**SYLLABUS**

**FOR**

**THREE-YEAR MASTER PROGRAMME**

**IN**

**MASTER IN COMPUTER APPLICATION**



|  |
| --- |
| **NAAC – A Grade** |

**DEPARTMENT OF COMPUTER SCIENCE AND APPLICATION**

**COLLEGE OF ENGINEERING & TECHNOLOGY**

**(An Autonomous and Constituent College of BPUT, Odisha)**

**Techno Campus, Mahalaxmi Vihar, Ghatikia,**

**Bhubaneswar-751029, Odisha, INDIA**

[**www.cet.edu.in**](http://www.cet.edu.in)

**Ph. No.: 0674-2386075 (Off.), Fax: 0674-2386182**

**COURSE: MCA (CSA - Master in Computer Application)**

**Duration: 3 years (Six Semesters)**

**Abbreviations Used: U= UG, I= Integrated, P= PG**

**PC= Professional Core PE= Professional Elective OE= Open Elective**

**LC= Lab Course MC= Mandatory Course AC= Audit Course**

**L= Lectures P= Practical/Laboratory IA\*= Internal Assessment**

**T= Tutorial PA= Practical Assessment EA=End-Semester Assessment**

**\*Internal Assessment Max. Mark (30 marks) consists of Mid Semester (20 marks) and Quiz+Assignment (10 marks)**

**Subject Code Format:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **Prog (U/I/P)** | **Type (PC/PE/OE/LC/MC/AC)** | | **Department (CE/EE/IE/ME/…)** | | **Semester (1/2/…/0)** | **Serial No. (1/2/3/…/99)** | |

**1st SEMESTER**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl. No.** | **Subject**  **Type** | **Subject Code** | **Subject**  **Name** | **Teaching Hours** | | | **Credit** | **Maximum Marks** | | | |
| **L** | **T** | **P** | **IA** | **EA** | **PA** | **Total** |
| 1 | Core 1 | PPCCA101 | Problem Solving and Program Design | 4 | 0 | 0 | 4 | 30 | 70 | - | 100 |
| 2 | Core 2 | PPCCA102 | Computer Organization and Architecture | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 3 | Core 3 | PPCCA103 | Web Design and Development | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 4 | Core 4 | PPCCA104 | Computational Mathematics | 4 | 0 | 0 | 4 | 30 | 70 | - | 100 |
| 5 | Mandatory 1 | PMCMH102 | Engineering Economics | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 6 | Mandatory 2 | PMCMH103 | Business Communication | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 7 | Lab 1 | PLCCA101 | Problem Solving and Program Design Lab | 0 | 0 | 4 | 2 | - | - | 100 | 100 |
| 8 | Lab 2 | PLCCA102 | Computer Organization and Architecture Lab | 0 | 0 | 2 | 1 | - | - | 100 | 100 |
| 9 | Lab 3 | PLCCA103 | Web Design and Development Lab | 0 | 0 | 2 | 1 | - | - | 100 | 100 |
| 10 | Lab 4 | PLCMH104 | Business Communication Lab | 0 | 0 | 2 | 1 | - | - | 100 | 100 |
| **Total** | | | | **20** | **0** | **10** | **25** | **180** | **420** | **400** | **1000** |

**2nd SEMESTER**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl. No.** | **Subject**  **Type** | **Subject Code** | **Subject**  **Name** | **Teaching Hours** | | | **Credit** | **Maximum Marks** | | | |
| **L** | **T** | **P** | **IA** | **EA** | **PA** | **Total** |
| 1 | Core 5 | PPCCA201 | Data Structures | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 2 | Core 6 | PPCCA202 | Programming with Java | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 3 | Core 7 | PPCCA203 | Operating Systems | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 4 | Core 8 | PPCCA204 | Database Management Systems | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 5 | Core 9 | PPCCA205 | Software Engineering | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 6 | Mandatory 3 | PMCMH201 | Organizational Behaviour | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 7 | Lab 5 | PLCCA201 | Data Structures Lab | 0 | 0 | 2 | 1 | - | - | 100 | 100 |
| 8 | Lab 6 | PLCCA202 | Programming with Java Lab | 0 | 0 | 4 | 2 | - | - | 100 | 100 |
| 9 | Lab 7 | PLCCA203 | Operating Systems Lab | 0 | 0 | 2 | 1 | - | - | 100 | 100 |
| 10 | Lab 8 | PLCCA204 | Database Management Systems Lab | 0 | 0 | 2 | 1 | - | - | 100 | 100 |
| 11 | Lab 9 | PLCCA205 | Software Engineering Lab | 0 | 0 | 2 | 1 | - | - | 100 | 100 |
| **Total** | | | | **18** | **0** | **12** | **24** | **180** | **420** | **500** | **1100** |

**3rd SEMESTER**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl. No.** | **Subject Type** | **Subject Code** | **Subject**  **Name** | **Teaching Hours** | | | **Credit** | **Maximum Marks** | | | |
| **L** | **T** | **P** | **IA** | **EA** | **PA** | **Total** |
| 1 | Core 10 | PPCCA301 | Design and Analysis of Algorithms | 4 | 0 | 0 | 4 | 30 | 70 | - | 100 |
| 2 | Core 11 | PPCCA302 | Theory of Computation | 4 | 0 | 0 | 4 | 30 | 70 | - | 100 |
| 3 | Core 12 | PPCCA303 | Computer Networks | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 4 | Core 13 | PPCCA304 | Cryptography and Network Security | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 5 | Core 14 | PPCCA305 | Data Science with Python | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 6 | Project 1 | PPRCA306 | Minor Project-I | 0 | 0 | 4 | 2 | - | - | 100 | 100 |
| 7 | Lab 10 | PLCCA301 | Computer Networks Lab | 0 | 0 | 4 | 2 | - | - | 100 | 100 |
| 8 | Lab 11 | PLCCA302 | Cryptography and Network Security Lab | 0 | 0 | 2 | 1 | - | - | 100 | 100 |
| 9 | Lab 12 | PLCCA303 | Data Science with Python Lab | 0 | 0 | 4 | 2 | - | - | 100 | 100 |
| **Total** | | | | **17** | **0** | **14** | **24** | **150** | **350** | **400** | **900** |

