: Leaky bucket algorithm, Token bucket algorithm, Resource reservation, Admission control, Packet scheduling.

### [9] THE APPLICATION LAYER

Application Layer Protocols: File transfer protocol, Domain Name System, Electronic mail (SMTP, IMAP, POP), HTTP.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Behrouz A. Forouzan, Data Communications and Networking, 4th Edition, Publisher; Tata- McGraw Hill, 2007
2. Andrew S. Tanenbaum, Computer Networks, 4th Edition, Prentice-Hall of India(PHI), 2011.
3. William Stallings, Data & Computer Communications, eighth Edition, Maxell Macmillan Int, 2011
4. Leon Garcia & Widjaja, Communication Networks, Fundamental Concepts & key Architectures, Second edition, Publisher: Tata- McGraw Hill, 2004.

## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember Reference models, functionalities and protocols of each layer. | Remember |
| CO2 | Be able to understand design issues of each layer, service provided by each layer, protocols and techniques, functions of network devices. | Understand |
| CO3 | Be able to apply the concepts of Error control, Flow control, Access control, IP addressing, Subnetting-SuperNetting, Routing algorithm, Congestion control and network programming techniques to resolve the problem in an existing network or system. | Apply |
| CO4 | Be able to analyse working of various Data Link Layer protocols and techniques , Network protocols and techniques, network devices and configure it if necessary. | Analyze |
| CO5 | Be able to evaluate one or more techniques or protocols to solve or configure the given network or payload and be able to clearly decide a choice under the given context. | Evaluate |
| CO6 | Be able to design a small to medium efficient LAN, WAN in CISCO Packet tracer and client server paradigm for a given problem using the concepts of computer networks. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 2 | 3 |
| **CO2** | 3 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 2 | 3 |
| **CO3** | 3 | 3 | 1 | 2 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 |
| **CO4** | 3 | 2 | 0 | 1 | 0 | 3 | 1 | 0 | 0 | 0 | 0 | 1 | 3 | 2 | 3 |
| **CO5** | 3 | 2 | 0 | 1 | 0 | 3 | 1 | 0 | 0 | 0 | 0 | 1 | 3 | 2 | 3 |
| **CO6** | 2 | 2 | 1 | 2 | 3 | 3 | 1 | 0 | 0 | 3 | 1 | 0 | 3 | 2 | 3 |
| **Avg.** | 2.67 | 1.83 | 0.33 | 1 | 0.5 | 2.33 | 0.5 | 0 | 0 | 0.5 | 0.17 | 1.33 | 3 | 2 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Study of different types of Network cables and Practically implement the cross-wired cable and straight through cable using crimping tool.
2. Introduction to Packet tracer and study of Network devices in detail.
3. Implement a Local Area Network using Hub, Switch and Wireless access Point.
4. Study of network IP, Classification of IP address, Subnetting, Super netting.
5. Study of basic Network Commands, Router configuration and Connecting two different networks using Router in Packet tracer.
6. Configure a Network using Distance Vector Routing protocol: - Case study RIP.
7. Configure a Network using Link State Vector Routing protocol. Case study OSPF.
8. Connecting Wide Area Networks using PPP and HDLC.
9. Configure HTTP, DNS, DHCP and FTP servers in Packet tracer.
10. Write a program to implement TCP/IP protocol using socket programming using UNIX.

## B. TECH. SEMESTER – IV (IT)

## SUBJECT: DATABASE MANAGEMENT SYSTEM

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | – | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PCC-CS501

## A. COURSE OBJECTIVES

- To explain basic database concepts, data models, schemas, instances, and applications of database systems
- To demonstrate the use of constraints and relational algebra principles and operations.
- To describe the basics of SQL and construct queries using SQL / POSTGRESQL
- To emphasize the importance of normalization in databases.
- To facilitate students in Database designing and implementation through projects.
- To familiarize issues of concurrency control and transaction management.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] BASIC CONCEPTS

Purpose of database system, View of data, Database abstraction and Models, Database Languages, Transaction management, Storage management, Database administrator, Database users, Overall system structure..

### [2] ENTITY RELATIONSHIP MODEL

Entity sets, Relationship sets, Attributes, Constraints, Keys, Entity relationship diagrams, Weak entity sets, Generalization, Specialization,Aggregation, Design of an E-R database schema, Reduction of an E-R schema to tables.

### [3] RELATIONAL DATABASE MANAGEMENT SYSTEM

Relational Model Structure of database, Relational algebra, Extended relational algebra operation, tuple relational calculus, Domain relational calculus, Modification of database, Views. Structured Query Language Background, Basic structure. Integrity Constraints Domain constraints, Referential integrity, assertions, Triggers, Functional Dependencies. Database Design Pitfalls in relational database design, decomposition, Normalization, I, II, III normal Forms, Normalization using functional dependencies, Normalization using multi valued dependencies, Domain key normal form, Alternative approach to database design.

### [4] NO SQL

Introduction to NOSQL,Structure of NoSQL,NoSQL Queries

### [4] FILE SYSTEM STRUCTURE

Indexing & Hashing, File organization, Organization of records in files, Data dictionary storage, Basic concepts of indexing, Order indices,B- Tree index files, B+ -Tree index files, Static hashing & Dynamic Hashing.

### [5] QUERY PROCESSING

Overview, Catalog information for cost estimation, Measures of query cost, Selection operation, Sorting, Join operation, Other operations, Choice of evaluation plans.

### [6] TRANSACTION PROCESSING

Transaction concepts, Transaction state, Implementation of atomicity & durability, Concurrent executions, Serializability, Conflict Serializability, View serializability, Testing of conflict and view serializability.

### [7] CONCURRENCY CONTROL

Lock based protocols, Time-stamp based protocol, Validation based protocol, Multiple granularity, Multi-version schemes, Deadlock handling.

### [8] RECOVERY SYSTEM

Failure classification,Storage structure,Recovery & Atomicity,Log-based recovery, Shadow paging,Recovery with concurrent transactions,Buffer management, Failure with loss of non- volatile storage,Advance recovery techniques.

### [9] DISTRIBUTED DATABASES

Homogeneous and heterogeneous databases,Distributed Transactions

### [10] SECURITY AND INTEGRITY OF DATABASE

Overview of Security Mechanisms

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Silberschatz,Korth,Sudarshan,Database System Concepts, Fourth Edition,McGrow Hill publication, 2001
2. Ramez Elmasri, Sham Navathe, Fundamentals of Database Systems,Sixth Edition,Pearson Publication,2009
3. C.J.Date, An Introduction to Database Systems, 7th Edition , Addison-Wesley Publication,1999

## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember the core concepts of database management system including definitions, query syntaxes, file storage and recovery mechanism, etc. | Remember |
| CO2 | Be able to understand different types of keys, databases, transactions, concurrency control and requirement of database management system | Understand |
| CO3 | Be able to apply the concept of database design techniques to solve the given problem on different types of applications. | Apply |
| CO4 | Be able to analyse the given problem and be able to choose appropriate database concepts to create normalized and optimized databases. | Analyze |
| CO5 | Be able to evaluate the user requirements against Database Design and create Normal forms with transaction validation . | Evaluate |
| CO6 | Be able to design and create a solution for variety of applications for given problem statement using ER-model, Relational Schema, Data Dictionary and database diagram.Be able to apply concepts to write, document, debug, run and test SQL, NO SQL, PLSQL, triggers and cursors for applications. And should also be able to work as an individual and in a team with an approach of life-long learning for various types of Relational and Non - relational databases and perform CRUD operations using Postgresql or Oracle platform. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 3 | 3 |
| CO2 | 3 | 2 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 2 | 3 | 3 |
| CO3 | 2 | 3 | 1 | 1 | 1 | 2 | 1 | 0 | 0 | 1 | 0 | 1 | 2 | 3 | 3 |
| CO4 | 2 | 3 | 3 | 2 | 3 | 2 | 2 | 0 | 0 | 2 | 0 | 1 | 3 | 3 | 3 |
| CO5 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 2 | 0 | 1 | 2 | 3 | 3 |
| CO6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 1 | 3 |
| Avg. | | 1.83 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0.83 | 2.33 | 2.67 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Introduction to SQL, an exercise on data types in SQL & Data Definition Language Commands.
2. Exercise on Data Manipulation Language and transaction control commands.
3. Exercise on Types of Data Constraints.
4. Exercise on Joins (single-table or multi-table) and using normalization.
5. Exercise on group-by clause and date arithmetic.
6. Exercise on different functions (aggregate, math and String).
7. Exercise on different types of subqueries.
8. Introduction to PL/SQL, Control Structures, Procedures and Functions, view.
9. Introduction to triggers and cursors.
10. Mini project on designing and implementing a database management system.

## B. TECH. SEMESTER – IV (IT)

## SUBJECT: DESIGN & ANALYSIS OF ALGORITHM

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | -- | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PCC-CS404

## A. COURSE OBJECTIVES

- To make students analyze the asymptotic performance of algorithms.
- To make students familiar with major algorithms and corresponding application areas.
- To help students understand the important algorithmic design paradigms and their methods of analysis.
- To guide students for applying appropriate algorithms to solve common engineering design requirements.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] INTRODUCTION TO ALGORITHMS

Definition of algorithm,Characteristic of algorithms,Types of algorithm design
Paradigm ,Recursive Algorithms,The Need for Analysis.

### [2] ANALYZING ALGORITHMS

Analysis Techniques - Space and Time Complexity, Asymptotic Notations for analysis of algorithms - Omega, Theta, Big Oh, Little Oh, Little omega, Recurrence relations and Analysis of recursive algorithm, Recurrence tree, Substitution method, the Master method.

### [3] DIVIDE AND CONQUER ALGORITHM DESIGN STRATEGY

Introduction to Divide and Conquer, Binary search, Merge Sort, Quicksort.

### [4] GREEDY ALGORITHM DESIGN STRATEGY

Introduction to Greedy Methods; Knapsack Problem, Minimum Cost Spanning Trees, Optimal Merge Patterns, Single-Source Shortest Paths.

### [5] DYNAMIC PROGRAMMING DESIGN STRATEGY

Introduction to Dynamic Programming; Multistage Graphs, Matrix Chain

Multiplication, Single-Source and All-Pairs Shortest Paths, Travelling Salesperson Problem, Longest Common Subsequence.

### [6] BACK TRACKING

Graph Traversal using DFS/BFS; Articulation point in Graph; Introduction to Backtracking. N-Queens Problem, Graph Colouring, Hamiltonian Cycles.

### [7] BRANCH-AND-BOUND

Introduction to Branch and Bound; Knapsack problem, Job assignment problem;Comparison of backtracking and branch and bound.

### [8] NP-HARD AND NP-COMPLETE

Definition of P and NP classes; Relation between complexity classes; Examples of problems in various classes.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. G. Brassard, P.Bratley,Fundamentals of Algorithmics, PHI, 2nd edition.
2. T .H . Cormen, C . E . Leiserson, R .L . Rivest, Introduction to Algorithms,PHI,3rd edition.
3. Ellis Horowitz and Sartaz Sahani, Fundamentals of Computer Algorithms,Computer Science Press, 2nd edition.
4. Aho, Ullman, Addision Wesley, Design & Analysis of Computer Algorithms,.
5. Kunth,The art of Computer Programming, Addision Wesley,Vol. I & III, 2nd edition.


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember basic concepts of algorithm design and analysis, space complexity, time complexity ,complexity classes hierarchy . To remember the well known problems like searching problems, sorting problems etc. | Remember |
| CO2 | Be able to understand basic concepts of asymptotic analysis; divide and conquer, greedy, dynamic programming , backtracking and branch bound algorithm techniques; NP classes. | Understand |
| CO3 | Be able to apply the algorithm design techniques to solve the real world problems from different domains like searching, sorting, graph theory, optimization, etc. | Apply |
| CO4 | Be able to analyze the different problems and do asymptotic analysis of various design strategies used to solve the problem. | Analyze |

| | | |
|---|---|---|
| CO5 | Be able to evaluate the performance of the algorithm on various types of data. | Evaluate |
| CO6 | Be able to design a solution for a given problem statement using design techniques introduced i.e. dynamic programming, greedy algorithm etc. to design algorithms for more complex problems. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 2 | 2 | 2 |
| **CO2** | 2 | 2 | 1 | 1 | 0 | 0 | 2 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 2 |
| **CO3** | 2 | 2 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 2 | 2 | 2 |
| **CO4** | 2 | 2 | 1 | 2 | 3 | 0 | 1 | 0 | 0 | 2 | 1 | 3 | 2 | 2 | 2 |
| **CO5** | 2 | 3 | 2 | 3 | 3 | 0 | 1 | 0 | 0 | 2 | 0 | 3 | 2 | 2 | 2 |
| **CO6** | 2 | 2 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 2 | 2 |
| **Avg.** | 2 | 2 | 1.33 | 1.17 | 1.17 | 0 | 0.83 | 0 | 0 | 1.33 | 0.5 | 1.33 | 2 | 2 | 2 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Study usage of GNU profiler tool and use it to analyze recursive and iterative solutions of problems-
   1.1 Fibonacci Series
   1.2 Tower of Hanoi.
2. Do empirical comparison of following algorithms for solving searching problems.
   2.1 Sequential Search
   2.2 Binary Search
3. Do empirical comparison of following algorithms for solving sorting problems. -
   3.1 Selection Sort
   3.2 Insertion sort
4. Do empirical comparison of following divide and conquer based algorithms for solving sorting problems. -
   4.1 Quick Sort
   4.2 Merge Sort
5. Do empirical comparison of the Prim's and Kruskal's algorithm, for solving the Graph problem to find a minimum spanning tree.
6. Solve the Graph problem to find shortest path, using greedy technique- Dijkstra's algorithm
7. Solve the 0/1 knapsack problem using dynamic programming technique
8. Solve the string editing problem using dynamic programming technique
9. Solve the N-queens problem using backtracking technique
10. Solve the knapsack using backtracking technique.

## B. TECH. SEMESTER – IV (IT)

## SUBJECT: MICROPROCESSOR PROGRAMMING AND INTERFACING

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PCC

## A. COURSE OBJECTIVES

- To familiarize students about the architecture and internal organization of 8086,80286,80386 and other advanced microprocessors.
- To discuss the key concept of interfacing of 8086 processor with memory chips, programmable peripheral devices and Input/output devices.
- To provide the conceptual foundation of interrupts and handling of interrupts in 8086 and 80386 processors in various operating modes.
- To provide the practical exposure to the students for assembly language programming and turbo assembler tool.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] MICROPROCESSOR ARCHITECTURES

Introduction, Main features of 8086, 8086 Pin diagram, 8086 internal architecture, Machine cycle and Instruction Cycle, Minimum and Maximum Mode, 8086 Memory System

### [2] 8086 ASSEMBLY LANGUAGE PROGRAMMING

Program Development Steps, Constructing the Machine Codes for 8086 Instructions, Addressing Modes, Assembly Language Program Development Tools.

### [3] 8086 INSTRUCTION SET AND ASSEMBLER DIRECTIVES

Assembler Directives, Data Transfer Instructions, Arithmetic Instructions, Logical Instructions, Shift and Rotate Instructions, Transfer or Branch Control Instructions, Looping or Iteration Control Instructions, String Instructions, Processor Control Instructions.

### [4] STACKS, PROCEDURES AND MACROS

Stack Instructions, Defining and Calling Procedure, Parameter Passing Methods, Working with Macros.

### [5] 8086 INTERRUPTS AND I/O

The 8086 Interrupts, Interrupt types, Interrupt processing

### [6] BASIC INTERFACING TECHNIQUES

Interfacing memory, Peripheral devices interfacing, programming and interfacing of VLSI based peripheral Devices like 8255, 8254, 8259, DMA Controller etc.

### [7] 80286/386/486 MICROPROCESSORS

Multi User/Multitasking Operating System Concepts, Introduction to 80286/80386 /80486, The 80286386 Segments, Descriptor Tables and Selectors. Real Address Mode and Protected Virtual Address Mode of 80286/80386/80486, Multitasking and Exceptions.

### [8] MODERN MICROPROCESSORS

The Pentium Architecture, Hyper-Threading and Multi-core Technologies, Study of latest microprocessors.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Douglas V. Hall, S.S.S.P. Rao, Microprocessors and Interfacing (Programming & Hardware), McGraw Hill Education (India) Private Limited
2. M. T. Savaliya, 8086 Programming and Advance Processor Architecture, WILEY -India
3. Barry B. Brey, Intel Microprocessors 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium And Pentium Pro Processor
4. A. K. Ray, K. M. Bhurchandi, Advanced Microprocessors and Peripherals, Tata McGraw-Hill Publishing Company Ltd.


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember the internal architecture, pin diagram, and functionality of each component of the 8086/80286 and other microprocessors, as well as interfacing devices like the 8255 and 8259. | Remember |
| CO2 | Be able to understand the use of various microprocessors, components, and interfacing devices, as well as the concepts of assembly language programming. | Understand |
| CO3 | Be able to apply the working principle of the microprocessor and various concepts of interfacing and assembly language programming to solve the given problem. | Apply |
| CO4 | Be able to analyze the given problem or program requirements with reference to available concepts of microprocessors, their interfacing, and their programming with various devices and solve the problem. | Analyze |

| | | |
|---|---|---|
| CO5 | Be able to evaluate one or more options to solve the given problem or program requirements and be able to decide a choice under the given context clearly. | Evaluate |
| CO6 | Be able to design an application/circuit/solution/program for a given problem statement using concepts of microprocessor programming and interfacing. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 1 | 1 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 3 | 3 | 2 |
| **CO2** | 1 | 1 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 3 | 3 | 2 |
| **CO3** | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 | 3 | 2 |
| **CO4** | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 3 | 3 | 2 |
| **CO5** | 2 | 2 | 1 | 0 | 0 | 0 | 0 | 3 | 0 | 1 | 0 | 0 | 3 | 3 | 2 |
| **CO6** | 3 | 2 | 2 | 0 | 3 | 0 | 0 | 3 | 1 | 2 | 0 | 0 | 3 | 3 | 2 |
| **Avg.** | 1.83 | 1.17 | 1.33 | 0 | 1.17 | 0 | 0 | 1.17 | 0.17 | 1.17 | 0 | 0.67 | 3 | 3 | 2 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. I) Introduction to 8085-Microprocessor Trainer Kit
   II) Introduction to 8085- Microprocessor Simulator
   III) Program to study basic Arithmetic Operation in 8085 simulator and Kit.
2. Observing T-state, Machine Cycle and Instruction Cycle.
3. Study of DOS Debug Commands and Turbo Assembler
4. Programs to study arithmetic instructions, logical instructions and branch control instructions of 8086 instruction set.
5. Programs for sorting an array in ascending and descending order
6. Study of DOS and BIOS function calls
7. Study of string related instructions
8. Study of various parameter parsing techniques to a procedure in assembly language programming.
9. Study of implementation of recursion in assembly language.
10. Study multi module programs.
11. Study of the interrupt and user defined interrupt service routine.