**4th SEMESTER**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl. No.** | **Subject**  **Type** | **Subject Code** | **Subject**  **Name** | **Teaching Hours** | | | **Credit** | **Maximum Marks** | | | |
| **L** | **T** | **P** | **IA** | **EA** | **PA** | **Total** |
| 1 | Core 15 | PPCCA401 | Artificial Intelligence | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 2 | Core 16 | PPCCA402 | Enterprise Java Technologies | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 3 | Core 17 | PPCCA403 | Compiler Design | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 4 | Core 18 | PPCCA404 | Optimisation Techniques | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 5 | Professional Elective 1  (Any One) | PPECA401 | Computer Graphics | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| PPECA402 | Data Warehousing and Mining |
| PPECA403 | Internet of Things |
| 6 | Mandatory 4 | PMCMH401 | Personality  and Soft Skill Development | 0 | 0 | 4 | 2 | - | - | 100 | 100 |
| 7 | Mandatory 5 | PMCMH402 | Group Discussion and Seminar | 0 | 0 | 4 | 2 | - | - | 100 | 100 |
| 8 | Lab 13 | PLCCA401 | Artificial Intelligence Lab | 0 | 0 | 2 | 1 | - | - | 100 | 100 |
| 9 | Lab 14 | PLCCA402 | Enterprise Java Technologies Lab | 0 | 0 | 4 | 2 | - | - | 100 | 100 |
| 10 | Lab 15 | PLCCA403 | Compiler Design Lab | 0 | 0 | 2 | 1 | - | - | 100 | 100 |
| 11 | Lab 16 | PLCCA404 | Elective-I Lab | 0 | 0 | 2 | 1 | - | - | 100 | 100 |
| **Total** | | | | **15** | **0** | **18** | **24** | **150** | **350** | **600** | **1100** |

Students must undertake a summer internship programme at the end of the 4th semester which will be evaluated in 5th semester.

**5th SEMESTER**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl. No.** | **Subject Type** | **Subject Code** | **Subject**  **Name** | **Teaching Hours** | | | **Credit** | **Maximum Marks** | | | |
| **L** | **T** | **P** | **IA** | **EA** | **PA** | **Total** |
| 1 | Core 19 | PPCCA501 | Models and Patterns for Web Development | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| 2 | Professional Elective 2  (Any One) | PPECA501 | Multimedia and Computer Animation | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| PPECA502 | Machine Learning |
| PPECA503 | Mobile Computing |
| PPECA504 | Software Project Management |
| 3 | Professional Elective 3  (Any One) | PPECA505 | Virtual Reality | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| PPECA506 | Statistical Computing |
| PPECA507 | Cloud Computing |
| PPECA508 | Software Testing |
| 4 | Open Elective 4  (Any One) | POECA501 | Digital Image Processing | 3 | 0 | 0 | 3 | 30 | 70 | - | 100 |
| POECA502 | Big Data Analytics |
| POECA503 | Natural Language Processing |
| POECA504 | Wireless Sensor Networks |
| POECA505 | Parallel and Distributed Systems |
| POECA506 | Soft Computing |
| POEMH507 | Entrepreneurship Development |
| POEMH508 | Marketing Management |
| 5 | Mandatory 6 | PMCMH501 | Professional Ethics | 2 | 0 | 0 | 2 | 30 | 70 | - | 100 |
| 6 | Project 2 | PPRCA506 | Minor Project-II | 0 | 0 | 4 | 2 | - | - | 100 | 100 |
| 7 | Project 3 | PPRCA507 | Summer Internship Evaluation | 0 | 0 | 4 | 2 | - | - | 100 | 100 |
| 8 | Lab 17 | PLCCA501 | Models and Patterns for Web Development Lab | 0 | 0 | 2 | 1 | - | - | 100 | 100 |
| 9 | Lab 18 | PLCCA502 | Elective-II Lab | 0 | 0 | 2 | 1 | - | - | 100 | 100 |
| 10 | Lab 19 | PLCCA503 | Elective-III Lab | 0 | 0 | 2 | 1 | - | - | 100 | 100 |
| **Total** | | | | **14** | **0** | **14** | **21** | **150** | **350** | **500** | **1000** |

**6th SEMESTER**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sl. No.** | **Subject Type** | **Subject Code** | **Subject**  **Name** | **Teaching Hours** | | | **Credit** | **Maximum Marks** | | | |
| **L** | **T** | **P** | **IA** | **EA** | **PA** | **Total** |
| 1 | Project 4 | PPRCA601 | Industrial Training cum Project | 0 | 0 | 30 | 15 | - | - | 100 | 100 |
| **Total** | | | | **0** | **0** | **30** | **15** | **-** | **-** | **100** | **100** |

**Abstract of Credit and Marks Distribution**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Semester** | **Maximum Credits** | **Maximum Marks** |
| 1 | 1st Semester | 25 | 1000 |
| 2 | 2nd Semester | 24 | 1100 |
| 3 | 3rd Semester | 24 | 900 |
| 4 | 4th Semester | 24 | 1100 |
| 5 | 5th Semester | 21 | 1000 |
| 6 | 6th Semester | 15 | 100 |
| **Total** | | **133** | **5200** |

**Semester-1**

**Core 1: Problem Solving and Program Design (PPCCA101)**

**Course Objectives:**

1. To provide understanding of algorithmic approach to problem solving.
2. To provide knowledge on Procedural as well as Object Oriented Approaches to program design.
3. To provide elaborate knowledge on C language to write procedural programs.
4. To introduce relevant features of C++ language to write object oriented programs.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Develop skills to write computer programs to solve a variety of real-world problems.
2. Write programs using both procedural and object oriented approaches
3. Design programs using readable, reusable and cohesive modules.
4. Develop skills to use pointers and data files in programs.

**Course Prerequisites:**

1. This course does not require any prerequisite as such.

**Detailed Syllabus:**

**Module I**

***Introduction to Computer***: Basic Organization of a Computer, Hardware and Software, Programming Languages, Number System, Conversion.

***Program Development****:* Programming as Problem-Solving, Steps in Program Development, Algorithm, Flowchart, Pseudo code, Top-down and Bottom-up approaches, Characteristics of a good program, Structure of a C Program, Compiling, Linking and Executing Programs.

***C Language Fundamentals****:* Language Elements, Data Types, Variables and Constants, Operators, Expressions, Type Conversions, Statements, Managing Console Input and Output Operations, Function.

***Control Structures****:* Decision Making and Branching - If and Switch, Loop Structures - While, Do While and For, Unconditional Jumps - Continue, Break and Go To.

**Module II**

***Arrays and Strings****:* Concept, Declaration and Manipulation of Arrays, One Dimensional and Multidimensional Arrays, Sorting and Searching an Array, Concept of Strings, String Handling Functions, Array of Strings.

***Pointers****:* Pointer Variable and its Importance, Dereferencing, Pointer Arithmetic and Scale Factor, Pointers and Arrays, Pointer and Strings, Array of Pointers, Pointers to Pointers.

***Functions****:* Designing Structured Programs, User Defined and Standard Functions, Formal and Actual Arguments, Function Prototype, Parameter Passing, Functions Returning Multiple Values, Functions Returning Pointers, Pointers to Functions, Nesting of Functions, Recursion, Passing Arrays to Functions.

***Scope and Extent****:* Scope Rules, Storage Classes - Auto, Extern, Register and Static.

**Module III**

***Structures, Unions and Enumerations****:* Declaration and Initialization of Structures, Structure as Function Parameters, Structure Pointers, Unions, Enumerations.

***File I/O****:* Defining, Opening a File and Closing a File, Input/output Operations in Files, Random Access to Files, Error Handling. Command Line Arguments, Dynamic Memory Management, Pre-Processor Directives.

**Module IV**

***Introduction:*** Need of Object Orientation, Basic Concepts of Object Oriented Approach, Basic Program Construction in C++, Namespace, Data Types, Input and Output, Handling Exceptions.

***Objects and Classes****:* Defining and Using Classes, Constructors and Destructors, Controlling Accessibility, Public and Private Class Members, Member Functions, *this* pointer, *static* class data and *const* Member Functions, Constructor and Function Overloading.

***Inheritance:*** Base and Derived classes, Access Control Mechanisms, Types of Inheritance, Virtual Functions, Abstract Class and Pure Virtual Function, Virtual Base Class.

**Text Book:**

1. Jeri R. Hanly and Elliot B. Koffman, Problem Solving and Program Design in C, 8th Edition, Pearson Education, 2016. (Module-I, II, III)
2. The C++ Programming Language, Bjarne Stroustrup, Addition Wesley. (Module-IV)

**Reference Books:**

1. R. G. Dromey, How to Solve it by Computer. Prentice-Hall India EEE Series.
2. E. Balagurusamy, Programming in ANSI C, 4th edition, McGraw-Hill Publication, 2007.
3. Pradip Dey, Manas Ghosh, Programming in C, Second Edition, Oxford University Press, 2011.
4. Brian W. Kernighan, Dennis Ritchie, The C Programming Language, 2nd Edition, Prentice Hall, 1988.
5. Yashavant P. Kanetkar. Let Us C, BPB Publications, 2011.
6. Byron S Gottfried, Programming with C, Schaum’s Outlines, Second Edition, Tata McGraw Hill, 2006.
7. Bruce Eckel, Thinking in C++, Vol. 1: Introduction to Standard C++, 2nd Edition,

**Core 2: Computer Organizations and Architecture (PPCCA102)**

**Course Objectives:**

1. To study the basic organization and architecture of digital computers
2. To study design aspects of different subsystems of a computer system
3. To understand the instructions and instruction execution life cycle
4. To understand various data transfer techniques in digital computer.
5. To understand processor performance improvement using instruction level parallelism
6. To understand microprocessor and assembly language program

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Understand basic structure of a computer.
2. Understand computer instruction and its execution
3. Understand ALU, design basic circuits and perform computer arithmetic operations.
4. Understand different memory and their performance issues
5. Understand cache mapping techniques.
6. Understand I/O organization and data transfer techniques.
7. Understand processor performance and instruction level parallelism.
8. Write basic assembly language programs

**Course Prerequisites:**

This course does not require any prerequisite as such.

**Detailed Syllabus:**

**Module I**

***Introduction:*** Basic architecture of computer, Functional units, Operational concepts, Bus structures, Von Neumann Concept.