## B. TECH. SEMESTER – V (IT)

## SUBJECT: COMPUTER ORGANIZATION

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 3 | 0 | 0 | 3 | 3 | 60 | 0 | 0 | 0 | 60 |

Reference Code: PCC-CS402

## A. COURSE OBJECTIVES

- To teach the fundamental concepts underlying modern computer organization and architecture.
- To impart knowledge regarding the hardware design including logic design, basic structure and behavior of the various functional modules of the computer and how they interact to provide the processing needs of the user.
- To explain the machine level representation of data, instruction sets, computer arithmetic, CPU structure and functions
- To discuss concepts of memory system organization and architecture, input/output organization, multiprocessors and system organization.

## B. DETAILED SYLLABUS

**Unit** **Topic(s)**

**[1] COMPUTING AND COMPUTERS**

Introduction, Organization Vs Architecture, Evolution of computers, Basic Fundamental blocks of computers: CPU, Memory, I/O subsystems, control unit, data path unit, interconnection structure.

**[2] CPU BASICS**

Data Representation: Basic formats, fixed point and IEEE floating point number representation format. Addressing modes, Instruction Sets: formats and types, RISC Vs. CISC.

**[3] DATAPATH DESIGN**

Arithmetic Logic Units: Combinational and Sequential ALUs, Ripple carry adders, Carry Lookahead Adders, ALU Expansion Techniques. Booth's multiplication algorithm, Division Algorithm with and without restoration, Floating Point Arithmetic.

**[4] CONTROL UNIT DESIGN**

Basic concepts, Hardwired Control Unit, Micro programmed Control Unit, Micro instruction types, Control field Encoding, Microprogram Sequencer, Nano-programming basics.

**[5] PIPELINING**

Basic concepts of pipelining, Instruction Pipeline, Arithmetic Pipeline, Pipeline Performance: Throughput, Speedup, Collisions and Task Initiation Decision, Pipeline Hazards

**[6] MEMORY ORGANIZATION**

Memory Technology, Memory Systems, Virtual Memory, Multilevel memory, segmentation, paging & page Replacement Policies, Address Translation & Memory Allocation, Cache memory: Mapping function – Direct & Set-associative cache, Cache Write Policy, Block Replacement Algorithms, Cache Coherence.

**[7] INPUT & OUTPUT ORGANIZATION**

Programmed IO, Direct Memory Access, Interrupts, IO Processors, Bus Arbitration.

**[8] PARALLEL COMPUTER ORGANIZATION**

Processor Level Parallelism, Multiprocessor Systems, Classification of Parallelism, Data Level Parallelism, Many-core Architecture: GPUs & Future Architectures.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. JOHN.P. HAYES, Computer Architecture and Organization,, Computer science series, McGRAW-HILL
2. Morris Mano, Computer System Architecture, PHI.
3. William Stallings, Computer Organization & Architecture, PHI


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember basic concepts and structure of computers & different number systems. | Remember |
| CO2 | Be able to understand functional units of digital computer & processor architectures. | Understand |
| CO3 | Be able to apply the concepts and demonstrate the problems on circuit design, addressing modes, instruction set, instruction formats and memory management. | Apply |
| CO4 | Be able to analyze the given problem and be able to choose appropriate concept(s) of computer organization to solve the problem | Analyze |
| CO5 | Evaluate cost performance and design trade-offs in designing and constructing a computer processor including memory | Evaluate |
| CO6 | Be able to design adder, ALU control unit and memory management circuit for given problem statement and to prepare required logic diagrams and truth table using concept of computer organization | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 3 | 0 |
| CO2 | 2 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 3 | 0 |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO3** | **2** | **3** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **1** | **0** | **3** | **0** |
| **CO4** | **2** | **3** | **1** | **1** | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **1** | **0** | **3** | **0** |
| **CO5** | **2** | **2** | **1** | **1** | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **1** | **0** | **3** | **0** |
| **CO6** | **2** | **2** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **3** | **0** |
| **Avg.** | **2** | **2.17** | **0.67** | **0.67** | **0** | **0** | **0.33** | **0** | **0** | **0.67** | **0** | **0.83** | **0** | **3** | **0** |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## B. TECH. SEMESTER – V (IT)

## SUBJECT: THEORY OF AUTOMATA & FORMAL LANGUAGE

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | 0 | 0 | 4 | 4 | 60 | 40 | 0 | 0 | 100 |

Reference PCC-CS502

## A. COURSE OBJECTIVES

- To Teach fundamental concepts of various languages and machine automata.
- To explain how to use FA, PDA and TM for various language grammar.
- To impart knowledge of various grammar sets like Regex., CFG, CSL, Rec. Enum.
- To discuss how to design various automata design techniques and implementation.
- To demonstrate skills to trace various language strings for different machine automata.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] REVIEW OF MATHEMATICAL BACKGROUND

Sets, functions, logical statements, proofs, relations, languages, Mathematical induction, strong principle, Recursive definitions

### [2] REGULAR LANGUAGES AND FINITE AUTOMATA

Regular expressions, regular languages, applications, Finite automata, memory requirement in a recognizer, definition, representation, expanded notation, string recognition, union, intersection and complement of regular languages. Nondeterministic finite automata, lambda transitions, equivalence, algorithms, examples. Kleen's theorem. Minimization of Finite automata. Non-regular and regular languages, criterion, Pumping Lemma, decision problems and decision algorithms, Regular languages in relation to programming languages

### [3] CONTEXT-FREE LANGUAGES AND PUSH-DOWN AUTOMATA

Context-free languages, definition, union, concatenation, examples etc. derivation tree and ambiguity. Simplified and Normal forms,Chomsky normal form. Push-Down Automata, definition, examples, deterministic PDA, two types of acceptances and their equivalence. Equivalence of CFG and PDA. Introduction to parsing, top-down and bottom-up parsing. Non-CFL and CFL, Pumping Lemma for CFL, intersection and complement.

### [4] TURING MACHINES

Models of computation, TM definition, combining TMs, computing a function with TMs. variations on Turing Machines, double infinite and more than one Tapes, non-deterministic and Universal TM, Recursively Enumerable languages, Unrestricted and context-sensitive grammars and their relation to TM, Linear Bounded Automata,

Chomsky hierarchy, Unsolvable problems, Halting problem, Post's correspondence, applications to CFLs. Computability, Primitive recursive functions, computable functions, PR functions, bounded operations. Non-primitive recursive functions.

### [5] INTRODUCTION TO COMPUTATIONAL COMPLEXITY

Tractable problems, growth rate, time complexity of TM. NP-completeness.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. John C. Martin, Introduction to Languages and Theory of Computation" McGraw-Hill 1991
2. Marvin L. Minsky, Computation: Finite and Infinite" Prentice-Hall, 1967


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember different types of automata and language grammar | Remember |
| CO2 | Be able to understand and give examples for the different types of automata like finite automata, push down automata, linear bounded automata and turing machines. | Understand |
| CO3 | Be able to apply the concepts, features and methods to minimize finite automata or generate grammar. | Apply |
| CO4 | Be able to analyze the given regular expression, context free grammar or automata with reference to different languages.<br>Be able to evaluate or test various automata based on given grammar, rule or language. | Analyze |
| CO5 | Be able to evaluate or test various automata based on given grammar, rule or language. | Evaluate |
| CO6 | Be able to design appropriate automata for the different requirements feasible and as per the limitations and features needed . | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 | 3 | 2 | 3 |
| **CO2** | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 | 3 | 2 | 3 |
| **CO3** | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 | 3 | 2 | 3 |
| **CO4** | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 | 3 | 2 | 3 |
| **CO5** | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 | 3 | 2 | 3 |
| **CO6** | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 | 3 | 2 | 3 |
| **Avg.** | 2 | 1.83 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 2 | 3 | 2 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## B. TECH. SEMESTER – V (IT)

## SUBJECT: APP DEVELOPMENT

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PCC

## A. COURSE OBJECTIVES

- To discuss and demonstrate application development for the android platform using core android components, asynchronous communication, and other features.
- To discuss key concepts of android apps interacting with various external services and demonstrate creating such applications.
- To discuss managing user data and multimedia on a mobile device via the Android framework libraries and demonstrate by creating applications.
- To exemplify major sensors available on mobile devices and demonstrate their usage to enhance user interaction and feedback for applications.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] ANDROID OS AND APPLICATION DEVELOPMENT ENVIRONMENT

Architecture of Android OS, Linux kernel and processes, display driver, keyboard driver, WiFi driver, flash memory driver, audio driver, camera driver, power management, Android OS & GUI architecture, Android runtime, Android API and Java interface, Android boot process, safe mode, recovery mode, application development environment, Android SDK and components, emulator, Android Studio, Gradle build, Android project structure and files.

### [2] BUILDING BASIC INTERACTIVE APPLICATION

Activity and lifecycle, resources, basic User Interface, layouts: relative, linear, grid, layout configuration parameter and units, GUI widgets, event handling, async tasks, multithreading, splash screen, etc.

### [3] ADVANCED GUI HANDLING

Menus, data organization using views and adapters: list view, grid view, adapters, modular design using fragments: fragment lifecycle, various fragments, layout, other interaction components: action bars, activity, intent, and application navigation, tabs, dialog boxes, Android services: log, toast messages, notification.

### [4] DATA PERSISTENCE: FILE, DATABASE

Database (SQLite), storage of user data: read/write files (internal storage, SD card), preferences

### [5] GRAPHICS, IMAGE, AND MULTIMEDIA

Graphics: canvas and 2D drawing, manipulating graphics, animation, image & video handling: open a resource using intent, access camera, Create and play image files, play audio file, create and play video files

### [6] USE OF DEVICES, SENSORS, EXTERNAL SERVICES

GPS, accelerometer, compass, WiFi, bluetooth, barcode, QR code reading, GET, POST, services, RESTful web services, JSON data, Single-Sign on, maps, email, SMS, USB debugging, Android Debug Bridge (adb), app deployment on playstore.

### [7] INTRODUCTION TO APP DEVELOPMENT USING FRAMEWORK

Layers of framework (e.g., flutter), introduction to the language of the framework (e.g., dart for flutter), creating app using widgets, layout, assets, Handling, gestures/events, state management, accessing API, development tools and app deployment.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Dawn Griffiths and David Griffiths, Head First Android Development, O'Reilly Medai, 3rd Edition, 2021
2. Eric Windmill, Flutter in Action, Manning, 2020
3. J F DiMarzio, Beginning Android Programming with Android Studio, Wrox, Wiley Brand
4. Android Developer Guides, Online, https://developer.android.com/guide
5. Ian Darwin, Android Cookbook: Problems and Solutions for Android Developers, O'Reilly
6. Alberto Miola, Flutter Complete Reference: Create beautiful, fast and native apps for any device, Independently Published (ISBN 9798691939952), 2020
7. Flutter documentation, Online, https://docs.flutter.dev/


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember and use Android programming concepts and API interfaces, classes, methods, features, and services. | Remember |
| CO2 | Be able to understand and use Android API, Widgets, and various services/capabilities provided by Android. | Understand |

| | | |
|---|---|---|
| CO3 | Be able to apply the concepts of basic and advanced GUI programming concepts to solve the given problem on GUI based application. | Apply |
| CO4 | Be able to analyze the given problem or program requirements with reference to available Android API, Widgets, and various services/capabilities and solve the problem. | Analyze |
| CO5 | Be able to evaluate one or more options on widgets, services, event handling, and sensors to solve the given problem or program requirements and be able to clearly decide a choice under the given context. | Evaluate |
| CO6 | Be able to design and create an android application for a given problem statement or application requirements. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 2 | 3 |
| **CO2** | 2 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 2 | 3 | 2 | 3 |
| **CO3** | 3 | 2 | 1 | 1 | 1 | 2 | 1 | 0 | 0 | 1 | 0 | 1 | 3 | 2 | 3 |
| **CO4** | 2 | 2 | 3 | 2 | 3 | 2 | 3 | 0 | 0 | 2 | 1 | 2 | 3 | 2 | 3 |
| **CO5** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 1 | 1 | 2 | 3 | 2 | 3 |
| **CO6** | 0 | 2 | 2 | 2 | 1 | 0 | 2 | 0 | 2 | 3 | 1 | 3 | 3 | 2 | 3 |
| **Avg.** | 1.83 | 1.67 | 1.5 | 1.5 | 1.17 | 1 | 1.5 | 0 | 0.33 | 1.33 | 0.5 | 1.83 | 3 | 2 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Installation, configuration, and study of android development environment and its required dependency components/software.
2. Implement an app to demonstrate use of basic GUI widgets and event handling.
3. Implement an app to demonstrate data organization using Views and Adapters.
4. Implement an app to demonstrate use of fragments, application navigation, and toast messages.
5. Implement an app to demonstrate use of SQLite and Preferences.
6. Implement an app to demonstrate use of Graphics API, Camera access, and Gallery Intent.
7. Implement an app to demonstrate use of splash screen, audio file, and Async task.
8. Implement an app to demonstrate use of GPS and WiFi devices
9. Implement an app to demonstrate use of Web Service using JSON data format.

10. Perform debugging using Android Debug Bridge.
11. Setup a development environment for flutter and implement an app using stateless widget.
12. Implement an app using stateful widget and assets.

Note: 5, 6, 7, and 9: each can be completed in 1 hour.

## B. TECH. SEMESTER – V (IT)

## SUBJECT: ADVANCED JAVA TECHNOLOGY (ELECTIVE-1)

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | - | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PEC

## A. COURSE OBJECTIVES

- To teach fundamental knowledge of design principles such creational/structural/behavioral patterns, collection framework.
- To explain how to use JDBC API for RDBMS in the web based applications for real-time solutions to problems.
- To impart knowledge of Server-Side Programming using Servlets and Java Server Pages.
- To discuss concepts of making user interactive programs using GUI/HTML/JSP programming, and demonstrate their use in solving problems.
- To demonstrate skills to write, debug, deploy and execute dynamic web based application programs and enable them to create enterprise solutions using MVC design patterns

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] GENERIC AND COLLECTIONS API

Generic, Collections, Collection, List, Set, Queue, Map, Iterator and Enumeration , Iterator Design Pattern

### [2] JAVA FUNCTIONAL PROGRAMING AND ANNOTATIONS

Functional programming: lambda functions, default methods, introduction to Stream processing, Java annotations.

### [3] DESIGN PRINCIPLES AND DESIGN PATTERNS

Design principles, interface, abstract class, inheritance, polymorphism, abstract class versus interface, object composition, inheritance, generic. Overview of design patterns, types of design patterns: Creational design pattern, Structural design patterns, and Behavioral design patterns. Design Patterns: Factory and Abstract Factory, Prototype, Singleton, Builder, Adapter, Decorator, Facade, Iterator.

### [4] JDBC API

JDBC versus ODBC, JDBC driver types, JDBC URL, Drivers Two-tier versus three tier models. API: DriverManager, Driver, Connection, Statement, ResultSet Operations: Insert, Update, Delete, Create, Select Mapping of SQL data types with Java data types, Prepared Statement, Transaction and Savepoint mechanism,

Enterprise JDBC (Support from Container): Enterprise Resource, DataSource, Connection pooling, JNDI Introduction to Data Access Object (model)

### [5] INTRODUCTION TO MVC, JAVA EE AND JAKARTA EE COMPONENTS

MVC Architect Design Pattern: JavaBean as a model, Servlet as a controller and JSP as a view. Decorator, MVC, Chain of responsibility design patterns.

### [6] SERVLETS

Jakarta Servlet, Introduction and use of Application server (Sun AS and Apache Tomcat). Web application structure (.war), deployment descriptor, packaging, development and deployment, Jakarta Servlet for a server-side API for handling HTTP requests and responses. Servlet life cycle, request-response concepts, and its operations. Variable Scopes: application, session, request Using JavaBean. Servlet Parameters: Context, Init Parameter, Session Management and Cookies (HttpSession and Cookie objects). Handling HTTP requests and responses, Handling GET/POST requests, Request Dispatcher, Status codes, errors, Response headers, Servlet: Events and Listener, Servlet Filter.

### [7] SERVER PAGES AND STANDARD TAG LIBRARY

Jakarta Server Pages, JSP life-cycle, and its phases, syntax and semantics of: Directives, Standard actions, scripting elements: declaration, expression, and scriptlet. Variable scopes, implicit objects, Comments, Collection and map management Using JavaBean in JSP, EL expressions. JSTL: i) Introduction to JSTL concepts: taglib, uri, and tld ii) core tags: set, out, redirect, url, import, param, control structures: if, forEach, forToken, choose, array, collection and map management iii) SQL tags: setDataSource, query, update, param, transaction.

### [8] WEBSOCKET AND JAVAMAIL API

Jakarta EE WebSocket: two-way communication channels, Jakarta Mail API.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Eric Freeman, Head First Design Patterns.
2. Keogh, James, J2EE Complete Reference Publication: McGraw-Hill.
3. Bryan Basham, Kathy Sierra, Bert Bates, Head First Servlets & JSP, 2nd Edition, Publication: O-reilly.
4. Bruce W. Perry, Java Servlet & JSP Cookbook Publisher: O'Reilly.

## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember advanced java programming concepts, core design pattern concepts such as MVC architecture, server, enterprise solutions etc. | Remember |
| CO2 | Be able to understand advanced java programming language and its concepts, that are used by industry in software and application developments. | Understand |
| CO3 | Be able to apply the concepts of Advanced Java API and design patterns, like Creational design pattern, Structural design patterns, Behavioral design patterns to solve the given problem in Web based application | Apply |
| CO4 | Be able to analyze the given problem and use various advanced java API and design patterns such as Collection API, patterns , JDBC/Servlet/JSP/Web-Server/Application-Server to solve the problem. | Analyze |
| CO5 | Be able to evaluate one or more options to solve the given problem or program requirements and be able to decide a choice under the given context clearly. | Evaluate |
| CO6 | Be able to create a solution for N-Tiered application for given problem statement and to prepare required design diagrams,MVC architecture, specification, class/interface structures, etc. using Advanced Java concepts | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 2 | 2 | 3 |
| **CO2** | 3 | 2 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 2 | 2 | 3 |
| **CO3** | 2 | 3 | 1 | 1 | 1 | 2 | 1 | 0 | 1 | 1 | 0 | 1 | 3 | 2 | 3 |
| **CO4** | 2 | 3 | 3 | 2 | 3 | 2 | 2 | 0 | 1 | 2 | 0 | 1 | 3 | 2 | 3 |
| **CO5** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 1 | 2 | 0 | 1 | 3 | 2 | 3 |
| **CO6** | 0 | 1 | 1 | 1 | 1 | 2 | 1 | 0 | 1 | 1 | 0 | 1 | 3 | 3 | 3 |
| **Avg.** | **1.83** | **2** | **1.33** | **1.17** | **1.17** | **1.33** | **1.17** | **0** | **1** | **1.17** | **0** | **1** | **2.67** | **2.17** | **3** |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Implement programming solution that uses abstract class, interface, object composition, and inheritance
2. Implement programming solutions that use Java-8 functional programming.
3. Implement programming solutions that implement factory, builder, and decorator design patterns.
4. Implement a GUI based application using JDBC API for database's CRUD operations.
5. Implement a Java application that can demonstrate jdbc's prepared statements, transaction management, and connection pooling.
6. Implement a Servlet to allow students to register as users.
7. Implement a web application that employs the use of session management.
8. Implement a web application that demonstrates an implementation of servlet filtering.
9. Implement a JSP based web application that demonstrates the use of the java bean.
10. Implement a JSP based (MVC application) web application which allows the user to edit his/her database information.
11. Implement a group chat application using websocket.
12. Integrate javamail in an existing web application.