***Basic Processing:*** Instruction code, Instruction set, Instruction sequencing, Instruction Cycle & Execution Cycle, Instruction format, Addressing modes, Micro instruction, Data path and control path design, Micro programmed vs. Hardwired controlled unit, RISC vs. CISC.

***Arithmetic:*** Design of ALU, Binary arithmetic, Addition and Subtraction of signed number, Multiplication of Positive number, Signed operand multiplication, Division, Floating point number representation and arithmetic.

***Digital Electronics:*** Boolean algebra, Digital Logic, Truth Tables, K map, Number system, Flip- Flop

**Module II**

***Memory:*** Memory Hierarchy, RAM, ROM, Cache memory organization, Mapping techniques, Virtual memory, Memory Interleaving, Secondary Storage, Flash drives.

**Module III**

***Input/output:*** Accessing I/O devices, I/O mapped I/O, Programmed I/O, Memory Mapped I/O, Interrupt Driven I/O, Standard I/O interfaces, Synchronous and Asynchronous Data transfer, DMA data transfer.

***Introduction to Parallel processing:*** Flynn’s Classification, Pipelining, Super Scalar processors, Array processing, vector processing.

8085 Microprocessor and Assembly Level Programming using 8085 microprocessor

**Text Books:**

1. William Stalling, Computer Organization and Architecture, Pearson Education (Module I, II, II)
2. M. Mano, “Computer System and Architecture”, PHI. (Module IV)

**Reference Books:**

1. J. P. Hayes, “Computer Architecture and Organization", MGH
2. A.S. Tananbaum, “Structured Computer Organization”, Pearson Education
3. Alan Clements, Computer Organization and Architecture, Cengage.
4. C. Hamacher, Z. Vranesic, S. Zaky, “Computer Organization”, McGraw-Hill Education India

**Core 3: Web Design and Development (PPCCA103)**

**Course Objectives:**

1. To provide basic understanding of the Internet and World Wide Web.
2. To provide elaborate knowledge on how to use HTML, CSS and JavaScript to develop webpages.
3. To provide understanding on Front-End Libraries such as jQuery and Bootstrap to develop webpages.
4. To provide knowledge and skills on tools and techniques to develop and implement web projects.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Develop webpages using core front-end coding languages such as HTML, CSS and JavaScript.
2. Create Responsive Websites compatible with different devices and screen sizes.
3. Display ability to develop professional websites using Bootstrap and jQuery.
4. Create and maintain websites on the Internet.

**Course Prerequisites:**

Basic understanding of computer and programming.

**Detailed Syllabus:**

**Module-I**

***Introduction to Internet and World Wide Web****:* Introduction to Internet, client- server model, IP address, protocols, Basic Services, the Internet verses the World Wide Web, Domain Name, URL, Evolution of World Wide Web, Web 2.0.

***Page Structuring using HTML****:* Structure of a webpage, Basic formatting markups, Adding links, images, Table markup, Lists, Forms, Div and Span, Semantic markups in HTML 5.

***Basics of Web Graphics****:* Image formats, Size and Resolution, Transparency, Scalable Vector Graphics, Image Optimization.

***Presentation using CSS****:* Overview of CSS, benefits of CSS, Basic syntax of writing style rules, Selectors, Types of Style sheet, Inheritance and Cascading styles, Text and Font properties, Color and Background properties, Box Model, Page Layout, Floating and Positioning, Styling forms and tables, Basic responsive web design.

**Module-II**

***Page Interaction using JavaScript****:* Introduction to JavaScript, Adding JavaScript to a page, Basics of JavaScript Language, variable, data types, operators, array, control structures, Browser objects, Events, Document Object Model, Accessing page contents using JavaScript, Form validation using JavaScript,

***Introduction to XML and AJAX****:* Basics of XML document, DTD, Schema, XML Http Request object, Sending request and receiving server response using AJAX.

***Introduction to jQuery****:* Basics of jQuery, Selecting elements, Handling events, Applying effects and animations, Manipulating DOM, jQuery and AJAX.

**Module-III**

***Introduction to Bootstrap****:* Overview of Bootstrap, Grid basics, Using Bootstrap Base CSS, Typographic elements, Colors, Images, Buttons, Navs, Navbar, Carousel, Forms.

***Website Design Principles****:* Understanding the web design environment, Designing for multiple screen resolutions, Crafting the look and feel of the site, Creating a unified site design, Designing for the user, Designing for accessibility.

***Site Planning and Publishing****:* Understanding the web site development process, Creating a site specification, Identifying the content goal, Analyzing your audience, Creating conventions for filenames and URLs, Setting a directory structure, Creating a site storyboard, Publishing the web site, Testing the web site, Basic understanding of Search Engine Optimization.

**Main Texts:**

1. The Web Warrior Guide to Web Design Technologies, Don Gosselin, et. al, Cengage Learning

**Recommended Texts:**

1. Learning Web Design, Jennifer N. Robbins, O'Reilly Media
2. HTML & CSS: Design and Build Websites, Jon Duckett, John Wiley
3. jQuery IN ACTION, Bear Bibeault et.al., Dreamtech

**Core 4: Computational Mathematics (PPCCA104)**

**Course Objectives:**

1. To understand various concepts in several areas of discrete mathematics.
2. To develop problem-solving techniques using those mathematical concepts.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Represent various real-world problems using mathematical concepts.
2. Use those mathematical concepts in practical applications.
3. Apply Graph theory in solving computer science problems

**Course Prerequisites:**

This course does not require any prerequisite as such.

**Detailed Syllabus:**

**Module-I**

***Logic, Sets & Counting***

Logic: Propositions and logical operators - Truth table - Propositions generated by a set, Equivalence and implication - Basic laws- Some more connectives - Proofs in Propositional calculus - Predicate calculus.

Sets: Basic Definitions - Venn Diagrams and set operations - Laws of set theory - Principle of inclusion and exclusion – partitions.

Permutation and Combination

Relations: Properties of relations - Matrices of relations - Closure operations on relations - Recurrence relations Functions: injective, subjective and objective functions.

**Module-II**

***Matrices***

Matrices, Rank of Matrix, Solving System of Equations- Gauss Elimination, Eigen Values and Eigen Vectors, Inverse of a Matrix

**Module-III**

***Graphs***

Basic terminology, Eulerian paths and circuits, Hamiltonian paths and circuits, Graph coloring. Trees: definition and properties, tree traversals— preorder, inorder, postorder, binary trees, spanning trees, Graph traversals — BFS and DFS.

**Module-IV**

***Numerical Methods***

Computing Arithmetic, errors, Significant Digits and Numerical Instability, Root finding methods: Bisection, Newton Raphson, Secant and Regula Falsi., Interpolation, Richardson’s extrapolation principle.

Differentiation and Integration: First and second order ordinary differentiation, Trapezoidal and Simpsons method. Solving first order differential equation by Euler’s method.

**TEXT BOOKS:**

1. Kenneth H. Rosen, “Discrete Mathematics and Its Applications”, Tata McGraw Hill, Fourth Edition (Module- I, II & III).
2. S. S. SASTRY, “**INTRODUCTORY METHODS OF NUMERICAL ANALYSIS**”,  [PHI Learning Pvt. Ltd.](https://books.google.co.in/url?id=y88tGx22FzAC&pg=PR2&q=http://www.phindia.com&clientid=ca-print-prentice_hall_india&linkid=1&usg=AFQjCNHMIWVRSLHd03bWD1Ag0bla_5edjA&source=gbs_pub_info_r) Fifth Edition, 2012 ( Module- IV)

**REFERENCE BOOKS:**

1. A. Tamilarasi & A.M. Natarajan, “Discrete Mathematics and its Application”, Khanna Publishers, 2nd Edition 2005.
2. M.K. Venkataraman “Engineering Mathematics”, Volume II, National Publishing Company, 2nd Edition, 1989.
3. Juraj Hromkovic, “Theoretical Computer Science”, Springer Indian Reprint, 2010.
4. David Makinson, “Sets, Logic and Maths for Computing”, Springer Indian Reprint, 2011.
5. K Atkinson. “An Introduction to Numerical Analysis”, 2nd ed., John Wiley, 1989

**MC 1: Engineering Economics (PMCMH102)**

**Course Objectives:**

1. To develop better understanding of the concept of engineering economics including costing and financial accounting.
2. To provide knowledge on various approaches of economics, costing & accounting related to business.
3. To provide elaborate knowledge on the effect of various transactions on business functions.
4. To introduce relevant features of engineering economics & accounting as a language of business.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Develop skills to deal with a variety of real-world business problems.
2. Write accounting concepts as applicable to balance sheet & income statements
3. Know the importance of elements of cost in a business transaction.

**Course Prerequisites:**

This course does not require any prerequisite as such.

**Detailed Syllabus:**

**Module-I**

Engineering Economics- Nature, Scope, Basic problems of an economy, Micro Economics and Macro Economics. Demand- Meaning of demand, Demand function, Law of Demand and its exceptions, Determinants of demand, Elasticity of demand & its measurement (Simple numerical problems to be solved), Supply-Meaning of supply, Law of supply and its exception, Determinants of supply, Elasticity of supply, Determination of market equilibrium (Simple numerical problems to be solved). Production-Production function, Laws of returns: Law of variable proportion, Law of returns to scale

**Module- II**

Cost and revenue concepts, Basic understanding of different market structures, Determination of equilibrium price under perfect competition (Simple numerical problems to be solved), Break Even Analysis-linear approach (Simple numerical problems to be solved). Banking -Commercial bank, Functions of commercial bank, Central bank, Functions of Central Bank. Inflation-Meaning of inflation, types, causes, measures to control inflation. National Income-Definition, Concepts of national income, Method of measuring national income.

**Module-III**

Time value of money- Interest - Simple and compound, nominal and effective rate of interest, Cash flow diagrams, Principles of economic equivalence. Evaluation of engineering projects-Present worth method, Future worth method, Annual worth method, Internal rate of return method, Cost benefit analysis for public projects. Depreciation- Depreciation of capital assert, Causes of depreciation, Methods of calculating depreciation (Straight line method, Declining balance method), After tax comparison of project.

**Text Books:**

1. Riggs, Bedworth and Randhwa, “Engineering Economics”, McGraw Hill Education India
2. Principles of Economics, Deviga Vengedasalam; Karunagaran Madhavan, Oxford University Press.
3. Engineering Economy by William G.Sullivan, Elin M.Wicks, C. Patric Koelling, Pearson
4. R. Paneer Seelvan, “Engineering Economics”, PHI
5. Ahuja, H.L., “Principles of Micro Economics”, S. Chand & Company Ltd
6. Jhingan, M.L., “Macro Economic Theory”
7. Macro Economics by S.P. Gupta, TMH

**MC 2: Business Communication (PMCMH103)**

**Course Objectives**:

1. To introduce students to various building blocks of communication, both within and outside their formal articulations.
2. To train students in the basic science of writing and help them use the same in various sites such as report, paragraph etc.
3. To create conditions in the classroom that encourages students to engage in meaningful conversation.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Develop skills to communicate effectively in formal settings.
2. Develop skills to write CV, Report, Minutes, Business Letters etc.
3. Develop skills to present effectively on topical issues.