## B. TECH. SEMESTER – V (IT)

## SUBJECT: INTERNET OF THINGS (ELECTIVE-1)

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PEC

## A. COURSE OBJECTIVES

- To teach fundamental knowledge of IoT boards,sensors,actuators
- To explain how to use IoT board, sensors for real time solutions to problem.
- To impart knowledge of wireless technologies, communication protocols, low power consumptions devices, device addressing,
- To discuss concepts to integrate IoT devices with cloud and security challenges.
- To demonstrate skills to design, write, debug, deploy and implement real time IoT applications.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] INTRODUCTION TO INTERNET OF THINGS

Introduction to IoT, Characteristics of IoT, Architectural Overview, Design principles and needed capabilities, IoT challenges, IoT Applications, Sensors, Actuators, Basics of Networking, M2M and IoT Technology Fundamentals - Devices and gateways, Data management, Business processes in IoT, Everything as a Service (XaaS), Levels of IoT and Security aspects in IoT.

### [2] IOT - HARDWARE

Design Methodology, Embedded computing logic, Microcontroller, System on Chips, IoT system building blocks, IoT Boards - Arduino, NodeMcu, Raspberry Pi, IDE programming - Interfaces and Raspberry Pi with Python Programming.

### [3] IOT PROTOCOLS

IoT Access Technologies - Physical and MAC layers - topology and Security of IEEE 802.15.4, 802.15.4g, 802.15.4e, 1901.2a, 802.11ah and LoRaWAN. Network Layer: IP versions, Constrained Nodes and Constrained Networks. Optimizing IP for IoT: From 6LoWPAN to 6Lo, Routing over Low Power and Lossy Networks. Application Transport Methods: Supervisory Control and Data Acquisition, Programming API's (using Python/Node.js/Arduino) for Communication Protocols-MQTT, XMPP, ZigBee, Bluetooth, CoAP, UDP, TCP.

### [4] IOT AND CLOUD

Role of Cloud in IoT, Solution framework for IoT applications- Implementation of Device integration, Data acquisition and integration, Device data storage - Unstructured data storage on cloud/local server, Authentication, authorization of devices.

### [5] IOT CASE STUDIES

IoT case studies and mini projects based on Industrial automation, Transportation, Agriculture, Healthcare, Home Automation, Smart city.

### [6] SECURITY IN IOT

Various security issues and need, architecture, requirement, challenges and algorithms.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Vijay Madisetti, Arshdeep Bahga, Internet of Things, "A Hands on Approach", University Press
2. Dr. SRN Reddy, Rachit Thukral and Manasi Mishra, "Introduction to Internet of Things: A practical Approach", ETI Labs
3. Pethuru Raj and Anupama C. Raman, "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", CRC Press
4. Jeeva Jose, "Internet of Things", Khanna Publishing House, Delhi
5. Adrian McEwen, "Designing the Internet of Things", Wiley
6. Raj Kamal, "Internet of Things: Architecture and Design", McGraw Hill
7. Cuno Pfister, "Getting Started with the Internet of Things", O Reilly Media


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember the components of IoT architecture, protocol stack, | Remember |
| CO2 | Be able to understand the IoT ecosystem and role of each component, required hardware and their specifications, security aspect. | Understand |
| CO3 | Be able to apply the concepts of data acquisition, hardware interaction, data transfer, data storage, data processing and data security to solve the given IoT problem. | Apply |

| | | |
|---|---|---|
| CO4 | Be able to analyze the given IoT problem or program requirements with reference to available devices, concepts, features and solve the problem. | Analyze |
| CO5 | Be able to evaluate one or more options to solve the given problem or program requirements and be able to clearly decide a choice under the given context. | Evaluate |
| CO6 | Be able to design and create an application/solution for given problem statement or IoT based application | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 1 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 3 | 2 | 3 |
| **CO2** | 1 | 1 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 3 | 2 | 3 |
| **CO3** | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 | 2 | 3 |
| **CO4** | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 3 | 2 | 3 |
| **CO5** | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 3 | 0 | 1 | 0 | 0 | 3 | 2 | 3 |
| **CO6** | 2 | 1 | 2 | 0 | 3 | 0 | 0 | 3 | 1 | 2 | 0 | 0 | 3 | 2 | 3 |
| **Avg.** | 1.33 | 0.83 | 1.33 | 0 | 1.17 | 0 | 0 | 1.17 | 0.17 | 1.17 | 0 | 0.67 | 3 | 2 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Familiarization with Arduino/Raspberry Pi and perform necessary software installation.
2. To interface LED/Buzzer with Arduino/Raspberry Pi and write a program to turn ON LED for 1 sec after every 2 seconds.
3. To interface Push button/Digital sensor (IR/LDR) with Arduino/Raspberry Pi and write a program to turn ON LED when push button is pressed or at sensor detection.
4. To interface DHT11 sensor with Arduino/Raspberry Pi and write a program to print temperature and humidity readings.
5. To interface motor using relay with Arduino/Raspberry Pi and write a program to turn ON motor when push button is pressed.
6. To interface OLED with Arduino/Raspberry Pi and write a program to print temperature and humidity readings on it.
7. To interface Bluetooth with Arduino/Raspberry Pi and write a program to send sensor data to smartphone using Bluetooth.

8. To interface Bluetooth with Arduino/Raspberry Pi and write a program to turn LED ON/OFF when ‘1’/’0’ is received from smartphone using Bluetooth.
9. Write a program on Arduino/Raspberry Pi to upload temperature and humidity data to thingspeak cloud.
10. Write a program on Arduino/Raspberry Pi to retrieve temperature and humidity data from thingspeak cloud.
11. To install MySQL database on Raspberry Pi and perform basic SQL queries.
12. Write a program on Arduino/Raspberry Pi to publish temperature data to MQTT broker.
13. Write a program on Arduino/Raspberry Pi to and print it.
14. Write a program to create TCP server on Arduino/Raspberry Pi and respond with humidity data to TCP client when requested.
15. Write a program to create UDP server on Arduino/Raspberry Pi and respond data to UDP client when requested.

## B. TECH. SEMESTER – V (IT)

## SUBJECT: SOFTWARE ENGINEERING (ELECTIVE-2)

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | - | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PEC

## A. COURSE OBJECTIVES

- To teach fundamental concepts and application of software development phases and SDLC models like Waterfall model, prototyping model, evolutionary model, agile model and spiral model.
- To discuss the importance and the concepts of software project management, requirement analysis, requirement specification, risk management and quality assurance for software projects.
- To teach various project estimation techniques like project size, project effort and project cost estimation.
- To demonstrate skills to design UML diagrams like use case diagram, class diagram, interaction diagram, activity diagram and state chart diagram using NetBeans.
- To demonstrate how to perform unit testing using JUnit and system testing using Selenium for defect free software projects.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] SOFTWARE DEVELOPMENT LIFE CYCLE

Overview, Need, Coverage of topics project feasibility analysis, Software Process Models, Incremental Process Models, Evolutionary, Process Models, Component based development Model, Rapid Application Development Model, Unified Process, Agile Methodology (Agile Process Model).

### [2] SOFTWARE PROJECT MANAGEMENT

Software Project management, Cost Estimation Techniques, Loc based estimation. FP-based estimation, COCOMO II. Project Scheduling and Timeline Charts.

### [3] RISK MANAGEMENT

Risk Management, Risk identification, Risk assessment & refinement, Change Management, Software configuration management.

### [4] REQUIREMENT ENGINEERING

Requirement engineering tasks, Initiating the Requirement Engineering Process, Eliciting Requirements, System Engineering, System Analysis: SRS document preparation.

### [5] SYSTEM DESIGN

Software Design concepts and principles, Architectural design, overview of function oriented design, Object oriented design, UML Diagrams like Use Case Model, Class Diagram, Interaction Diagram, Activity Diagram, State Chart Diagram, Package, Component and Deployment Diagrams.

### [6] USER INTERFACE DESIGN

Characteristics of a good user interface, concepts and type of user interface, Component based GUI Development and user interface design methodology.

### [7] SOFTWARE TESTING

Coding Standards, Code Review techniques, Software Testing strategies, Software Testing techniques, Black box testing, White box testing, Debugging concepts, Control structure testing, Object oriented testing and System testing methods.

### [8] SOFTWARE RELIABILITY

Hardware vs Software Reliability and Reliability Metrics

### [9] SOFTWARE QUALITY ASSURANCE

ISO 9000 Standard and SEI Capability Maturity Model

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Roger S. Pressman, Software Engineering - A practitioner's Approach, Seventh Edition, McGraw Hill Pub, 2009
2. Rajib Mall, Fundamentals of Software Engineering, Fourth Edition, PHI, 2014
3. Martin Fowler, UML Distilled, Third Edition, Pearson, 2004
4. Waman S Jawadekar, SOFTWARE ENGINEERING: Principles and Practice, Second Edition, Tata Mcgraw hill, 2008
5. Grady Booch, Object-oriented Analysis and Design with Applications, Third Edition, Pearson Edu., 2009


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to understand and use basic principle of software engineering, Object Oriented Design, Software Testing & Quality Assurance. | Remember |
| CO2 | Be able to understand the significance of software engineering and phases of SDLC model for the development of quality and cost-effective software. | Understand |

| | | |
|---|---|---|
| CO3 | Be able to apply the concepts of SDLC models like waterfall, prototype, spiral and agile to solve the given problem. Also be able to apply the concepts of software project management for project scheduling, team management and risk management. | Apply |
| CO4 | Be able to analyze the project definition to perform project cost estimation using techniques like COCOMO and Halstead metrics. | Analyze |
| CO5 | Be able to evaluate one or more software design and SDLC model option for the given project definition and be able to clearly decide a choice under the given context. | Evaluate |
| CO6 | Be able to prepare different diagrams like class, interaction, Activity, State-chart for given project definition also be able to create the project schedule as per the requirements. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 3 |
| **CO2** | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 3 |
| **CO3** | 2 | 0 | 2 | 2 | 2 | 0 | 2 | 1 | 1 | 2 | 3 | 1 | 2 | 3 | 3 |
| **CO4** | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 3 | 3 | 3 |
| **CO5** | 0 | 2 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 3 | 3 | 3 |
| **CO6** | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 3 | 1 | 2 | 1 | 2 | 3 | 3 |
| **Avg.** | **0.5** | **0.67** | **1** | **0.5** | **0.5** | **0.67** | **0.33** | **0.17** | **0.67** | **0.5** | **1.5** | **0.33** | **2.33** | **2.67** | **3** |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. To study phases of SDLC and applicability of different SDLC Models.
2. To perform requirement engineering tasks.
3. To perform the system analysis : Requirement analysis, SRS
4. To perform the user's view analysis : Use case diagram.
5. To draw the structural view diagram : Class diagram, Object diagram
6. To draw the behavioral view diagram : Sequence diagram, Collaboration diagram
7. To draw the behavioral view diagram : State-chart diagram, Activity diagram
8. To perform Project Management with an Agile Model using Jira Tool.
9. To perform software testing using the testing tool : Unit testing and Integration testing.
10. To perform software testing using the testing tool : System testing.
11. To draw the implementation view diagram: Component diagram.
12. To draw the implementation view diagram: Deployment diagram.

## B. TECH. SEMESTER – V (IT)

## SUBJECT: ADVANCED ALGORITHMS (ELECTIVE-2)

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | 0 | 2 | 6 | 5.0 | 60 | 40 | 25 | 25 | 150 |

Reference Code PEC

## A. COURSE OBJECTIVES

- To help students develop a sound theoretical understanding of advanced algorithms and practical problem solving skills using them
- To help students develop basic knowledge of a wide range of advanced algorithm design techniques including dynamic programming, linear programming, approximation algorithms, and randomized algorithms.
- To help students develop basic advanced algorithm analysis skills for analyzing the approximation ratio of approximation algorithms and the probability of randomized algorithms.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

**[1] OVERVIEW OF ALGORITHMS AND ASYMPTOTIC NOTATIONS.**
Review of order rotation & growth of functions, recurrences,

**[2] DESIGN PARADIGMS OVERVIEW :**
Overview of Divide and Conquer, Greedy and Dynamic Programming strategies.
Basic search and traversal techniques for graphs, Backtracking, Branch and Bound.

**[3] THEORY OF NP- HARD AND NP-COMPLETE PROBLEMS:**
P, NP and NP-Complete complexity classes; A few NP-Completeness proofs

**[4] STRING MATCHING:**
Introduction to string-matching problem, Naïve algorithm and complexity analysis.

**[5] APPROXIMATION ALGORITHMS:**
Introduction, Combinatorial Optimization, approximation factor, PTAS, FPTAS, Approximation algorithms for vertex cover, TSP, knapsack, etc.

**[6] PROBABILISTIC ALGORITHMS & RANDOMIZED ALGORITHMS:**
Numerical probabilistic algorithms, Las Vegas and Monte Carlo algorithms, Applications on graph problems

**[7] PARALLEL ALGORITHMS:**

Introduction, Models, speedup and efficiency, Some basic techniques, Examples .

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. T.H. Cormen, C.E.Leiserson and R.L. Rivest, Introduction to Algorithms.
2. G.Brassard and P.Bratley, Fundamentals of Algorithmics.
3. D.Harel, Algorithmics :The spirit of computing.
4. M.R. Garey and D.S. Johnson, Freeman, Computers and Intractability: A Guide to the Theory of NP-Completeness.


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | To remember basic concepts of algorithms, space complexity, time complexity ,complexity classes hierarchy . To remember the well known problems like searching problems, sorting problems etc. | Remember |
| CO2 | Be able to have understanding on a wide range of advanced algorithmic problems, their relations and variants, and application to real-world problems. | Understand |
| CO3 | Be able to apply the algorithm design techniques to solve the real world problems from different domains like searching, sorting, graph theory, optimization, etc. | Apply |
| CO4 | Be able to develop basic advanced algorithm analysis skills for analyzing the approximation ratio of approximation algorithms and the probability of randomized algorithms. | Analyze |
| CO5 | Be able to evaluate the performance of the algorithm using asymptotic analysis and describe the relative merits of the worst-, average-, and best-case of algorithm. | Evaluate |
| CO6 | Be able to design a solution for a given problem statement using design techniques including dynamic programming, approximation algorithms, and randomized algorithms. for more complex problems. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 3 | 2 | 3 |
| **CO2** | 2 | 2 | 1 | 1 | 0 | 0 | 2 | 0 | 0 | 1 | 1 | 2 | 3 | 2 | 3 |
| **CO3** | 2 | 2 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 3 | 2 | 3 |
| **CO4** | 2 | 2 | 1 | 2 | 3 | 0 | 1 | 0 | 0 | 2 | 1 | 3 | 3 | 2 | 3 |
| **CO5** | 2 | 3 | 2 | 3 | 3 | 0 | 1 | 0 | 0 | 2 | 0 | 3 | 3 | 2 | 3 |
| **CO6** | 2 | 2 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 | 2 | 3 |
| **Avg.** | **2** | **2** | **1.33** | **1.17** | **1.17** | **0** | **0.83** | **0** | **0** | **1.33** | **0.5** | **1.33** | **3** | **2** | **3** |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Solve problems using greedy, DP and branch bound techniques And compare.
2. Solve knapsack problem using approximation algorithm.
3. Solve traveling salesman problems using approximation algorithms.
4. Solve string matching problem using approximation paradigm.
5. Solve n queens problem using non deterministic algorithm.
6. Solve quicksort using randomized algorithm
7. Solve the primality testing problem using monte carlo algorithm.
8. Solve the primality testing problem using lasvegas algorithm.
9. Solve sorting problems using parallel algorithms.
10. Solve graph problem - Chinese postman problem ,using appropriate algorithm design paradigm.

## B. TECH. SEMESTER – V (IT)

## SUBJECT: E-COMMERCE AND E-SECURITY (OPEN ELECTIVE-1)

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 3 | 0 | 0 | 3 | 3.0 | 60 | 0 | 0 | 0 | 60 |

Reference Code OEC

## A. COURSE OBJECTIVES

- To explain and demonstrate the symmetric key cryptography, asymmetric key cryptography.
- To illustrate the Block cipher and Stream cipher algorithms with its importance.
- To discuss and Analyze the Public key cryptography and public key generation algorithms and distribution techniques.
- To impart knowledge of Hash Algorithms, MAC, and digital signatures with its applications.
- To explain cryptography concepts in the security protocol development such as kerberos,MIME , IPsec and SSL/TLS and other security mechanisms such as firewall.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] INTRODUCTION TO E-COMMERCE

Introduction to E-Commerce, Transactions on E-Commerce, Requirements of Security on E-Commerce

### [2] CONVENTIONAL ENCRYPTION TECHNIQUES

Introduction, Basic encryption techniques, simplified DES, block cipher mode of operation, traffic confidentiality and key distribution, Random Number Generation.

### [3] PUBLIC KEY CRYPTOGRAPHY

RSA algorithm, Key management, Elliptic Curve Cryptography, Diffie-Hellman Key Exchange

### [4] MESSAGE AUTHENTICATION AND HASH FUNCTIONS

Authentication requirement, Functions, Message Authentication Code (MAC), Hash Functions (SHA-1), Digital signature standard DSS).

### [5] NETWORK SECURITY

Authentication Protocols Like Kerberos, X.509 Directory Authentication Services

### [6] IP SECURITY EMAIL SECURITY

IP security overview, architecture, authentication header, Encapsulation security payload, S/Mime, Web security, Firewall.

### [7] SAFE ELECTRONIC COMMERCE

Secure transport protocol, secure E-payment protocol, secure electronic transaction.EPS,EDI,Types of Electronic Banking.