**Course Prerequisites:**

This course does not require any prerequisite as such.

**Detail Syllabus**

**Module - I**

***Basics of Communication in Practice***

* 1. Types of Communication in an organization: Formal (internal and external) and Informal (grapevine)
  2. Communication Channels: Upward, Downward, Diagonal and Horizontal
  3. Introduction to cross-cultural communication.
  4. Bias-free communication & use of politically correct language in communication
  5. Importance of reading and ethics of writing
  6. Negotiation Skills, Argumentation & Consensus building.

**Module-II**

***Business Writing***

* 1. Skills of Writing: Coherence, Cohesion, Sentence Linkers, Clarity of Language and stylistic variation, process of writing.
  2. Paragraph writing: Topic Sentence, Supporting sentence & Concluding sentence, Logical structuring (Inductive approach and deductive approach)
  3. Letters, Applications
  4. Reports and Proposals
  5. Memos, Notices, Summaries, Abstracts& e-mails
  6. Writing a CV/Resumeˈ: Types of CV
  7. Writing a Cover letter

**Module -III**

***Speaking and Presentation***

* 1. Oral Presentation: 4 P’s of presentation, PPT
  2. Group Discussion: Structured and Un-structured, Various types of topics (abstract, absurd, contemporary etc.)
  3. Types of Interview: Preparing an Interview and techniques
  4. Grooming and dress code, Personality development

**Lab 1: Problem Solving and Program Design Laboratory (PLCCA101)**

***Note:*** This course shares the objectives and outcomes of its associated theory course PPCCA101. Suitable execution environment preferably Linux will be used to carry out laboratory exercises. Exercises will primarily follow algorithmic approach as provided in reference book serial number 1. The programs will follow proper modeling either function-oriented or object-oriented as the case may be. The exercises suggested below are illustrative in nature. Additional exercises suitably may be suggested by the faculty concerned to meet the course objectives.

**List of Exercises:**

1. Write a program that will exchange the values of three variables a, b and c as follows: the variable *b* will hold value of *a*, *c* will hold the value of *b* and *a* will hold the value of the variable *c*.
2. Write a program which reads a set of marks in an examination, count the number of pass marks, number of fail marks, percentage of pass and fail.
3. Write a program to find the harmonic mean of a set of n numbers.
4. Write a program to count the number of digits in an integer.
5. Write a menu based program to find LCM and GCD of a set of numbers
6. Write a program to convert binary numbers to octal and binary numbers to decimal.
7. Write a program to find the maximum, the minimum and how many times they both occur in an array of numbers.
8. Write a program to find the kth smallest element in an array of numbers.
9. Write a menu based program to implement sorting algorithms.
10. Write a program to perform operations such as multiplication and transpose on matrices.
11. Write programs involving string manipulation (to be decided by faculty).
12. Write programs involving pointer arithmetic (to be decided by the faculty)
13. Write a program involving recursive function (to be decided by the faculty)
14. Write programs involving structures and unions (to be decide by the faculty)
15. Write programs involving data files (to be decided by the faculty)
16. Write an Object Oriented Program to find the area and perimeter of a circle.
17. Write a menu based Object Oriented Program to perform operations in a bank account.
18. Write OO programs to implement overloading of function and constructor.
19. Write an OO program to process student results in an examination. Model students and marks scored in an examination as classes using has-a relation. Print the grade sheet of a student.
20. Write an OO program to find the area and perimeter of different shapes such as rectangles, triangles and squares. Use inheritance appropriately.

**Lab 2: Computer Organizations and Architecture Laboratory (PLCCA102)**

**Part-I:** Digital Logic Design Experiments:

Multiplexers & Decoders, Counters, Shift Registers, Binary Adders & Sub tractors, A L U

**Part-II:**

**A: 8085 Assembly Language Programming:**

8085 Assembly Language Programming using the 8085 µP Trainer Kit:

1. Write the working of 8085 µP Trainer and basic architecture of 8085 along with small introduction.
2. Study the complete instruction set of 8085 and write the instructions in the instruction set of 8085 along with examples.
3. Write an assembly language code using 8085 µP Trainer to implement data transfer instruction.
4. Write an assembly language code using 8085 µP Trainer to store numbers in reverse order in memory location.
5. Write an assembly language code using 8085 µP Trainer to implement arithmetic instruction.
6. Write an assembly language code using 8085 µP Trainer to add two numbers using lxi instruction.
7. Write an assembly language code using 8085 µP Trainer to add two 8 bit numbers stored in memory and also storing the carry.
8. Write an assembly language code using 8085 µP Trainer to implement logical instructions.
9. Write an assembly language code using 8085 µP Trainer to implement stack and branch instructions.

**B: Computer Organization:**

Pentium class PC architecture familiarization hardware & software parts demonstration, Troubleshooting of PC, Laptops, Server and Loading of Operating System, Antivirus and other software packages.

**Lab 3: Web Design and Development Laboratory (PLCCA103)**

Note: This course shares the objectives and outcomes of its associated theory course PPCCA103.

Suitable page authoring IDEs and development tools will be used to carry out laboratory exercises.