### [8] INTRODUCTION TO BLOCKCHAIN

Introduction to BlockChain with Architecture and Conceptualization, Types of ledger , Distributed Consensus, smart contract using solidity

### [9] INTRODUCTION TO LATEST SECURITY PROTOCOLS

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. William Stallings, Cryptography and Network Principles and Practice,Third Edition, Pearson Education, 2003
2. Daniel Minoli and Emma Minoli, Web Commerce Technology Handbook,Tata McGraw Hills, 1998
3. Imran Bashir, Mastering Blockchain: Distributed Ledger Technology, decentralization, and smart contracts explained, Second Edition,Packt Publishing, 2018
4. Er. Kailash Aseri and Mr. O.P. Gera., Current Trends of I. T. and Cyber Security,Horizon books publisher, 2014


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Able to remember encryption techniques and its applications | Remember |
| CO2 | Able to understand the importance of Asymmetric and symmetric cryptography techniques with various algorithms | Understand |
| CO3 | Be able to apply various concepts of cryptography like Asymmetric and symmetric algorithms and hash functions to solve various problems | Apply |
| CO4 | Be able to analyze the given problem and be able to choose appropriate concept(s) of cryptography, hashing and digital signature to solve the problem. | Analyze |
| CO5 | Be able to evaluate one or more security concepts to solve the given problem or program requirements and be able to choose the correct concept for problem solving. | Evaluate |

| | | |
|---|---|---|
| CO6 | Be able to design and create security applications/solutions for given problems based on requirements. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | **2** | **1** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **3** | **2** | **3** |
| **CO2** | **1** | **1** | **1** | **1** | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **1** | **3** | **2** | **3** |
| **CO3** | **1** | **2** | **1** | **1** | **1** | **2** | **1** | **0** | **0** | **1** | **0** | **1** | **2** | **3** | **3** |
| **CO4** | **1** | **2** | **2** | **2** | **3** | **2** | **2** | **0** | **0** | **2** | **0** | **1** | **2** | **3** | **3** |
| **CO5** | **2** | **3** | **3** | **2** | **2** | **2** | **2** | **0** | **0** | **2** | **0** | **1** | **3** | **2** | **3** |
| **CO6** | **2** | **1** | **2** | **2** | **2** | **2** | **0** | **0** | **0** | **0** | **0** | **1** | **3** | **2** | **3** |
| **Avg.** | **1.5** | **1.67** | **1.5** | **1.5** | **1.33** | **1.33** | **1** | **0** | **0** | **1** | **0** | **1** | **2.67** | **2.33** | **3** |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## B. TECH. SEMESTER – VI (IT)

## SUBJECT: DATA ANALYSIS AND INFORMATION EXTRACTION

## (OPEN ELECTIVE-2)

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 3 | - | - | 3 | 3 | 60 | - | - | - | 60 |

Reference Code OEC

## A. COURSE OBJECTIVES

- To teach the key concepts of data mining, data warehouse and data preprocessing techniques like data cleaning, data integration, data reduction and data transformation for maintaining the quality of data.
- To teach concepts of various data mining techniques and its application like classification and clustering for extraction of useful information from the data.
- To discuss different attribute types of data objects, the importance of attribute relevance analysis and how to select the right data mining techniques for a given problem.
- To impart knowledge of mining frequent patterns using association rule mining and correlation analysis.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] INTRODUCTION TO DATA MINING

An Overview of data mining, data mining primitives, KDD process, data-warehouse and data-mart.

### [2] DATA DESCRIPTION

Data Objects and Attribute Types, Statistical Description and Measuring Data Similarity and Dissimilarity.

### [3] DATA PRE-PROCESSING

Overview, need for pre-processing, Issues related to efficient data handling (Extraction, Transformation, and updating of large databases), Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization.

### [4] DATA WAREHOUSE AND A MULTI-DIMENSIONAL DATA MODEL

Data Warehouse Architecture, Data Warehouse Modeling: Data Cube and OLAP, Data Warehouse Design and Use.

### [5] DATA GENERALIZATION

Attribute Oriented Induction for Data Characterization and Class Comparisons.

### [6] MINING ASSOCIATION RULES

Basic Concepts, Market Basket Analysis, mining single-dimensional Boolean Association Rules from transactional database, Mining Multilevel Association Rules from transaction database, Mining multidimensional association rules from relational databases and data warehouses. From Association Mining To correlation analysis

### [7] CLASSIFICATION & PREDICTION

An Overview & Basic Concepts Classification by decision tree induction Bayesian Classification, Classification by Back Propagation, Classification Based on concepts from Association Rules Mining, Other methods, such as Genetic Algorithm, Fuzzy Set Approach, Case Based Reasoning, Etc. Prediction, Classifier Accuracy

### [8] CLUSTER ANALYSIS

An overview & basic concepts partitioning methods hierarchical method, Density-Based methods, Grid-based methods, Model-based clustering methods, Outlier analysis

### [9] MINING COMPLEX TYPES OF DATA

Mining Time Series & Sequence Data, Trend analysis Forecasting & Smoothing techniques.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Jiawei Han, Michelin Kambern & Jian Pei, Data Mining Concepts and Techniques, Third Edition, Morgan Kaufmann Publishers, (2011)
2. Sam Ananory & Dennis Murray, Data Warehousing in the Real World, Second Edition, Publisher: Addision-Wesley, (2000)
3. W. B. Frakes & R. Baeza - Yates, Information Retrieval: Data Structures & Algorithms, First Edition, Publisher: Prentice-Hall, New Jersey, (1992)
4. Michael J. A. Berry, & Gordon Linoff, Data mining techniques: For Marketing, Sales, Customer Support, Second Edition, Publisher: John Wil ey & Sons, 2004
5. Jit S. Chandan, Statistics for Business & Economics, First Edition, Publisher: Vikas Publishing. (1999)

## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember and use concepts of data mining and knowledge discovery process. | Remember |
| CO2 | Be able to understand and use concepts of data pre-processing, knowledge discovery, association rule mining, classification and clustering. | Understand |
| CO3 | Be able to apply the concepts of schema diagram, data cleaning and integration process, Data mining algorithm as per given problem. | Apply |
| CO4 | Be able to analyze inconsistency, normalization, incompleteness, interesting pattern and cluster analysis for given problem. | Analyze |
| CO5 | Be able to evaluate one or more classification and clustering algorithm for given problem and build efficient solutions. | Evaluate |
| CO6 | Be able to design an appropriate data mining pipeline for a given problem statement and to prepare required data mining model, algorithm and methods. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 2 | 2 | 2 | 2 |
| CO2 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 2 | 2 | 2 | 2 |
| CO3 | 2 | 1 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 2 | 3 | 2 |
| CO4 | 1 | 2 | 0 | 2 | 0 | 0 | 2 | 0 | 1 | 0 | 1 | 1 | 2 | 3 | 3 |
| CO5 | 1 | 3 | 1 | 1 | 1 | 0 | 2 | 1 | 1 | 0 | 0 | 1 | 2 | 3 | 3 |
| CO6 | 0 | 2 | 1 | 2 | 0 | 0 | 2 | 1 | 1 | 1 | 0 | 1 | 2 | 3 | 3 |
| Avg. | 0.67 | 1.33 | 0.33 | 1.5 | 0.17 | 0.67 | 1 | 0.33 | 0.5 | 0.67 | 0.17 | 1.33 | 2 | 2.67 | 2.5 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## B. TECH. SEMESTER – VI (IT)

## SUBJECT: LANGUAGE TRANSLATOR

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PCC-CS601

## A. COURSE OBJECTIVES

- To enrich knowledge of language translation and compiler design.
- To introduce languages concepts of grammars, parsing, semantic analysis, intermediate code, code generation, code optimization, symbol table management and programming paradigms.
- To extend the knowledge of parser by parsing LL parser and LR parser.
- To provide practical programming skills necessary for constructing a compiler.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] LANGUAGE TRANSLATION OVERVIEW

Phases in language translation, overview of system software used during translation – language processors, linker, loader. Types of language processors –assembler, interpreter, compiler etc.

### [2] COMPILER STRUCTURE

Analysis-synthesis model of compilation, various phases of a compiler.

### [3] LEXICAL ANALYSIS

Interface with input, parser and symbol table, token, lexeme and patterns. Difficulties in lexical analysis. Error reporting. Implementation. Regular definition, Transition diagrams

### [4] SYNTAX ANALYSIS

CFGs, ambiguity, associativity, precedence, Top-down parsing-recursive descent parsing, LL Parser, transformation on the grammars (Left recursion, left factoring), predictive parsing, Bottom-up parsing, operator precedence grammars, LR parsers (SLR, LR)

### [5] SYNTAX DIRECTED TRANSLATION

Inherited and synthesized attributes, L- and S-attributed definitions, semantic stacks in bottom- up compilation, action symbols in top-down

### [6] INTERMEDIATE CODE GENERATION

Intermediate languages, Issues in implementation.

### [7] RUN TIME SYSTEM

Storage organization, activation tree, activation record, parameter passing, symbol table, static, dynamic and heap storage allocation, garbage collection.

### [8] SYMBOL TABLE MANAGEMENT

Symbol table organizations for blocked and non-blocked languages.

### [9] CODE OPTIMIZATIONS

Machine dependent, machine independent optimizations

### [10] ERROR DETECTION AND RECOVERY

classes of errors, error handling strategies

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Jean Paul Tremblay and Paul G. Sorenson, The theory and practice of Compiler Writing, BS Publications
2. A. V. Aho, R. Sethi, and J. D. Ullman, Compilers: Principles, Techniques and Tools, Publisher Addison -Wesley
3. Allen Holup, Compiler design in C Publisher, PHI
4. Dhamdhere, Compiler Construction Publisher, McMillan India


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember the types of translators, phases of translators. | Remember |
| CO2 | Be able to understand the different phases of compiler., syntax-directed translation and run-time environment. | Understand |
| CO3 | Be able to apply different algorithms to sample language and generate various processing phases of the translator. | Apply |
| CO4 | Be able to analyze the merits and demerits of various parsers, intermediate code generators etc. used at various phases . | Analyze |
| CO5 | Be able to compare various design choices for designing the various phases of the compiler. | Evaluate |
| CO6 | Design a lexical analyzer and other phases for a sample language | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 |
| **CO2** | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 |
| **CO3** | 2 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 |
| **CO4** | 0 | 2 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 |
| **CO5** | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 |
| **CO6** | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 |
| **Avg.** | 1.33 | 1.17 | 0.83 | 0.83 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Study the Compilers and various related tools
2. Study flex tool.
3. Using the flex tool generate a scanner for a language.
4. Write a program for generating scanner software (w/o using any automated tools)
5. Study yacc tool.
6. Implement a language processor using yacc.
7. Write a program for left recursion problem removal.
8. Write a program for left factoring problem removal.
9. First and Follow set computation for LR parser.
10. Write a program to implement a predictive parser(RDP).

## B. TECH. SEMESTER – VI (IT)

## SUBJECT: APPLIED OPERATING SYSTEM

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | - | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PCC-CS403

## A. COURSE OBJECTIVES

- To teach system calls for managing processes, memory and the file system.
- To make students understand the data structures and algorithms used to implement an OS.
- To explain the services provided by the operating system.
- To impart knowledge of the structure and organization of the file system.
- To demonstrate the process, how processes are synchronized and scheduled.
- To discuss deadlock management in process.
- To make them understand different approaches to memory management, virtual memory,etc.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] INTRODUCTION

What is an OS?, Simple Batch Systems, Multi programmed Batched Systems, Time Sharing Systems, Personal-Computer Systems, Parallel Systems, Distributed Systems, Real-Time System

### [2] OPERATING SYSTEM STRUCTURES

Computer-System Operation, I/O Structure, Storage Structure, Storage Hierarchy, System components, OS services, System calls, System programs, system structure, Virtual machines, Cloud OS, Hypervisor(VMM), Types of hypervisors, Software Virtualization.

### [3] PROCESS MANAGEMENT

Process concept, Process Scheduling, Operation on Processes, Process Creation, Process Deletion, Cooperating processes, Interprocess Communication using pipe and shared memory.

## [4] CPU SCHEDULING

Basic concept, scheduling criteria, scheduling algorithms, First Come First Serve algorithm, Shortest Job First algorithm, Shortest Remaining Time First algorithm , Priority Scheduling algorithm. Round Robin algorithm, Multilevel Feedback Queue algorithm.

## [5] PROCESS SYNCHRONIZATION

Critical section problem, synchronization hardware,Test and Set Instructions, Swap Instructions, semaphores, classic problems of synchronization, Reader Writer problem, Bounded buffer producer consumer problem, dining philosopher problem, critical regions, monitors.

## [6] DEADLOCKS

Deadlock characteristics, methods for handling deadlocks, deadlock prevention, deadlock avoidance, deadlock detection, recovery from deadlocks, combined approach for deadlock handling.

## [7] MEMORY MANAGEMENT

Logical versus Physical Address space, Swapping, Contiguous Allocation, Paging, TLB, structure of page table, Hierarchical paging, Hashed Page Table, Inverted Page Table, Segmentation, Segmentation with Paging

## [8] VIRTUAL MEMORY

Demand Paging, Performance of Demand Paging, Page Replacement, Page replacement algorithms, FIFO page replacement, Optimal page replacement, LRU page replacement, Allocation of frames, Thrashing, Other Considerations, Demand segmentation.

## [9] FILE-SYSTEM INTERFACE

File concept, File attributes, File operations, Access methods, Directory Structure, Protection, Consistency

## [10] FILE-SYSTEM IMPLEMENTATION

File-System Structure, allocation methods - contiguous allocation, linked allocation, Indexed allocation , Free-space Management, Directory Implementation, Efficiency and performance

## [11] I/O SUBSYSTEMS

I/O Hardware, Interrupts, Direct Memory Access, Application I/O interface

### [12] SECONDARY-STORAGE STRUCTURE

Disk Structure, Disk scheduling - FCFS Scheduling, SSTF Scheduling, SCAN Scheduling, C-SCAN Scheduling, C-LOOK Scheduling , Disk Management, Swap-space management

### [13] PROTECTION

Goals of protection, domain of protection, access matrix, implementation of access matrix, revocation of access rights, capability based systems, languages based protection.

### [14] SECURITY

The problem, authentication, one-time password program threats, system threats, threat monitoring, encryption, computer security classification.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Silbertschatz, Galvin, Operating System Concept, 9th ed,Addison Wesley, 2013
2. Milan Malinkovic, Operating system Concepts TMH, 2nd ed.
3. William Stallings, Operating System: PHI, 2nd ed.


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember the definitions ,structures, models and concepts of Operating System | Remember |
| CO2 | Be able to understand and use the system calls , functions, concepts of process, concurrency, synchronization , memory management, virtual memory, deadlock,etc. | Understand |
| CO3 | Be able to Illustrate and apply the concept of operating systems in various domains of concurrent transactions, virtual memory, CPU Scheduling, Paging, etc | Apply |
| CO4 | Be able to analyze and compare algorithms for given Operating system problem and solve it | Analyze |
| CO5 | Be able to critique and judge the algorithms for various domains of inter process communication, process communication, file systems , memory management, Concurrent processes, etc. | Evaluate |
| CO6 | Be able to implement ,document, debug, and run the programs to solve problems related to scheduling, handling processes, pipes, semaphores, shared memory etc. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 3 | 3 |
| CO2 | 2 | 2 | 2 | 1 | 0 | 3 | 0 | 0 | 0 | 1 | 0 | 2 | 1 | 3 | 3 |
| CO3 | 2 | 2 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 2 | 2 |
| CO4 | 2 | 2 | 1 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 3 | 2 | 2 |
| CO5 | 2 | 2 | 2 | 2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 3 | 2 | 3 |
| CO6 | 2 | 2 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 3 | 2 | 3 |
| Avg. | 2 | 1.83 | 1.33 | 1.33 | 0.67 | 0.5 | 0.33 | 0 | 0 | 0.33 | 0 | 1 | 2.33 | 2.33 | 2.67 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Study of UNIX commands with options.
2. Program maintenance using make utility.
3. Study system calls related to process & process control.
4. Study system calls related to file operations.
5. Study of functions related to threads (POSIX).
6. Inter process communication (POSIX-IPC) using pipe.
7. Inter process communication (POSIX-IPC) using shared memory.
8. Study system calls related to semaphore.
9. Simulation of Process scheduling algorithm: Feedback policy.
10. Simulation of I/O requests scheduling algorithm: Elevator algorithm.
11. Simulation of deadlock handling algorithm: Banker's algorithm.
12. Simulation of Memory management algorithm: LRU page replacement algorithm.

## B. TECH. SEMESTER – VI (IT)

## SUBJECT: DATA ANALYTICS USING PYTHON

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 0 | 0 | 2 | 2 | 1 | 0 | 0 | 25 | 25 | 50 |

Reference Code PCC

## A. COURSE OBJECTIVES

- To teach basic fundamentals of Python programming like operator, loop, sets, condition, function.
- To explain how to use python libraries like sci-kit learn, numpy, matplotlib.
- To discuss how to handson different data pre-processing techniques like cleaning, transformation, wrangling, re-shape.
- To impart the knowledge of data plotting and visualization charts like line plot, bar chart, pie-chart, box plot.
- To demonstrate skill to develop data models like regression, decision tree, naive bayes, text mining, clustering, web scraping.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] Basics of Python Programming

Variables, Operators, Data Structures, Condition, Loop, Functions

### [2] Getting Start with Numpy and Pandas

Creating and accessing Data Vector, Maths Operations, Data Frame creation, reading & writing, Handling Data Frame

### [3] Data Pre-Processing and Data Wrangling

Data Cleaning, Transformation, String Manipulation, Data Wrangling join, combined and re-shape

### [4] Plotting and Visualization

Line plot, Bar-chart, Histograms, Box-plot, Pie-chart using Matplotlib

### [5] Model Development

Regression and Correlation Analysis, Decision Tree, Naive Bayes and K-NN Classification, K-Means Clustering

### [6] Text Mining using Textblobs and NLTK

TF-IDF Generation, Sentiment Analysis, Word-Cloud, POS tagging

### [7] Web-Scraping

Sample Web-scraping Example using BeautifulSoup and Pandas library

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Wes McKinney, Python for Data Analysis Second Edition, O'REILLY Publishers
2. Jiawei Han & Micheline Kamber, Data Mining – Concepts & Techniques, Publisher: Morgan Kaufmann Publishers
3. John Paul Mueller & Luca Massaron, Python for Data Science for Dummies, Wiley Brand Publishers


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember and use python programming language concepts, libraries, methods. | Remember |
| CO2 | Be able to understand concept of data pre-processing, handling and analysis | Understand |
| CO3 | Be able to apply data pre-processing steps, libraries and methods for data analytics. | Apply |
| CO4 | Be able to analyze the given datasets or data modelling techniques, concepts, and python libraries to solve the problem. | Analyze |
| CO5 | Be able to evaluate one or more data analytics techniques and compare the result outputs | Evaluate |
| CO6 | Be able to design and create a solution or model to predict, classify hidden patterns using visualization techniques. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 3 | 2 | 2 | 3 | 3 | 0 | 0 | 3 | 0 | 3 | 0 | 2 | 3 | 2 | 3 |
| CO2 | 3 | 2 | 2 | 2 | 3 | 0 | 0 | 3 | 0 | 3 | 0 | 2 | 3 | 2 | 3 |
| CO3 | 3 | 2 | 2 | 2 | 3 | 0 | 0 | 3 | 0 | 3 | 0 | 2 | 3 | 2 | 3 |
| CO4 | 3 | 2 | 2 | 3 | 3 | 0 | 0 | 3 | 0 | 3 | 0 | 2 | 3 | 2 | 3 |
| CO5 | 3 | 2 | 2 | 3 | 3 | 0 | 0 | 3 | 0 | 3 | 0 | 2 | 3 | 2 | 3 |
| CO6 | 3 | 2 | 2 | 3 | 3 | 0 | 0 | 3 | 0 | 3 | 0 | 2 | 3 | 2 | 3 |
| Avg. | 3 | 2 | 2 | 2.67 | 3 | 0 | 0 | 3 | 0 | 3 | 0 | 2 | 3 | 2 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Introduction to Python programming for Data Analytics..
2. To perform creating and accessing Data Vector, Maths Operations, Data Frame creation, reading & writing, Handling Data Frame using Pandas and Numpy.
3. To perform Data Pre-processing and Wrangling tasks like data cleaning, transformation, join, re-shape, string manipulation on sample data-set.
4. To perform Data Visualization and plotting techniques like Line plot, Bar chart, Pie chart, Box plot using Matplotlib libraries.
5. To perform Regression Analysis using Sci-kit learn package in Python.
6. To perform Decision Tree Classification using Sci-kit learn package in Python.
7. To perform Naive Bayes and K-NN (k-nearest neighbor) Classification technique using Sci-kit learn package in python.
8. To perform KMeans and DBSCAN Clustering technique using Sci-kit learn package in python.
9. To perform Text Mining using textblob in python (TF-IDF generation, sentiment analysis, word-cloud, POS tagging).
10. To perform basic Web-Scraping tasks using the Beautifulsoup package in Python.