**List of Exercises:**

1. Develop a webpage to display your curriculum vitae. Use appropriate markups to structure the page.
2. Develop a user registration form. Use appropriate form controls for submitting user information.
3. Use CSS in the curriculum vitae developed earlier to enhance its presentation.
4. Use CSS in the registration form to enhance its appearance.
5. Use JavaScript to validate data of the registration form at the client-side.
6. Develop XML document for a list of Books in a library
7. Use AJAX to develop a webpage that fetches data from the server.
8. Use jQuery to display a list of students and their details in a table.
9. Use Bootstrap to develop a website for your department.
10. Publish the website on the Internet with basic search engine optimization for it.

**Lab 4: Business Communication Lab (PLCMH104)**

**Course Objectives**:

* To enable the students engage in polite, negotiating and argumentative conversation.
* To train the learners in writing CV, Report, Minutes, Business Letters etc.
* To give students an opportunity of power point presentation relating to topical issues.

There will be 10 lab sessions of 2 hours each. Lab sessions will be used to give the students an in-hand experience of communication taking place in an organization. This will help the students to understand the requirement of communication in the workplace. Students will be encouraged to brush-up themselves in activities based on all the modules of theory taught in the class room. Special emphasis will be given to speaking and writing business correspondences.

***List of Experiments:***

**Ist session:**

Speaking: Greeting an acquaintance/ friend, introducing oneself, introducing a third person to a friend, breaking off a conversation politely, leave-taking, Describing people, objects, places, processes etc. (1 Hour), Writing an application (1 Hour)

**IInd session:**

Speaking: making and responding to inquiries; expressing an opinion; expressing agreement/ disagreement, contradicting/ refuting an argument; expressing pleasure, sorrow, regret, anger, surprise, wonder, admiration, disappointment etc (1 Hour), Writing an informal letter/Business Letter (1 Hour)

**IIIrd session:**

Speaking: Narrating or reporting an event (1 Hour), Writing a Report (1 Hour)

**IVth session:**

Speaking: Ordering / directing someone to do something, Making requests; accepting / refusing a request, Expressing gratitude; responding to expressions of gratitude, Asking for or offering help; responding to a request for help, Asking for directions (e.g. how to reach a place, how to operate a device etc.) and giving directions, Speaking: asking for and granting/ refusing permission, prohibiting someone from doing something, suggesting, advising, persuading, dissuading, making a proposal, praising, complimenting, felicitating, expressing sympathy (e.g. condolence etc.), Complaining, criticizing, reprimanding etc., (1 Hour), Writing a proposal (1 Hour)

**Vth Session:**

Speaking: Understanding and interpreting graphs, flowcharts, pictograms, pictures, curves etc., (1 Hour), Writing: Describing, explaining and interpreting graphs, flowcharts, pictograms, pictures, curves etc.

**VIth session:**

Speaking: Group discussion (1 Hour), Writing a memo, notice and circular (1 Hour)

**VIIth session:**

Speaking: Public speaking, in-house communication on work-related situations (1 Hour), Writing a CV (1 Hour)

**VIIIth session:** Presentation 1 (Students will make and present a topic in power point on a pre-assigned topic) (1 Hour), Writing an e-mail (1 Hour)

**IXth session:** Presentation 2 (Students will make and present a topic in power point on a pre-assigned topic) (1 Hour), Writing an abstract (1 Hour)

**Xth session:** Presentation 3 (Students will make and present a topic in power point on a pre-assigned topic) (1 Hour), Writing a summary (1 Hour)

**Note: 70 marks will be devoted for sessions, 10 marks for record submission, 10 marks for viva-voce and 10 marks for project work.**

**End term assignment:** Students are required to make a review report of at least 5 pages on a topic of their own choice (The topic should be pre-approved by teacher).

**Semester-2**

**Core 5: Data Structures (PPCCA201)**

**Course Objectives:**

1. To provide knowledge and understanding of various basic and advanced data structures available in computing domain.
2. To provide skills to write programs to implement various data structures using procedural or object oriented programming languages.
3. To provide knowledge to analyze problems in application domains and design solution using data structures.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Be well-versed in various standard data structures available in computing domain.
2. Write programs to perform operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.
3. Analyze problems; choose the appropriate data structures and write program solutions to problems in specified applications using such data structures.

**Course Prerequisites:**

This course requires understanding of computer programming in any language.

**Detailed Syllabus:**

**Module I**

*Fundamentals:* Introduction to Data Structures, Classification of Data Structures, Algorithms, Measuring Space and Time Complexities, Asymptotic Notations, Abstract Data Types.

*Arrays:* Storage Structures for Arrays, Sparse Matrixes, Strings, Pattern Matching.

*Stacks and Queues:* Representation, Operations on Stacks and Queues, Applications of Stack and Queues.

*Linked Lists:* Dynamic Memory Management, Single Linked Lists, Double Linked Lists,

Circular Linked Lists, Linked Stacks and Queues, Operations on Polynomials.

**Module II**

*Trees:* Terminology, Representation, Binary Trees, Binary Search Trees, Searching, Insertion and Deletions Operations in a Binary Search Tree, Height Balanced Trees, M-way Search Trees, B-Trees, B+ Trees, General Trees, Representation of General Trees and Binary Trees, Forests, Application of Trees.

**Module III**

*Graphs:* Terminology, Representation, Path Matrix, Graph Traversal, Shortest Path Problems, Topological Sort.

*Searching and Sorting Techniques:* Linear and Binary Search, Bubble Sort, Insertion Sort, Selection Sort, Quick Sort, Merge Sort, Heap and Heap Sort, Radix Sort, Comparison of Sorting Techniques.

*Hashing:* Hash Functions and Hashing Techniques, External sorting.