## B. TECH. SEMESTER – VI (IT)

## SUBJECT: FULL STACK DEVELOPMENT (ELECTIVE-3)

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PEC

## A. COURSE OBJECTIVES

- To provide conceptual foundations of client-side, server-side, and ORM frameworks.
- To demonstrate features and capabilities of frameworks in client-side and server side applications for solving problems.
- To provide practical exposure to using frameworks in solving problems related to web applications and enterprise applications.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] INTRODUCTION TO FULL STACK DEVELOPMENT

Architecture and components of full stack project, web browser developer tools, back-end server and frameworks, database servers, front-end frameworks, stages of development and various tools.

### [2] FOUNDATION OF FRONT-END TECHNOLOGIES

Semantic HTML, CSS, AJAX, JSON, Java Script: var, let, const, DOM API, module import/export, error handling, Advanced JavaScript: arrow function, closure, promise, async, await.

### [3] React JS UI LIBRARY

Components, introduction to class components and lifecycle methods, functional components, JSX, rendering, props, state, forms and event handling, JSX expressions & conditional rendering, lists and keys, hooks: useState, useReducer, useEffect with axios, useContext, useRef, useMemo, useCallback, React router: router API, useNavigate, useParams, patterns: HoC and render prop.

### [4] STATE MANAGEMENT

State management with Redux: store, action, action creator, reducer, dispatch, connect, useSelector, useDispatch, Introduction to Redux middleware: redux thunk, async action creator.

### [5] ORM FRAMEWORK-HIBERNATE

ORM, Hibernate framework, Hibernate API interfaces and classes, hibernate configuration and properties for database connectivity, hbm2ddl, Hibernate mapping file, JPA annotations, mapping between tables: one to one, one to many, bidirectional mapping, many to one, many to many, CRUD operations and DAO, hibernate object state.

### [6] BACKEND MVC FRAMEWORK: SPRING AND SPRING MVC

Spring framework, dependency injection/inversion of control, introduction to features of spring framework, Front Controller design pattern and Spring MVC, Flow of Spring MVC application, Spring MVC configuration, annotations, path variable, request parameter, form submission and validation, session, introduction to template engine-Thymeleaf.

### [7] API DEVELOPMENT WITH SPRING BOOT

Spring Boot framework, class level and method level annotations, application config file, JSON request handling, JSON response generation, creating REST services for various HTTP methods, API testing with POSTMAN, layered architecture of backend application, DTO, controller, service, repository with Spring Data JPA, model, mapper.

### [8] INTRODUCTION TO SPRING SECURITY

Spring Security architecture, filter chain and Interceptor design pattern, SecurityContext, authentication API, authorization API, authority, roles, permissions for URLs, method level security.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Ranga Rao Karanam, Mastering Spring 5: An effective guide to build enterprise applications using Java Spring and Spring Boot framework, 2nd Edition, 2019
2. Mark Tielens Thomas, React in Action, 2018
3. Laurentiu Spilca, Spring Security in Action, Manning, 2020.
4. Hibernate ORM Documentation, Online, https://hibernate.org/orm/
5. React JS Documentation, Online, https://reactjs.org/
6. Spring Framework Documentation, Online, https://docs.spring.io/spring-framework/docs/current/reference/html/
7. Iuliana Cosmina, Pivotal Certified Professional Spring Developer Exam-A Study Guide, Apress

## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember and use features and capabilities of frontend and backend frameworks. | Remember |
| CO2 | Be able to understand and use various concepts, API, architecture, and tools of React, Hibernate, and Spring frameworks. | Understand |
| CO3 | Be able to apply the programming concepts of React, Hibernate, and Spring frameworks to solve the given problem on UI creation, repository creation, service creation, and API creation. | Apply |
| CO4 | Be able to analyze the given problem or program requirements with reference to available concepts, features, and API of React, Hibernate, Spring MVC, Spring Security, and Spring Boot. | Analyze |
| CO5 | Be able to evaluate one or more options to solve the given problem or program requirements or design requirements and be able to clearly decide a choice under the given context. | Evaluate |
| CO6 | Be able to design and create an application/solution for given problem statement or application requirements using React, Hibernate, Spring MVC, Spring Security, and Spring Boot frameworks. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 3 | 3 |
| **CO2** | 2 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 2 | 3 | 3 | 3 |
| **CO3** | 3 | 2 | 1 | 1 | 1 | 2 | 1 | 0 | 0 | 1 | 0 | 1 | 3 | 3 | 3 |
| **CO4** | 2 | 2 | 3 | 2 | 3 | 2 | 3 | 0 | 0 | 2 | 1 | 2 | 3 | 3 | 3 |
| **CO5** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 1 | 1 | 2 | 3 | 3 | 3 |
| **CO6** | 0 | 2 | 2 | 2 | 1 | 0 | 2 | 0 | 2 | 3 | 1 | 3 | 3 | 3 | 3 |
| **Avg.** | 1.83 | 1.67 | 1.5 | 1.5 | 1.17 | 1 | 1.5 | 0 | 0.33 | 1.33 | 0.5 | 1.83 | 3 | 3 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Implement a JavaScript based app to demonstrate the use of manipulating DOM nodes and their attributes using DOM API within a web page created using HTML and CSS.
2. Implement a JavaScript based app using modern features: arrow function, promise, async, and await and understand troubleshooting using Browser's Developer Tools..
3. Implement a React JS based app using functional components to demonstrate use of the following (1) JSX, conditional rendering, lists, and props, (2) event handling of form fields with their state managed using useState.
4. Implement a React JS based app using functional components to demonstrate the use of the following: (1) useEffect with calling API using axios, (2) context API with reducer.
5. Implement a React JS based app using functional components to demonstrate use of routing in react..
6. Implement state management in React JS app and use redux thunk to perform async operations.
7. Implement a hibernate based application to demonstrate CRUD operations using DAO pattern.
8. Implement a hibernate based application to demonstrate relationships among hibernate entities.
9. Implement a SpringMVC based application to perform form handling with data binding and its use in JSP.
10. Implement a SpringMVC based application to perform form validation and its use in Thymeleaf template file.
11. Implement a Spring Boot based application to demonstrate the use of GET and POST API interacting with a database using JPA repository and API testing using POSTMAN.
12. Implement a Spring Boot and Spring Security based application to demonstrate use of authentication and authorization.

Note: 1, 5, 7, and 8: each can be completed in 1 hour.

## B. TECH. SEMESTER – VI (IT)

## SUBJECT: CLOUD COMPUTING (ELECTIVE-3)

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | - | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PEC

## A. COURSE OBJECTIVES

**The objectives of teaching this course are:**

- To teach the fundamental concepts and technologies for cloud computing and cloud infrastructure.
- To impart the concepts of compute virtualization, storage virtualization, network virtualization and their usage in cloud infrastructure.
- To discuss the VM lifecycle model, Open Source Cloud Implementation and its Administration.
- To explain AWS cloud, AWS services, cloud security and containerization using docker.
- To demonstrate skills in designing and implementing modern cloud applications.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] HISTORY OF DISTRIBUTED COMPUTING AND INTRODUCTION TO CLOUD COMPUTING

History of distributed computing:Cluster Computing, Grid Computing, Service Oriented Computing (Web Services). Introduction to Cloud Computing: Emergence of Cloud from Web 2.0, Virtualization, Single Sign-On, Grid Computing, Cluster, Automation, Utility Computing NIST definition of Cloud Computing, Cloud Characteristics, Various types of Virtualizations, Benefits and drawbacks of Cloud Computing, Cloud Services (Iaas, PaaS, SaaS), Cloud Deployment Models (Public, Private, Hybrid, Community), Cloud Service Providers.

### [2] VIRTUALIZATION

Compute/CPU Virtualization: Types of instructions, Popek and Goldberg virtualization requirements, Virtualizable CPU, Non-Virtualizable CPU, Virtual Machine Monitor (VMM) and types of hypervisors: hosted and bare metal, Virtual Machine, Specification, and representation, Types of Virtualization: ParaVirtualization, Full Virtualization (Binary Translation), Full Virtualization (Hardware Assisted Virtualization), Introduction to Hypervisors: xen and kvm, Storage Virtualization: Architecture of Cloud Storage, Advantages of Cloud Storage over Traditional Storage, Block Level Storage Virtualization and File Level Storage Virtualization, Introduction to Storage Area Network (SAN) and Network Attached

Storage (NAS), Block Storage and Object Storage, Network Virtualization: Virtualization of NIC, VM Network and Virtual Firewall, Approaches of Network Virtualization: TAP/TUN+Bridge and Virtual Switch, Virtual Networking Modes and VLAN.

### [3] VM LIFECYCLE AND MANAGEMENT

VM Lifecycle, VM Configuration, VM Provisioning, VM Migration, types, and Live VM Migration, Scaling: Horizontal Scaling, Vertical Scaling, Server Consolidation.

### [4] OPEN SOURCE CLOUD IMPLEMENTATION AND ADMINISTRATION

Introduction to Open Source software for Private Cloud: Eucalyptus: architecture, major components, and services, Open Stack : architecture, major components, and services, Using euca2ools.

### [5] SECURITY IN CLOUD

Security Architecture Design, Identity and access management architecture, Isolation of users/VMs from each other, Data Security, Application Security, Virtual Machine Security, Virtualization System‐Specific Attacks: Guest hopping, attacks on the VM (delete the VM, attack on the control of the VM, code or file injection into the virtualized file structure), VM migration attack, hyperjacking.

### [6] CONTAINERIZATION

Virtual Machine vs Containers, Benefits of Containers, Use cases for Containers, Container orchestration, Docker.

### [7] CASE STUDY: AMAZON WEB SERVICES (AWS)

AWS Infrastructure and Services:Foundation Services (Compute, Networking, and Storage), Introduction to Platform Services, Using Amazon Web Services, Use the services using GUI, Use the services using CLI, Compute Service: Elastic Compute Cloud (EC2), Elastic Compute Cloud Storage Service, Other Services: CloudWatch, Elastic Load Balancer, Simple Notification Service, CloudFront, Security.

### [8] SERVERLESS COMPUTING

Introduction to Serverless Computing: Definition, characteristics, FaaS (Function as a Service), Architecture of Serverless Computing, Use cases for Serverless, Serverless Computing support by major Cloud Providers, Programming Model, Tools, and Frameworks.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Cloud Computing Bible, Barrie Sosinsky, Publisher: Wiley, 2011
2. Mastering Cloud Computing, Foundations and Applications Programming, Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, Publisher: Elsevier Inc

3. Mastering KVM Virtualization, Humble Devassy Chirammal, Prasad Mukhedkar, Anil Vettathu, Second edition, Publisher: PACKT, October 2020
4. Beginning Serverless Computing, Maddie Stigler, Publisher: APress, 2018
5. Virtualization From the Desktop to the Enterprise,Chris Wolf, Erick M. Halter, Publisher: APress, 2005
6. OpenStack Cloud Computing Cookbook, Kevin Jackson, Cody Bunch, Egle Sigler, Publisher: PACKT, fourth edition, 2018

## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember the concepts of cloud computing and virtualization. | Remember |
| CO2 | Be able to Understand architectures of private cloud and public cloud. | Understand |
| CO3 | Be able to Apply cloud computing concepts for AWS services. | Apply |
| CO4 | Be able to Analyze the given problem and use various cloud services. | Analyze |
| CO5 | Be able to evaluate one or more cloud solutions to solve the given problem requirements and be able to decide a choice under the given context clearly. | Evaluate |
| CO6 | Be able to Design and create a cloud based solution using the concepts of cloud computing. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 2 | 3 |
| CO2 | 3 | 2 | 1 | 1 | 0 | 0 | 3 | 0 | 0 | 1 | 0 | 3 | 3 | 2 | 3 |
| CO3 | 2 | 3 | 2 | 1 | 1 | 2 | 3 | 0 | 0 | 1 | 0 | 1 | 3 | 2 | 3 |
| CO4 | 2 | 3 | 2 | 2 | 3 | 2 | 2 | 0 | 0 | 2 | 0 | 1 | 3 | 2 | 3 |
| CO5 | 2 | 3 | 2 | 2 | 3 | 2 | 2 | 0 | 0 | 2 | 0 | 1 | 3 | 2 | 3 |
| CO6 | 0 | 2 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 3 | 2 | 3 |
| Avg. | 1.83 | 2.33 | 1.33 | 1.33 | 1.33 | 1 | 1.67 | 0 | 0 | 1.17 | 0 | 1.17 | 3 | 2 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Study and Installation of Virtualization Software (e.g., Oracle Virtual Box) and study of networking modes (Bridge Networking, Host Networking, Private Networking).
2. Installation of KVM and use of libvirt tool.
3. Study of Logical Volume Manager.
4. Study and installation of Storage as Service cloud computing environment (using FreeNAS/ownCloud).
5. Private Cloud Administration (Eucalyptus or OpenStack: IP Address configuration, Image Management, Security groups, Firewall, instance creation, instance access).
6. Usage of Public Cloud: Authentication, VM Provisioning, and other features.
7. Web Application Development and Testing in IAAS cloud environment (using putty, filezilla, SCP).
8. Study of Public cloud services.
9. Docker installation and orchestration application scenario.
10. Serverless computing on a public cloud (any one of Amazon AWS/GCP/Microsoft Azure).

## B. TECH. SEMESTER – VI (IT)

## SUBJECT: PROJECT-I

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 0 | 0 | 2 | 2 | 1 | 0 | 0 | 25 | 25 | 50 |

Reference Code PROJ-CS601

## A. COURSE OBJECTIVES

- To allocate a project in a group of minimum two students (the allocated project is desirable to be unique among all the other projects of the peer students of BTech IT programme)
- To guide students for various activities of software development such as analysis, design, implementation, and testing.
- To carry out regular reporting and monitor the progress of the project.
- To evaluate the developed project and the project report.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] REQUIREMENTS GATHERING AND PROJECT MANAGEMENT

Explore and evaluate requirements, perform market survey of similar app/functionalities, perform technical feasibility by implementing a mini task.

### [2] ANALYSIS

Analyse functional and non-functional requirements, prepare usecase diagram, prepare SRS document, prepare wireframes/mockup screens with business flow.

### [3] DESIGN

Prepare architecture diagram, class diagram, sequence diagram, database design, component diagram, and deployment diagram. Also prepare statechart and activity diagrams, if required.

### [4] IMPLEMENTATION

Implement prepared design using chosen development platform, IDE, and programming languages.

### [5] TESTING

Perform UI testing and integration testing.

### [6] REPORT WRITING

Write a project report in the standard format.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Roger S. Pressman, Software Engineering-A Practitioner's Approach, McGrawHill, 2019.
2. Robert C. Martin, Clean Code-A Handbook of Agile Software Craftsmanship, Prentice Hall, 2009.


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to explore and evaluate requirements of the project/system to be built in addition to feasibility study. | Evaluate |
| CO2 | Be able to analyse the functional and non-functional requirements of the project/system and be able to specify them in form of Use case diagram, System Requirements Specification, Hardware/Software Requirements, and User Interface diagram using appropriate UML tools. | Analyze |
| CO3 | Be able to prepare design and architecture of the system using sequence diagram, class diagram, component diagram, and deployment diagram using appropriate UML tools. | Create |
| CO4 | Be able to implement the solution using chosen development platform, language, and IDEs. | Create |
| CO5 | Be able to test, troubleshoot, and record test cases for implemented functionality, as a standalone unit and in integration with other functionalities. | Analyze |
| CO6 | Be able to work with an approach of life-long learning for various phases of software development lifecycle such as feasibility study, project planning, requirements analysis and specification, system design, implementation, unit testing, integration testing, and report writing. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 0 | 1 | 3 | 3 | 3 |
| **CO2** | 3 | 2 | 3 | 1 | 0 | 2 | 1 | 3 | 3 | 1 | 3 | 1 | 3 | 3 | 3 |
| **CO3** | 2 | 3 | 2 | 1 | 1 | 2 | 1 | 3 | 3 | 1 | 0 | 1 | 3 | 3 | 3 |
| **CO4** | 2 | 3 | 3 | 2 | 3 | 2 | 2 | 3 | 3 | 2 | 0 | 1 | 3 | 3 | 3 |
| **CO5** | 2 | 2 | 1 | 3 | 2 | 3 | 2 | 3 | 3 | 2 | 3 | 1 | 3 | 3 | 3 |
| **CO6** | 1 | 2 | 3 | 3 | 3 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| **Avg.** | 2 | 2.17 | 2 | 1.67 | 1.5 | 1.83 | 1.33 | 3 | 3 | 1.5 | 1.5 | 1.33 | 3 | 3 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## B. TECH. SEMESTER – VII (IT)

## SUBJECT: CONSTITUTION OF INDIA

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Reference Code MC

## A. COURSE OBJECTIVES

- To provide information about the Constitution of India.
- To educate the students about the fundamental Rights and Duties as a Citizen of India.
- To impart knowledge about the various functionaries associated with the Governance in the country.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] INDIAN CONSTITUTION: INTRODUCTION

Meaning of the Constitution, History, Preamble and Basic Structure, Salient Features and Characteristics.

### [2] CITIZENSHIP

Meaning, Constitutional Rights and Privileges of Citizens of India, Loss of Indian Citizenship, One Citizenship in India.

### [3] FUNDAMENTAL RIGHTS AND DUTIES

Definition, Right to Equality, Right to Freedom, Right Against Exploitation, Right to Freedom of Religion, Cultural and Educational Rights, Right to Constitutional Remedies, Scope of Right to Life and Personal Liberty, Rights following from other provisions of the Constitution, Difference between Fundamental Rights and Rights secured by other provisions of the Constitution, Fundamental Duties.

### [4] STATE POLICY

Directive Principles of State Policy, Classification and its Scope, Directives compared with Fundamental Rights, Implementation of the Directives.

### [5] ORGANS OF GOVERNANCE

Legislature: Parliament and its Composition, Qualifications, and Disqualifications of its members; Executive: President, Governor, and Council of Ministers; Judiciary: Appointments, Qualifications, Powers and Functions of Judges; Local Administrations: District, Municipal and Zila Panchayat

### [6] ELECTION COMMISSION

Role and Functioning of Election Commission, Chief Election Commissioner, State Election Commissioner.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Durga Das Basu, Introduction to the Constitution of India, Prentice Hall of India, New Delhi
2. R. C. Agarwal, Indian Political system, S. Chand and Company, New Delhi
3. Constitution of India, Government Publication
4. Sharma, Brij Kishore, Introduction to the Constitution of India, Prentice Hall of India, New Delhi
5. M. P. Jain, Indian Constitutional Law, 7th Edition, Lexis Nexis


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | To define and describe the Indian Constitution. | Remember |
| CO2 | To understand the basic functionality of various cadres under the Indian Constitution. | Understand |
| CO3 | To articulate rights and responsibilities during various situations. | Apply |
| CO4 | To Analyze the various situations based on demands and connect to appropriate authorities for solving the societal problems. | Analyze |
| CO5 | To appraise and criticize various solutions suggested by the Indian Constitution. | Evaluate |
| CO6 | To facilitate Indian Citizens about their Rights and Duties. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 3 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| CO2 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 3 | 1 | 2 | 0 | 0 | 0 | 0 | 0 |
| CO3 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 3 | 2 | 2 | 0 | 1 | 0 | 0 | 0 |
| CO4 | 0 | 0 | 0 | 0 | 0 | 3 | 1 | 3 | 1 | 2 | 0 | 2 | 0 | 0 | 0 |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO5** | **0** | **0** | **0** | **0** | **0** | **3** | **1** | **3** | **2** | **1** | **0** | **1** | **0** | **0** | **0** |
| **CO6** | **0** | **0** | **0** | **0** | **0** | **3** | **0** | **3** | **1** | **1** | **0** | **1** | **0** | **0** | **0** |
| **Avg.** | **0** | **0** | **0** | **0** | **0** | **2.5** | **0.33** | **3** | **1.33** | **1.5** | **0** | **0.83** | **0** | **0** | **0** |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## B. TECH. SEMESTER – VII (IT)

## SUBJECT: MACHINE LEARNING & DEEP LEARNING (ELECTIVE-4)

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PEC

## A. COURSE OBJECTIVES

- To present the fundamental concepts of machine learning - supervised, unsupervised, and reinforcement learning, along with the key algorithms.
- To provide a detailed understanding of data preprocessing, feature engineering, model training, and model evaluation techniques.
- To discuss various deep learning architectures like CNNs, autoencoders, and RNNs
- To develop problem-solving abilities by applying the knowledge of machine learning and deep learning through projects.
- To impart knowledge about recent libraries and tools for implementing machine learning and deep learning models.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] BASICS OF MACHINE LEARNING

Definition of ML, Types of ML (Supervised, Unsupervised, Reinforcement, etc.), Applications of ML, Machine Learning Design Cycle (Pipeline), Classification (Decision Tree Classification, Naïve Bayes, Support Vector Machines), Ensemble Learning and Random Forests (Bagging, Boosting), Linear Regression, Polynomial Regression, Logistic Regression, Clustering.

### [2] NEURAL NETWORKS

Introduction to Neural Networks, Single Layer Perceptron Model, Learning Rule, Gradient Decent Training, Multilayer Perceptron, Backpropagation, Issues with Backpropagation Learning, Avoiding Overfitting Through Regularization, Dimensionality Reduction, Curse of Dimensionality Problem, Principal Component Analysis (PCA).

### [3] CONVOLUTIONAL NEURAL NETWORKS

Introduction Deep learning and different types of Deep network architectures, Activation functions, and loss functions, Convolutional Neural Networks, Building Blocks of CNN (Convolution Layer, Pooling layer, dropout layer, Batch Normalization Layer, Fully Connected Layer), State-of-the-art CNNs, Transfer Learning, Residual Networks, Skip Connection Networks.

### [4] COMPUTER VISION

Image Classification, Image classification with localization, Semantic Segmentation, Object Detection, Siamese Network, Auto Encoders and Stacked Auto Encoders.

### [5] SEQUENCE MODELS

Recurrent Neural Network (RNN), Long-short Term Memory Network, Gated Recurrent Unit, Bi-directional RNN, Deep RNN

### [6] GENERATIVE AI

Generative AI, Large Language Models (LLM), Text Generation Models, Image Generation Models, Video Generation Models, Prompt Engineering, AI Tools for business applications.

### [7] ADVANCED TOPICS

Explainable AI, MLOps, Ethical & Social Considerations in ML

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Tom Mitchell, "Machine Learning", McGraw Hill
2. Michael A. Nielsen, "Neural Network and Deep Learning" Determination Press (http://neuralnetworksanddeeplearning.com)
3. Kevin P. Murphy, "Machine Learning: A Probabilistic Perspective", MIT Press
4. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press (https://www.deeplearningbook.org/)
5. C. Bishop, "Pattern Recognition and Machine Learning", Springer.
6. Richard O. Duda, Peter E. Hart, David G. Stork, "Pattern Classification", John Willey & Sons
7. Selected Research Papers


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | To represent the essential concepts, resources, approaches, and Python libraries for machine learning and deep learning with actual data sets. | Remember |
| CO2 | To understand the mathematical foundation required for machine learning algorithms. | Understand |
| CO3 | To apply various concepts of Neural Networks for various computer vision and natural language processing tasks. | Apply |

| | | |
|---|---|---|
| CO4 | Compare a range of various machine learning and deep learning algorithms along with their strengths & weaknesses and choose appropriate techniques to solve problems of moderate complexity. | Analyze |
| CO5 | Evaluate the performance of the machine learning and deep learning models | Evaluate |
| CO6 | Construct AI-based solutions for real-world problems, fine-tune the trained model, and evaluate the performance. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 |
| **CO2** | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 |
| **CO3** | 3 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 |
| **CO4** | 1 | 2 | 1 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 |
| **CO5** | 2 | 2 | 1 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 |
| **CO6** | 2 | 2 | 2 | 3 | 3 | 3 | 2 | 3 | 3 | 3 | 3 | 2 | 3 | 3 | 3 |
| **Avg.** | 2 | 1.5 | 0.83 | 1.17 | 0.5 | 1.17 | 0.33 | 0.5 | 0.5 | 0.5 | 0.5 | 0.33 | 3 | 3 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Introduction to Python, Numpy, Sklearn,
2. Implement Linear and Logistic Regression with Error Correction-based learning using Python Numpy
3. Introduction to Keras and TensorFlow - Classification
4. Implementation of SVM
5. Practicals related to Dimensionality reduction
6. Implement Single-layer and Multilayer Perceptron with Backpropagation using Keras and TensorFlow
7. Implement Convolutional Neural Network using Keras and TensorFlow
8. Implementation of Transfer Learning (VGG16, VGG19)
9. Implement Recurrent Neural Network/LSTM using Keras and TensorFlow
10. Implement Auto-encoders using Keras and TensorFlow
11. Introduction to GridWord Problem and Implement Temporal Difference Algorithm to solve GridWord problem using Reinforcement learning
12. Project Based on Machine Learning / Deep Learning

## B. TECH. SEMESTER – VII (IT)

## SUBJECT: DIGITAL IMAGE PROCESSING (ELECTIVE-4)

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PEC

## A. COURSE OBJECTIVES

- To make students familiar with digital image fundamentals.
- To explain simple image enhancement techniques in spatial and frequency domain.
- To teach the concepts of degradation function and restoration techniques.
- To explain image segmentation and representation techniques.
- To explain about image compression and recognition methods.
- To discuss about colour and morphological Image processing.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] INTRODUCTION

Fundamentals, Applications of Image Processing: Medical Imaging, Robot Vision, Character Recognition, Remote Sensing, Image Processing System Components, Image Sensing and Acquisition, Sampling and Quantization, relationships between pixels, Regions and Boundaries, Distance measures.

### [2] IMAGE ENHANCEMENT

Enhancement In The Spatial Domain: Gray Level Transformations, Histogram Processing, Enhancement Using Arithmetic/Logic Operations, Basics Types of spatial filters.

Enhancement In Frequency Domain:Introduction, Fourier Transform, Discrete Fourier Transform (DFT), properties of DFT , Discrete Cosine Transform (DCT), Image filtering in frequency domain.

### [3] IMAGE RESTORATION

Degradation models and properties, Noise models, Mean Filters, Order Statistics, Adaptive filters, Band reject Filters, Band pass Filters, Notch Filters, Optimum Notch Filtering, Inverse Filtering, Wiener filtering.

### [4] IMAGE SEGMENTATION

Thresholding methods, Edge detection Techniques(e.g. Sobel,Canny), Region based segmentation algorithms(e.g. Watershed, Region Growing)

### [5] WAVELETS AND MULTI - RESOLUTION PROCESSING

Image Pyramid, Subband Coding, Harr Transform, Multi resolution expression, Wavelet transform.

### [6] IMAGE COMPRESSION

Fundamentals, models, lossless and lossy compression, compression standards.

### [7] ADVANCED TOPICS IN IMAGE PROCESSING

Deep learning for Image Processing(e.g. Convolutional Neural Networks), Generative Adversarial Networks(GANs) for image generation and manipulation, and Transfer learning for image analysis tasks. Ethical & Social considerations for Image Processing.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Rafel C. Gonzalez and Richard E. Woods, Digital Image Processing, Fourth Edition, Pearson Education, 2018
2. Bhabatosh Chanda and Dwijesh Majumder, Digital Image Processing and Analysis, Second Edition, 1977
3. S Jayaraman and S Essakirajan, Digital Image Processing, Second Edition, Mc Graw Hill, 2020


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember different types of transformations that can be applied to images. | Remember |
| CO2 | Be able to understand basic concepts of image processing and image storage. | Understand |
| CO3 | Be able to apply different Image Restoration & Enhancement techniques for a given problem. | Apply |
| CO4 | Be able to analyze the correctness of algorithms using inductive proofs and loop invariants. | Analyze |
| CO5 | Be able to evaluate and compare the domains and methods of image processing. | Evaluate |
| CO6 | Be able to design an algorithm by making proper use of different image processing tools. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 2 | 3 |
| **CO2** | 2 | 2 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 3 | 2 | 3 |
| **CO3** | 2 | 3 | 1 | 2 | 1 | 2 | 0 | 0 | 0 | 1 | 0 | 1 | 3 | 2 | 3 |
| **CO4** | 2 | 3 | 3 | 3 | 3 | 2 | 0 | 0 | 0 | 2 | 0 | 1 | 3 | 2 | 3 |
| **CO5** | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 2 | 0 | 1 | 3 | 2 | 3 |
| **CO6** | 2 | 1 | 1 | 2 | 1 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 3 | 2 | 3 |
| **Avg.** | 2 | 2 | 1.33 | 1.83 | 1.5 | 1 | 0.17 | 0 | 0.5 | 1 | 0 | 0.83 | 3 | 2 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Image Printing Program Based on Halftoning.
2. Reducing the Number of Intensity Levels in an Image.
3. Zooming and Shrinking Images by Pixel Replication.
4. Zooming and Shrinking Images by Bilinear Interpolation.
5. Arithmetic Operations on images.
6. Image Enhancement Using Intensity Transformations.
7. Histogram Equalization.
8. Spatial Filtering.
9. Enhancement Using the Laplacian.
10. Unsharp Masking.

## B. TECH. SEMESTER – VII (IT)

## SUBJECT: DEVOPS

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 0 | 0 | 2 | 2 | 1 | 0 | 0 | 25 | 25 | 50 |

Reference Code PCC

## A. COURSE OBJECTIVES

- To describe the Modern software engineering agile relationship between development and IT operations.
- To introduce the concept of Automation and understanding the code quality.
- To demonstrate the usage of Continuous delivery and deployment management.
- To demonstrate automated system update and DevOps lifecycle model.
- To introduce the terminology, technology and its applications.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] DEVELOPMENT OPERATIONS (DEVOPS)

DevOps principles, Life cycle, pipelines, tools.

### [2] VERSION CONTROLLING

Version control, Central vs Distributed Version Controlling, Git Workflow, Installing Git, Important Git Commands, Remote repository, GitHub, and operations: clone, fork, push, pull, pull/merge request.

### [3] CONTINUOUS INTEGRATION AND CONTINUOUS DEPLOYMENT (CI/CD)

Jenkins Architecture, Setting up the CI/CD server pipeline automatically.

### [4] CONTAINERIZATION AND ORCHESTRATION

Docker Architecture and Components, Contanerizing app using Docker Containers, Kubernetes Architecture and Components, Container orchestration using Kubernetes.

### [5] INFRASTRUCTURE AS CODE (IAC)

Ansible Automation tool for provisioning, configuring, deploying, and managing applications.

### [6] MONITORING & OBSERVABILITY

Monitoring tools, logging practices, alerting, and notifications, log monitoring and analysis using graphing tool.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Sanjeev Sharma and Bernie Coyne, DevOps for dummies, 2nd Edition, John Wiley & Sons, Inc., 2015
2. Marc Hornbeek, Engineering DevOps,


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to identify components of Devops environment and its life-cycle model. | Remember |
| CO2 | Be able to understand the DevOps life-cycle model, Concepts and Tools and Deployment of DevOps tools. | Understand |
| CO3 | Be able to apply different concepts of Devops like version controlling, source code management, integration, testing, project management and code deployment using DevOps tools. | Apply |
| CO4 | Be able to assess various Devops techniques, tools and practices. | Analyze |
| CO5 | Investigate different DevOps Software development models, tools and APIs. | Evaluate |
| CO6 | To implement automated system update and DevOps lifecycle, Collaborate and adopt Devops in real-time projects | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 2 | 1 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 2 | 3 |
| CO2 | 2 | 1 | 0 | 0 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 3 | 3 | 2 | 3 |
| CO3 | 2 | 2 | 1 | 0 | 3 | 0 | 1 | 0 | 0 | 1 | 2 | 3 | 3 | 2 | 3 |
| CO4 | 2 | 2 | 2 | 2 | 3 | 0 | 2 | 0 | 0 | 0 | 3 | 3 | 3 | 2 | 3 |
| CO5 | 2 | 3 | 2 | 2 | 3 | 2 | 2 | 0 | 0 | 0 | 3 | 3 | 3 | 2 | 3 |
| CO6 | 2 | 3 | 3 | 1 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 2 | 3 |
| Avg. | 2 | 2 | 1.33 | 0.83 | 3 | 0.83 | 1.5 | 0.5 | 0.5 | 0.67 | 1.83 | 2.83 | 3 | 2 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Introduction to Development Operations, Key concept of Automation and CD/CI, and its relevance to the modern software engineering practices like agile and lean.
2. Source code Version Controlling, Central vs Distributed Version Controlling, Git Workflow, Installing Git, and Important Git Commands.
3. Managing Source Code on GitHub using Git.
4. Installation of Docker Desktop and Docker and use of existing images from Docker Hub for creating containers.
5. Containerizing application using Docker Container.
6. Orchestration of application components/microservices using Kubernetes and Docker containers.
7. Jenkins architecture, installation, configuration, and pipelining primitives.
8. Creating Jenkins Pipelines and triggering it based on events.
9. Installation and use of Ansible automation tool for provisioning, configuring, deploying, and managing applications/resources.
10. Continuous Monitoring using tools such as Prometheus and Grafana.

**Additional Experiment**

11. AWS DevOps Services (Code-commit, Code-build, Code-deploy, Code-pipeline, Cloud Formation and Terraform)

## B. TECH. SEMESTER – VII (IT)

## SUBJECT: ARTIFICIAL INTELLIGENCE (OPEN ELECTIVE-3)

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 3 | 0 | 0 | 3 | 3 | 60 | 0 | 0 | 0 | 60 |

Reference Code OEC

## A. COURSE OBJECTIVES

- To make students aware of the concepts of state space representation, exhaustive search, heuristic search, logic representation, uncertainty handling, Markov decision theory, and reinforcement learning model.
- To make students understand terms and notations related to search, knowledge representation, uncertainty handling, Markov decision theory, and reinforcement learning model.
- To make students capable of applying different appropriate search, knowledge representation, and uncertainty handling techniques to solve the given problem.
- To make students capable of analyzing and selecting the most appropriate techniques to solve AI problems.
- To make students capable of evaluating various AI techniques.
- To make students capable of creating AI-based solutions.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] INTRODUCTION

What Is AI? Definitions of AI- Thinking vs. Acting vs. Humanly vs. Rationally, Phases of AI, History and current status of AI.

### [2] PROBLEM-SOLVING BY BASIC SEARCH TECHNIQUES

The notion of state, problem space, Search algorithm terminologies, properties of search algorithms, and types of search algorithms. Algorithms: Breadth-first Search, Depth-first Search, Depth-limited Search, Iterative deepening depth-first search, Uniform cost search, Bidirectional Search.

### [3] PROBLEM-SOLVING BY INFORMED SEARCH TECHNIQUES

Heuristic Search Algorithms: Best first search, A* algorithm, Analysis of A* Admissible Heuristic, IDA*.

### [4] PROBLEM-SOLVING BY LOCAL SEARCH TECHNIQUES

Hill climbing, issues and solutions of hill climbing, simulated annealing, Genetic algorithm.

### [5] ADVERSARIAL SEARCH

Introduction, Types of Games, game example search space, Minimax Algorithm, Alpha-Beta Pruning.

### [6] LOGIC AND INFERENCE

Types of logic- PL, FOL, Probabilistic FOL, Fuzzy, etc; Concepts of expert systems, CNF, sound, complete, syntax, semantics, inference by rules, forward Chaining, and resolution.

### [7] UNCERTAINTY HANDLING

Uncertainty in AI-motivation, conditional independence, and Bayes rule, bayesian networks syntax, inference using variable elimination.

### [8] MARKOV DECISION PROCESS

Definition, example of policy, policy iteration, policy evaluation, applications, and extensions of MDP

### [9] REINFORCEMENT LEARNING

Reinforcement Learning -background concepts, model-based learning for policy evaluation, Q learning

### [10] ETHICS OF AI

Transparency, accountability, the privacy of AI systems, AI Governance, Fairness, and data bias.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Stuart Russell, Peter Norvig,"Artificial Intelligence: A Modern Approach", 3rd edition, PrenticeHall,2009.
2. Rich & Knight, "Artificial Intelligence, second edition",Tata Mc GrawHill.
3. Nils Nilsson, "Artificial Intelligence: A New Synthesis", Morgan Kaufmann, 1998.
4. David Poole, Alan Mackworth, "Artificial Intelligence: Foundations for Computational Agents", Cambridge Univ. Press, 2010.
5. Ian GoodFellow, Yoshua Bengio & Aaron Courville, "Deep Learning", MIT Press ,2016.


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Able to remember the concepts of state space representation, exhaustive search, heuristic search, logic representation, uncertainty handling, markov decision theory, and reinforcement learning model. | Remember |

| | | |
|---|---|---|
| CO2 | Able to understand terms and notations related to search, knowledge representation, uncertainty handling, Markov decision theory, and reinforcement learning model. | Understand |
| CO3 | Able to apply appropriate search and/or knowledge representation and uncertainty-handling techniques to solve the given problem. | Apply |
| CO4 | Able to analyze and select the most appropriate techniques to solve AI problems. | Analyze |
| CO5 | Able to evaluate various AI techniques. | Evaluate |
| CO6 | Able to create AI-based solutions. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 0 | 2 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 2 | 1 |
| **CO2** | 2 | 2 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 2 | 1 |
| **CO3** | 2 | 2 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 2 | 1 |
| **CO4** | 2 | 2 | 1 | 2 | 0 | 0 | 0 | 3 | 0 | 1 | 0 | 2 | 0 | 2 | 1 |
| **CO5** | 2 | 2 | 1 | 2 | 0 | 0 | 0 | 3 | 0 | 1 | 0 | 2 | 0 | 2 | 1 |
| **CO6** | 2 | 2 | 1 | 2 | 0 | 0 | 0 | 3 | 0 | 1 | 0 | 2 | 0 | 2 | 1 |
| **Avg.** | 1.67 | 2 | 1 | 2 | 0 | 0 | 0 | 1.5 | 0 | 1 | 0 | 2 | 0 | 2 | 1 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## B. TECH. SEMESTER – VII (IT)

## SUBJECT: DISTRIBUTED COMPUTING (ELECTIVE-5)

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code: PEC

## A. COURSE OBJECTIVES

- To teach fundamental knowledge of distributed computing paradigm
- To explain how to use TCP/UDP protocols for message communication between client-server
- To discuss concepts of microservices and its integration to enterprise applications.
- To impart knowledge of containerization and container orchestration.
- To provide insights for data intensive computing via apache hadoop and apache spark.
- To demonstrate skills to write, debug, deploy and execute distributed application programs and enable them to create enterprise solutions using socket programming, microservices, containerization, cloud computing etc.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] FUNDAMENTALS OF DISTRIBUTED COMPUTING

Definitions, Introduction of Distributed Computing (DC), Different forms of computing, The strengths and weaknesses of DC, Distributed Computing Paradigms and Abstraction, Paradigms for Distributed Applications, Trade-offs .

### [2] SOCKET PROGRAMMING ON UNIX

Basics of Various Networking Protocols - TCP, UDP, Socket API, Fundamentals of client-server communication using TCP, Socket Programming in UNIX, client-server communication using UDP, Elementary Socket Options.

### [3] gRPC

gRPC Concepts, gRPC Architecture, Application Development using gRPC.

### [4] CONTAINERIZATION - DOCKER

Docker Overview, Architecture, Installation and Setup, Docker Commands, Working with Containers, Docker Images, Docker Compose, Docker Engine, Storage, Docker Registry/Hub, Containers and hosts, Inspect active containers, Docker Networking, Persist Data with Docker Volumes, Dockerize apps using Docker storage, Docker-Compose, Work with Docker Repository.

## [5] CONTAINER ORCHESTRATION - KUBERNETES

Kubernetes Overview, Architecture, Installation and Setup, Application Deployment, Administration and Management of the K8s cluster, Core Components: Service, ConfigMap, Secret, StatefulSet, Ingress Kubernetes CLI (Kubectl), Persisting data with K8s Volumes, Namespaces.

## [6] MICROSERVICES, APIs AND INTEGRATION

Microservice Overview, Microservices Architecture Components and Functions, Microservice vs Monolithic Architecture, Microservices Architecture and Design Characteristics, Challenges of Microservices Architecture, Microservices and APIs, The role of microservices in DevOps, Key steps to Deploy Microservices, Orchestration of microservices, Tools to Deployment and Management of Microservices. Implementation of Microservices.

## [7] AUTHENTICATION AND AUTHORIZATION PROTOCOLS FOR DISTRIBUTED ENVIRONMENT

Authentication and Authorization: OAuth, OpenID Connect, SAML.

## [8] DISTRIBUTED SYSTEM PROCESSING USING APACHE KAFKA

Asynchronous communication, Fundamentals of Apache Kafka, Kafka clusters.

## [9] DATA INTENSIVE COMPUTING - (APACHE HADOOP AND APACHE SPARK).

Big Data Analytics, Hadoop Architecture, Hadoop Cluster Setup and Installation, HDFS, Map-Reduce, Word Count in Hadoop and HDFS, YARN, Apache Spark-Introduction to Data-intensive computing with Spark, In-Memory Computing with Spark.

## [10] ORCHESTRATION OF MICROSERVICES

Orchestration, Microservice orchestration, Benefits of Step Functions, Working of step functions, Use cases, AWS step functions: orchestration of microservices through AWS cloud.

## [11] CASE STUDIES: FACEBOOK/GOOGLE/NETFLIX ARCHITECTURE

Facebook, Google, Netflix - Netflix Technology Stack: AWS, Apache Kafka, Apache Chukwa, Apache Cassandra, Hadoop, CDN (Content Delivery Network), Batch Processing, Microservices Architecture.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. W. Richard Stevens, UNIX Network Programming, Publisher: Prentice Hall Publication
2. M. L. Liu, Distributed Computing: Concepts & Applications, Publisher: Addison Wiselly.
3. Raja Malleswara Rao Pattamsetti, Distributed Computing in Java 9, Publisher: Packt Publishing Limited.
4. Pradeep K. Sinha, Distributed Operating Systems: Concepts and Design, Publisher: PHI Publication.
5. George Coulouris, Jean Dollimore and Tim Kindberg, Distributed Systems: Concepts and Design, 4th Ed, Publisher: Addison Wesley.
6. John Wiley & Sons, The DevOps adoption playbook: a guide to adopting DevOps in a multi-speed IT enterprise.Sharma, S. (2017).


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember distributed computing concepts such as socket programming, distributed architecture, cloud, and microservice. | Remember |
| CO2 | Be able to understand distributed computing concepts such as sockets, remote method invocation, design, and architecture of distributed computing and cloud computing. | Understand |
| CO3 | Be able to apply the concepts of distributed technologies to solve the given problem in the distributed domain. | Apply |
| CO4 | Be able to analyze the given problem and provide the implementation. | Analyze |
| CO5 | Be able to evaluate one or more options to solve the given problem or program requirements and be able to decide a choice under the given context. | Evaluate |
| CO6 | Be able to create a solution for distributed applications for given problems through Socket programming, client-server programming, microservices, big data processing, containerization, etc. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 2 | 3 |
| **CO2** | 3 | 2 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 3 | 2 | 3 |
| **CO3** | 2 | 3 | 1 | 1 | 1 | 2 | 1 | 0 | 0 | 1 | 0 | 1 | 3 | 2 | 3 |
| **CO4** | 2 | 3 | 3 | 2 | 3 | 2 | 2 | 0 | 0 | 2 | 0 | 1 | 3 | 2 | 3 |
| **CO5** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 2 | 0 | 1 | 3 | 2 | 3 |
| **CO6** | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 2 | 0 | 1 | 3 | 2 | 3 |
| **Avg.** | **2** | **2.17** | **1.5** | **1.33** | **1.33** | **1.33** | **1.33** | **0** | **0** | **1.33** | **0** | **1** | **3** | **2** | **3** |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Unix socket programming: TCP-UDP client-server Program.
2. Configure the following options on the server socket and test them: SO_KEEPALIVE, SO_LINGER, SO_SNDBUF, SO_RCVBUF, TCP_NODELAY
3. gRPC based application development.
4. Containerization and orchestration using Docker and Kubernetes.
5. Create microservice-based application using Spring Boot.
6. Microservice orchestration using orchestration framework/service.
7. Asynchronous Communication using Apache Kafka.
8. Deployment of Apache Hadoop and monitoring the status of various Hadoop components.
9. Perform data-intensive computing using map-reduce based programming on a HADOOP cluster.
10. Authentication and authorization: OAuth, OpenID connect, SAML.
11. Case Study on different architecture designs: Netflix, Google Search, Facebook.

## B. TECH. SEMESTER – VII (IT)

## SUBJECT: ADVANCED OPERATING SYSTEM (ELECTIVE-5)

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PEC

## A. COURSE OBJECTIVES

- To present various OS structures and discuss various types of OSes.
- To provide a detailed understanding of process, control, interrupt, exception handling and signal handling.
- To exemplify multiprocessor systems, real-time systems, and scheduling in both types of systems.
- To discuss multithreading and demonstrate multicore programming using OMP.
- To show various memory management techniques and discuss data structures and related algorithms.
- To elaborate file-system interface and implementation and introduce to IO subsystem with major focus on Linux device drivers and modules.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] INTRODUCTION TO OS, TYPES OF OS, OS STRUCTURES

OS, Objective of OS, Kernel, Evolution of OS, Types of OS: Serial Processing, Batch Processing, Mainframe System, Desktop Systems, Multiprocessor Systems, Distributed Systems, Clustered Systems, Real-time Systems, and Handheld Systems. OS organization, components of OS, OS services, System calls, Mechanism vs Policy in OS design and implementation, OS structures: Simple, Layered, Microkernel, Virtual Machine,

### [2] INTERRUPT, EXCEPTION, AND SIGNAL HANDLING

Interrupts, interrupt vs exception, hardware device and interaction with CPU, System call, system call path, System call stub function, trap instruction, passing parameters to system call, adding a new system call, signal mechanism in Unix, process group and sending signal to process group,

### [3] MULTIPROCESSOR SYSTEMS

Multiprocessor system and Amdahl's Law, Classifying Multiprocessor system: classification based on Flynn's taxonomy, based on sharing, based on interconnection scheme, Multiprocessor Operating System: Private OS, Master-slave, Symmetric

Multiprocessor, Multiprocessor Synchronization: mutual exclusion, hardware instructions, spin locks, variations of spin locks, sleep/wakeup locks, Read/write locks,

## [4] MULTIPROCESSOR AND REAL-TIME SCHEDULING

Process Scheduling on uniprocessor vs multiprocessor, time sharing and space sharing, multilevel queue scheduling, process scheduling and thread scheduling, SMP process scheduling in Linux, multiprocessor thread scheduling, load sharing, gang scheduling, dedicated processor assignment, dynamic scheduling. Real-time scheduling: structure of real-time system, various terms in real-time scheduling, characteristics of real-time os, use of existing schedulers for real-time scheduling, static table driven scheduling, static-priority driven scheduling, dynamic-planning based scheduling, dynamic best effort scheduling, deadline scheduling, rate-monotonic scheduling, earliest deadline first scheduling, least laxity first scheduling, performance measures, process scheduling in Linux 2.6

## [5] DISTRIBUTED OPERATING SYSTEMS

Issues in Distributed Operating Systems. Inherent Limitations of a Distributed System, Lamport's Logical Clocks, Vector Clocks, Causal Ordering of Messages, Termination Detection. The Classification of Mutual Exclusion Algorithms in distributed systems, Non-token based algorithms -Lamport's Algorithm, The Ricart-Agrawala Algorithm, Maekawa's Algorithm, Token based algorithms - Suzuki-Kasami's Broadcast Algorithm, Singhal's Heuristic Algorithm, Raymond's Heuristic Algorithm. Distributed Deadlock Detection Algorithms

## [6] THREADS

Threads vs process, linux implementation of threads, user-level threads, kernel-level threads, thread programming model: master-slave, peer model, pipeline model. Pthread library.

## [7] MULTI-CORE PROGRAMMING

Evolution of multi-core technology, hyperthreading vs multi-core, design patterns for parallel programming: master-worker, Single-program Multiple-data, Pipelining, Divide and Conquer. OpenMP: MPI vs OpenMP, OpenMP solution stack, OpenMP programming model, OpenMP compiler directives, shared vs private variable in OpenMP, critical section,

## [8] DISTRIBUTED MEMORY MANAGEMENT

Memory management in older systems, Architecture and Motivation, Algorithms for Implementing DSM, Memory Coherence, Coherence Protocols, Design Issues

## [9] FILE SYSTEM INTERFACE AND IMPLEMENTATION

Unix File system, Linux file system (Ext2), VFS, Network file system, Directory cache, Buffer cache, Proc file system, Device special files, file system mounting.

### [10] I/O SUB-SYSTEMS

Device controller, device driver, interrupt mechanism, polling vs interrupt driven driver, I/O subsystem in Linux. Interrupt driven device driver, bottom-half processing, Linux loadable kernel module (LKM), kernel module, kernel module management commands, automatic loading/unloading of module, use of modprobe, Linux device driver: device special files, register/unregister device, kernel functions, handling interrupts in device driver, proc file system,

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Andrew S. Tanenbaum and Herbert Bos, Modern Operating Systems, Fifth Edition, Pearson, 2022.
2. William Stallings, Operating Systems: Internals and Design Principles, Ninth Edition, Pearson, 2021.
3. Kaiwan N Billimoria, Linux Kernel Programming, Packt Publishing Limited, 2021.
4. Andrew S. Tanenbaum, Maarten Van Steen, Distributed Systems, Second Edition, Pearson Prentice Hall, 2007.
5. Selected Research papers.


## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to remember and use various concepts of traditional systems, multiprocessor systems, and real-time systems. | Remember |
| CO2 | Be able to understand the applicability of various concepts of traditional systems, multiprocessor systems, and real-time systems for given problems or situations. | Understand |
| CO3 | Be able to apply the core and advanced concepts of OS theory and implementation to frame solutions. | Apply |
| CO4 | Be able to analyze the given problem statement or OS component and write/design its solution. | Analyze |
| CO5 | Be able to evaluate one or more options on OS structures, Multiprocessor systems, real-time systems, OS data structures, and OS algorithms. | Evaluate |
| CO6 | Be able to design or create an OS data structure, OS algorithm, OS command, or tool for a given problem statement or application requirements. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 | 1 |
| **CO2** | 2 | 2 | 2 | 1 | 0 | 3 | 0 | 0 | 0 | 1 | 0 | 2 | 2 | 2 | 2 |
| **CO3** | 3 | 2 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 2 | 0 |
| **CO4** | 2 | 2 | 2 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 |
| **CO5** | 2 | 2 | 2 | 2 | 1 | 0 | 1 | 0 | 3 | 2 | 2 | 3 | 2 | 2 | 2 |
| **CO6** | 2 | 2 | 2 | 3 | 3 | 0 | 1 | 0 | 3 | 2 | 2 | 3 | 2 | 2 | 2 |
| **Avg.** | 2.17 | 1.83 | 1.5 | 1.67 | 1.17 | 0.5 | 0.5 | 0 | 1 | 0.83 | 0.67 | 2 | 1 | 3 | 1 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. To study various system calls of Linux/Unix System.
2. To implement an executor that executes another command.
3. To study process tracing using strace and ltrace.
4. To study Multiprocessor Process Scheduling using a simulator.
5. To implement multithreading solution using pthread library.
6. To implement solution using OpenMP.
7. One Project and Presentation
    a. Implementation project in a group of two students:
        i. Process Explorer using proc (Implementation: C language)
        ii. System Explorer using proc (Implementation: C language)
        iii. Implementation of pstree command (Implementation: C language)
        iv. Implementation of a new system call (Implementation: C language)
        v. System call tracer-strace (Implementation: C language)
        vi. Detection of USB devices (Implementation: C language)
        vii. Implementation of Emulated Linux File system on windows (C or Java)
        viii. Process Explorer using proc (Implementation: Java language)
        ix. System Explorer using proc (Implementation: Java language)
        x. Implementation of pstree command (Implementation: Java language)
        xi. Modifying scheduling code of kernel
        xii. Implementation of Device Driver
        xiii. Implementation of Command Shell
    b. Study Projects with some implementation or demonstration
        i. Study of Linux ELF Loader
        ii. Study of integrating Bare Metal Hypervisor, Storage Area Network (SAN), and Network Attached Storage (SAN)
        iii. Study of NixOS: installation and changes
        iv. Packet path tracing in a network.

v. Study of Linux kernel for Android
vi. Study of Network Virtualization by Hypervisor
vii. Study of CPU virtualization by Hypervisor
viii. Study of Storage Virtualization by Hypervisor
ix. Study of Docker image format

## B. TECH. SEMESTER – VII (IT)

## SUBJECT: WEB TECHNOLOGY (ELECTIVE-6)

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code: PEC

## A. COURSE OBJECTIVES

- To teach how to use web controls to develop the website in asp.net
- To illustrate the development of Web Caching, Web services, and Web user control in asp.net
- To illustrate data binding and ADO.NET framework in asp.net
- To explain Three-tier architecture and MVC architecture with its implementation.
- To explain the website and web service deployment developed in asp.net

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] INTRODUCTION .NET PLATFORM

Featuring of .Net Platform Components of the .Net Platform

### [2] INTRODUCTION C#.NET

Developing applications using C#.NET, Partial class, Collection, Lists, Arrays, Strings, Events, Delegates, Threading, Exception handling, Networking, File I/O, Generic

### [3] ASP.NET SERVER CONTROLS

Server-side processing in ASP.NET Using HTML controls, Using ASP.NET Standard controls (BulletedList, Multiview and View, ImageMap, Wizard, Substitution, HiddenField, FileUpload), Using ASP.NET and Login Controls Using ASP.NET Validation controls Using ASP.NET Navigation controls

### [4] DATA MANAGEMENT WITH ADO.NET

Basic ADO.NET features, Using Connection, Command, DataReader, DataAdapter object Using Parameters, DataSet and DataTable, Display data from Database, Usage of Web control to access database Customize data bound result using style sections Modify data with SQL statements, Manipulating data within ADO.NET

### [5] DATA BINDING IN ASP.NET USING DATA SOURCE CONTROLS

Using bound list controls with data source controls, GridView, DetailView, FormView, DataList, DropDownList, TreeView, Menu, Adrotator

### [6] MVC ARCHITECTURE PATTERN

What is MVC architecture? The Concept of MVC Architecture – The Three Levels of Architecture, Benefits of Using MVC Framework, Discussion about Popular MVC Frameworks

### [7] ASP.NET MVC FUNDAMENTALS

Understanding of Controllers in ASP.NET MVC with ActionResults, Action Result types and parameters, Routing Class File and its parameter understanding, Passing Data to Views from controller using ViewBag and ViewData its advantages and Disadvantages, Understanding of Models and Binding of Model with View and Controller.

### [8] ENTITY FRAMEWORK WITH LINQ

Database First Model Vs Code First Model, Entity-Table Mapping, Context & Entity Classes, DbContext Class in Entity Framework, DbSet Entities, CRUD operation with Entity framework, Querying with LINQ -to-Entities query with Lazy, eager, and Explicit loading.

### [9] ASP.NET RAZOR VIEW AND HTML HELPER CLASS

Intro to Razor View Engines, Variables Razor Expressions Razor, Code Blocks, Razor @if, else if, else, and @switch, Razor For, Razor Foreach, Compounding of a statement by @using and understanding of HTML helpers like LabelFor, TextBoxFor, DropDownListFor, CheckBoxFor, and so on with display annotation

### [10] PARTIAL VIEW LAYOUT VIEW AND VALIDATION

Partial View and layout view Working with Shared Views Passing Data Dynamically to Shared View and validation controls, Web service using asp.net, User control using Asp.net

### [11] ADVANCED TOPICS IN WEB DEVELOPMENT IN ASP.NET

Web service using asp.net, User control using Asp.net, Caching in asp.net and MVC, Introduction to Dynamics 365, Steps of deployment of MVC application.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Adam Freeman, Pro. ASP.NET Core MVC Ninth Edition, Apress, 2022
2. Christian Nagel, Professional C# 7.0 and .NET Core 2.0,7th Edition, Wrox publication, 2018.
3. Bill evjen, Scott hanselman, Farhan muhammad, Srinivasa sivakumar, Devin rader, Professional ASP.NET 2.0, Special Edition,Wiley India Pvt Ltd, 2006

4. Matthew MacDonald, Beginning ASP.NET 2.0 in C# 2005: From Novice to Professional,Apresss, 2006

## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to understand and use web programming language and MVC programming concepts and namespaces | Remember |
| CO2 | Be able to Understand and use web programming language and MVC concepts and namespaces of asp.net | Understand |
| CO3 | Be able to apply the concepts of ASP .Net technology to solve the given problem on Web-based applications using MVC or Distributed application. | Apply |
| CO4 | Be able to analyze the given problem and choose appropriate concept(s) of ASP .Net such as MVC programming, core .NET, and ADO. Net/Web-service/user control. | Analyse |
| CO5 | Be able to evaluate one or more options to solve the given problem or program requirements and be able to decide a choice under the given context. | Evaluate |
| CO6 | Be able to create a solution for N-Tiered application for a given problem statement and to prepare required design diagrams, MVC architecture, specification, etc. using .Net concepts | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 3 | 3 | 3 |
| **CO2** | 3 | 2 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 3 | 3 | 3 |
| **CO3** | 2 | 3 | 1 | 1 | 1 | 2 | 1 | 0 | 1 | 1 | 0 | 1 | 3 | 2 | 3 |
| **CO4** | 2 | 3 | 3 | 2 | 3 | 2 | 2 | 0 | 1 | 2 | 0 | 1 | 3 | 2 | 3 |
| **CO5** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 1 | 2 | 0 | 1 | 3 | 3 | 3 |
| **CO6** | 0 | 1 | 1 | 1 | 1 | 2 | 1 | 0 | 2 | 1 | 0 | 1 | 3 | 2 | 3 |
| **Avg.** | **1.83** | **2** | **1.33** | **1.77** | **1.17** | **1.33** | **1.17** | **0** | **1.17** | **1.17** | **0** | **1** | **3** | **2.5** | **3** |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. Learn Basic Calculator with asp.net and other control structures with C#.NET

2. Create a User Profile with the following requirements with appropriate validation control
   - Enter First Name and last name
   - Select the country from the drop-down list and
   - Select gender from RadiobuttonList
   - Show an image of the user using Image Control
   - Allow user to select Date of birth from Calendar Control
   - Allow user to select Hobbies using CheckboxList

3. Create an application with GridView and Detail View with Sql Data Source which will satisfy the following requirements
   - Details of all the students should be displayed in Gridview from student relation
   - Users can Select, Edit, and Delete individual students.
   - click Select Button the student record will be displayed in the details view
   - In detail, the view User can Insert, Update, and Delete the record.

4. Create an application using SqlClient which will upload all data of userprofile to the database and also implement different state management techniques.

5. Create an MVC application to implement controllers that pass the data between View and Controller using Viewdata and Viewbag.

6. Create an application that retrieves data from a database using Entity Framework, DbContext Class, and DbSet Entities in Entity Framework.

7. Implement the CRUD operations using Linq in MVC.

8. Create cache applications in which there should be an implementation of File-based dependency, key-based dependency, and Time-based dependency.

9. Implement the Calculator application using the MVC framework and learn how to use HTML helpers LabelFor, TextBoxFor, DropDownListFor, and CheckBoxFor.

10. Implement Web service by using asp.net. Developers want to create a service that returns student details from student relations whose student ID is given as input. Also, implement a code to call this service from the .aspx page and display data in grid view. Structure of student relation is : (student_id, student_password, student_Name, student_marks)

## B. TECH. SEMESTER – VII (IT)

## SUBJECT: SPEECH & NATURAL LANGUAGE PROCESSING (ELECTIVE-6)

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 4 | 0 | 2 | 6 | 5 | 60 | 40 | 25 | 25 | 150 |

Reference Code PEC

## A. COURSE OBJECTIVES

- To make students aware of the concepts and terminologies used in language and speech processing tasks.
- To make students understand the fundamental concepts of natural language processing and automatic speech recognition
- To make students capable of applying data processing techniques in speech and natural language form.
- To make students capable of evaluating the suitability of technique for speech and natural language processing-related applications.
- To make students capable of analyzing and comparing various methodologies options available for processing data in natural language
- To make students capable of creating applications of natural language and speech processing systems using various techniques.

## B. DETAILED SYLLABUS

**Unit** **Topic(s)** **Hrs.**

**[1] INTRODUCTION TO NLP AND SPEECH PROCESSING [01]**

Definition, Applications of Speech and Natural Language Processing in Artificial Intelligent Systems.

**[2] PREPROCESSING [03]**

Basic units- Words, Corpora, utterance, tokens, lemmas, code-switching, tokenization - Byte pair encoding algorithm, stemming-porter stemmer algorithm, Word Normalization, Case folding, Lemmatization, and Stemming.

**[3] N-GRAM LANGUAGE MODELS [03]**

bigram, maximum likelihood estimation (MLE), higher order n-grams, Evaluating Language Models- perplexity, unknown words, Smoothing- laplace smoothing.

**[4] NAIVE BAYES AND CLASSIFICATION [05]**

Naive Bayes Classifiers, Training the Naive Bayes Classifier, Worked example, Naive Bayes as a Language Model, Evaluation: Precision, Recall, F-measure, Test sets and Cross-validation, Statistical Significance Testing

**[5] VECTOR SEMANTICS AND EMBEDDINGS [05]**

Lexical Semantics -word similarity and relatedness, semantic frames, connotations, Vector Semantics- vector embeddings, vector space model, Cosine for measuring similarity, TF-IDF: Weighing terms in the vector, Applications of the tf-idf vector models, other embeddings eg. Word2vec.

**[6] SEQUENCE LABELING FOR PARTS OF SPEECH AND NAMED ENTITIES [07]**

Word Classes, Part-of-Speech Tagging, Named Entities and Named Entity Tagging, HMM Part-of-Speech Tagging, Conditional Random Fields (CRFs), Evaluation of Named Entity Recognition

**[7] SPEECH RECOGNITION AND PROCESSING [06]**

The Automatic Speech Recognition (ASR) Task, Feature Extraction for ASR: Log Mel Spectrum, ASR Evaluation: Word Error Rate, Text to speech tasks -preprocessing, normalization, spectrom prediction; TTS evaluation

**[8] ANNOTATING LINGUISTIC STRUCTURES [06]**

Syntactic structure: Context-Free Grammars, Constituency Parsing, Treebanks, Dependency Parsing, Semantic structure : Relation Extraction, WordNet, Lexicons for Sentiment, Discourse structure: Coreference Resolution-Coreference Tasks and Datasets, Phonetics- Phonetic Resources.

**[9] RECENT ADVANCEMENTS IN SNLP DOMAIN [04]**

Large language models, RNN, transformers, and other models.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Dan Jurafsky and James Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition,Second Edition, Prentice Hall, 2009.
2. Manning, C. D. and H. Schütze: Foundations of Statistical Natural Language Processing. The MIT Press. 1999.
3. Barton, E., Berwick, R., and Ristad, E. Computational Complexity and Natural Language: The MIT Press. 1987.
4. Allen, J. Natural Language Understanding. The Benajmins/Cummings Publishing Company Inc. 1994.
5. Brady, J., and Berwick, R. Computational Models of Discourse. The MIT Press, 1983.
6. Steven Bird, Ewan Klein, and Edward Loper, Natural Language Processing with Python, O'Reilly, First Edition, 2009.

## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | To remember the concepts and terminologies used in language and speech processing tasks. | Remember |
| CO2 | To understand the fundamental concepts for natural language processing and automatic speech recognition | Understand |
| CO3 | To apply techniques for processing data in speech and natural language form. | Apply |
| CO4 | To analyze and compare various methodologies options available for processing data in natural language | Analyze |
| CO5 | To evaluate the suitability of technique for speech and natural language processing related applications. | Evaluate |
| CO6 | To create applications of natural language and speech processing systems using various techniques. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 3 | 3 | 2 | 3 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 3 | 1 | 2 | 2 |
| CO2 | 3 | 3 | 2 | 3 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 3 | 1 | 2 | 2 |
| CO3 | 3 | 3 | 2 | 3 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 3 | 1 | 2 | 2 |
| CO4 | 3 | 3 | 2 | 3 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 3 | 1 | 2 | 2 |
| CO5 | 2 | 2 | 2 | 3 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 3 | 1 | 2 | 2 |
| CO6 | 1 | 2 | 2 | 3 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 3 | 1 | 2 | 2 |
| Avg. | 2.5 | 2.67 | 2 | 3 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 3 | 1 | 2 | 2 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## F. LIST OF EXPERIMENTS

1. To access and study various text Corpora and Lexical Resources - wordnet, Brown, etc.
2. (i) To convert speech to text and text to speech, (ii) To represent text.
3. (i) To Compute statistics about the text - frequency, average word length, total words in a text, etc. (ii) To do text tokenization using- nltk, regex, spacy etc.

4. To do text processing- stop word removal using various stopword lists in nltk, gensim, spacy, etc.
5. To do text stemming using porter stemmer.
6. To do word tagging- using various taggers like Regular expression tagger, UnigramTagger etc.
7. To extract information from text- NER, Chunking
8. To analyze sentence structure- constituency parsing and dependency parsing
9. To analyze meaning of sentence- discourse semantics
10. To learn text classification- naive Bayes classification.

## B. TECH. SEMESTER – VII (IT)

## SUBJECT: PROJECT-II

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 0 | 0 | 2 | 2 | 1 | 0 | 0 | 25 | 25 | 50 |

Reference Code PROJ-CS701

## A. COURSE OBJECTIVES

- To allocate a project having a research component in a group of a minimum of two students (the allocated project is desirable to be unique among all the other projects of the peer students of the BTech IT programme)
- To guide students in conducting thorough research within the project domain, assisting them in identifying relevant literature and research gaps.
- To facilitate discussions and brainstorming sessions to refine project requirements based on research insights, ensuring alignment with industry standards and emerging trends.
- To provide guidance on integrating innovative design concepts derived from research findings into the system architecture and user interface design.
- To carry out regular reporting and monitor the progress of the project with an emphasis on supervising students in the implementation process, encouraging them to apply research-driven requirements and design decisions effectively.
- To assist in the rigorous validation and evaluation of the developed system, helping students assess its effectiveness and contribution to addressing research objectives.
- To nurture effective communication and reporting skills among students, providing feedback on project documentation and presentations to ensure clarity and coherence in conveying research methodologies, findings, and outcomes.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] RESEARCH AND REQUIREMENTS IDENTIFICATION

Define the scope of the project and potential research areas, Conduct a literature review, Formulate research questions, Explore methodologies, and Develop a plan for data collection, experiments, and data analysis.

### [2] RESEARCH ANALYSIS AND CONCEPTUALIZATION

Analyze the gathered research data to extract insights and trends relevant to the project. Refine and expand functional and non-functional requirements based on research findings, prepare use case diagrams, prepare SRS document integrating research insights, and Prepare wireframes/mockup screens with system flow.

### [3] RESEARCH DESIGN AND PROTOTYPE DEVELOPMENT

Design system architecture considering the research-driven requirements and scalability. Develop class diagrams, sequence diagrams, and necessary design artifacts. Design the database schema for prototype development and prepare or collect the required data for the research component. Create a component and deployment diagram. Also prepare statechart and activity diagrams, if required. Prototype the system incorporating the designed architecture and interfaces for validation and feedback.

### [4] IMPLEMENTATION AND VALIDATION

Implement the prototype using selected development platforms, tools, and programming languages. Conduct unit testing to ensure the correctness of individual components. Perform integration testing to verify the interactions between system modules. Validate the system against the initial research objectives and requirements. Iterate on the implementation based on feedback and validation results.

### [5] EVALUATION AND REPORTING

Evaluate the developed system's performance, usability, and effectiveness based on research-driven criteria. Write a comprehensive project report detailing the research methodology, findings, system design, implementation process, and evaluation results. Present the project findings and outcomes to stakeholders, including peers and faculty members. Reflect on the challenges faced and lessons learned during the project lifecycle. Receive feedback from stakeholders and incorporate suggestions for future improvements.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Casey Rosenthal, Lorin Hochstein, Aaron Blohowiak, Nora Jones, Ali Basiri, Chaos Engineering, O'Reilly, 2017.
2. Kirsty Williamson and Graeme Johanson (Editors), Research Methods: Information, Systems, and Contexts, Elsevier, 2018.
3. Niall Richard Murphy, Betsy Beyer, Chris Jones, and Jennifer Petoff, Site Reliability Engineering: How Google Runs Production Systems, O’ Reilly, 2016
4. C. R. Kothari and Gaurang Garg, Research Methodology: Methods and Techniques, Second Edition, New Age International (P) Limited, Fifth Edition, 2023.

## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to conduct comprehensive research within the project domain, encompassing literature review, data collection, and analysis, to identify pertinent research gaps and opportunities for innovation. | Evaluate |
| CO2 | Be able to refine project requirements through critical analysis of research findings, translating them into clear and concise system specifications that encompass both functional and non-functional aspects. | Analyze |
| CO3 | Be able to conceptualize and implement innovative design solutions informed by research insights, effectively integrating them into the system architecture and user interface design to address identified research challenges. | Create |
| CO4 | Be able to execute the project implementation phase proficiently, utilizing selected development platforms, languages, and tools, while ensuring alignment with research-driven requirements and design decisions. | Create |
| CO5 | Be able to conduct rigorous validation and evaluation of the developed system against research objectives, demonstrating the capacity to assess its efficacy, reliability, and contribution to knowledge advancement. | Evaluate |
| CO6 | Be able to communicate research methodologies, findings, and project outcomes effectively through written reports and oral presentations, catering to diverse audiences and stakeholders. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 0 | 1 | 3 | 3 | 3 |
| **CO2** | 3 | 2 | 3 | 1 | 0 | 2 | 1 | 3 | 3 | 1 | 3 | 1 | 3 | 3 | 3 |
| **CO3** | 2 | 3 | 2 | 1 | 1 | 2 | 1 | 3 | 3 | 1 | 0 | 1 | 3 | 3 | 3 |
| **CO4** | 2 | 3 | 3 | 2 | 3 | 2 | 2 | 3 | 3 | 2 | 0 | 1 | 3 | 3 | 3 |
| **CO5** | 2 | 2 | 1 | 3 | 2 | 3 | 2 | 3 | 3 | 2 | 3 | 1 | 3 | 3 | 3 |
| **CO6** | 1 | 2 | 3 | 3 | 3 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| **Avg.** | 2 | 2.17 | 2 | 1.67 | 1.5 | 1.83 | 1.33 | 3 | 3 | 1.5 | 1.5 | 1.33 | 3 | 3 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)

## B. TECH. SEMESTER – VIII (IT)

## SUBJECT: INDUSTRIAL INTERNSHIP

| Teaching Scheme (Hours/Week) | | | | Credits | Examination Scheme | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Lect | Tut | Prac | Total | | Ext | Sess. | TW | Prac | Total |
| 0 | 6 | 24 | 30 | 18 | 0 | 0 | 150 | 350 | 500 |

Reference Code PROJ-CS801

During this semester-long industrial training program, undergraduate students will engage in real-world projects within the IT industry. The primary focus will be on the practical application of software development methodologies to solve industry-relevant problems. Students will follow a structured software development lifecycle, encompassing various phases from requirements gathering to project delivery.

## A. COURSE OBJECTIVES

- To provide students with hands-on experience in developing software solutions for real-world problems.
- To familiarize students with industry-standard software development processes and methodologies.
- To cultivate skills in project management, software analysis, design, implementation, testing, and documentation.
- To encourage collaboration and communication within a professional IT environment.

## B. DETAILED SYLLABUS

**Unit Topic(s)**

### [1] PROJECT WORK IN INDUSTRIAL TRAINING

Students will undertake a semester-long project involving the design and development of a software system.

Projects may also involve the analysis and improvement of existing industry systems.

Emphasis will be placed on adhering to the standard software development process, including requirements gathering, system analysis, design, implementation, testing, documentation, and user manual preparation.

Project guides may visit the project sites.

Assessment will be based on project reports, seminars, demonstrations, and viva examinations.

### [2] REQUIREMENTS GATHERING AND PROJECT MANAGEMENT

Students will explore and evaluate project requirements, conduct market surveys, and assess technical feasibility through practical tasks.

### [3] ANALYSIS

Students will analyze functional and non-functional requirements, create use case diagrams, prepare software requirement specifications (SRS), and develop wireframes/mockup screens.

### [4] DESIGN

Students will create architecture diagrams, class diagrams, sequence diagrams, database designs, and deployment diagrams.

Additional diagrams such as state charts and activity diagrams may be required based on project needs.

### [5] IMPLEMENTATION

Students will implement the designed solution using appropriate development platforms, integrated development environments (IDEs), and programming languages.

### [6] TESTING

Students will conduct UI testing and integration testing to ensure the functionality and robustness of the developed software.

### [7] REPORT WRITING

Students will compile project reports following a standard format, documenting their project journey, methodologies, findings, and outcomes.

## C. RECOMMENDED TEXT / REFERENCE BOOKS


1. Roger S. Pressman, Software Engineering-A Practitioner's Approach, McGrawHill, 2019.
2. Robert C. Martin, Clean Code-A Handbook of Agile Software Craftsmanship, Prentice Hall, 2009.
3. Martin Fowler, UML Distilled: A Brief Guide to the Standard Object Modeling Language, Addison-Wesley, 2023.

## D. COURSE OUTCOMES

| CO# | CO Statement | Skill |
|---|---|---|
| CO1 | Be able to explore and evaluate requirements of the project/system to be built in addition to feasibility study. | Evaluate |
| CO2 | Be able to analyse the functional and non-functional requirements of the project/system and be able to specify them in form of Use case diagram, System Requirements Specification, Hardware/Software Requirements, and User Interface diagram using appropriate UML tools. | Analyze |
| CO3 | Be able to prepare design and architecture of the system using sequence diagram, class diagram, component diagram, and deployment diagram using appropriate UML tools. | Create |
| CO4 | Be able to implement the solution using chosen development platform, language, and IDEs. | Create |
| CO5 | Be able to test, troubleshoot, and record test cases for implemented functionality, as a standalone unit and in integration with other functionalities. | Analyze |
| CO6 | Be able to work with an approach of life-long learning for various phases of software development lifecycle such as feasibility study, project planning, requirements analysis and specification, system design, implementation, unit testing, integration testing, and report writing. | Create |

## E. COURSE MATRIX

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 0 | 1 | 3 | 3 | 3 |
| **CO2** | 3 | 2 | 3 | 1 | 0 | 2 | 1 | 3 | 3 | 1 | 3 | 1 | 3 | 3 | 3 |
| **CO3** | 2 | 3 | 2 | 1 | 1 | 2 | 1 | 3 | 3 | 1 | 0 | 1 | 3 | 3 | 3 |
| **CO4** | 2 | 3 | 3 | 2 | 3 | 2 | 2 | 3 | 3 | 2 | 0 | 1 | 3 | 3 | 3 |
| **CO5** | 2 | 2 | 1 | 3 | 2 | 3 | 2 | 3 | 3 | 2 | 3 | 1 | 3 | 3 | 3 |
| **CO6** | 1 | 2 | 3 | 3 | 3 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| **Avg.** | 2 | 2.17 | 2 | 1.67 | 1.5 | 1.83 | 1.33 | 3 | 3 | 1.5 | 1.5 | 1.33 | 3 | 3 | 3 |

1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High)