**Text Books:**

1. Tremblay, Jean-Paul, and Paul G. Sorenson, "An introduction to data structures with applications", McGraw-Hill.
2. Aaron M. Tenenbaum, Data Structures Using C

**Reference Books:**

1. Richard F. Gilberg & Behrouz A. Forouzan, Data Structures A Psedocode Approach with C, Second Edition, CENGAGE Learning.
2. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, “Fundamentals of Data Structures in C”, Universities Press Pvt. Ltd.
3. Seymour, Lipchitz. “Data Structures with C.” TMH.

**Core 6: Programming with Java (PPCCA202)**

**Course Objectives:**

1. To provide elaborate knowledge on standard Java language.
2. To provide knowledge on Object Oriented Approach to program design.
3. To introduce JavaFX for GUI Development

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Develop skills to write Java programs to solve a variety of real-world problems.
2. Write programs using object oriented approach and standard Java
3. Develop client server applications using network sockets
4. Develop skills to write Desktop Applications involving GUI and Databases.
5. Design programs using readable, reusable and cohesive modules.

**Course Prerequisites:**

This course does not require any prerequisite as such.

**Detailed Syllabus:**

**Module I**

Features of Java, Data types, operators & expressions, control structures, arrays, Classes, objects & methods, constructors, garbage collection, access qualifiers, Overloading, String Handling – string operations, character extraction, string comparison, searching and modifying strings, String Buffer, String Builder, Packages, Interfaces, Wrapper classes, Static variables and methods.

**Module II**

Inheritance: single and multilevel inheritance, method overriding, abstract class, use of super and final keywords. Exception Handling: Exception types, uncaught exceptions, multiple catch clauses, nested try statements, built-in exceptions, creating your own exceptions. Multithreading: Java thread model, creating multiple threads, thread priorities, synchronization, inter-thread communication, suspending, resuming and stopping threads; Familiarity with Java Collection Framework.

I/O Streams: Console I/O, Files I/O – Byte Streams, Character Streams, Object Serialization; Socket Programming: TCP Socket, Datagram Socket.

**Module III**

JDBC programming: JDBC Drivers, Creating connection, executing queries and stored procedures, handling database transactions.

GUI Development: AWT Classes, Window fundamentals, working with graphics, working with color & fonts. Event handling in Java, Delegation Event Model, Swing Package: JFrame, JPanel, swing GUI controls, layout managers, working with menus, Introduction to JavaFX

**Text Books:**

1. Liang Y. Daniel, Introduction to Java Programming, Pearson Education.
2. Herbert Schildt, The Complete Reference Java 2, Tata McGraw Hill

**Reference Books:**

1. E. Balaguruswami, Programming with Java, Tata McGraw Hill.
2. Mughal K.A., Rasmussen R.W., A Programmer’s Guide to Java Certification, Addison-Wesley

**Core 7: Operating Systems (PPCCA203)**

**Course Objectives:**

1. To study the main components of an OS and their functions.
2. To study the concept of process, process management and CPU scheduling algorithms.
3. To study the concepts and implementation Memory management policies and virtual memory
4. To study the file system, its implementation and disk management.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Understand the managerial roles of OS for resource, file system, process, memory and I/O.
2. Understand the process management policies and scheduling of processes by CPU
3. Understand process synchronization and coordination handled by operating system
4. Understand and analyze the memory management and its allocation policies.
5. Conceptualize the components involved in designing a contemporary OS.

**Course Prerequisites:**

This course requires understanding of computer organization and programming in any language.

**Detailed Syllabus**

**Module I**

Operating System Introduction- Functions, Characteristics, Structures - Simple Batch, Multi programmed, timeshared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating-System services, System Calls, Virtual Machines. Process and CPU Scheduling - Process concepts and scheduling, Operation on processes, Cooperating Processes, Threads, and Interposes Communication Scheduling Criteria, Scheduling Algorithm, Multiple -Processor Scheduling, Real-Time Scheduling.

**Module II**

Process Management and Synchronization - The Critical Section Problem, Synchronization Hardware, Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors.

Deadlocks - System Model, Dead locks Characterization, Methods for Handling Deadlocks Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock.

**Module III**

Memory Management and Virtual Memory- Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with Paging. Demand Paging, Performance of Demanding Paging, Page Replacement, Page Replacement Algorithm, Allocation of Frames, Thrashing.

File System Interface and Implementation- Access methods, Directory Structure, Protection, File System Structure, Allocation methods, Free-space Management, Directory Management, Directory Implementation, Efficiency and Performance.

I/O Management – I/O software and its types, Disk Scheduling.

**Text Books:**

1. Operating System Concepts, by Abraham Silberschatz, Greg Gagne, and Peter Baer Galvin, Ninth Edition, John Wiley & Sons.

**Reference Books**

1. Operating Systems: Internals and Design Principles, by William Stallings, 8th edition Pearson Education Limited, 2014
2. Operating systems - A concept based Approach, by D.M Dhamdhere, 3rd Edition, Tata McGraw- Hill, 2012.
3. Operating systems, by Harvey M Deital, 3rd Edition, Pearson Education, 2011.

**Core 8: Database Management System (PPCCA204)**

**Course Objectives:**

1. To provide understanding on fundamental concepts of Relational database systems
2. To provide knowledge on Modeling and Design of Relational Databases
3. To provide elaborate knowledge on how to query databases
4. To provide understanding on database transactions

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Understand basic concepts of Relational databases management systems
2. Model data requirements of real-world applications
3. Develop databases for a variety of applications
4. Write SQL queries to perform simple to complex data manipulation tasks.

**Course Prerequisites:**

This course requires basic understanding of computer and programming.

**Detailed Syllabus:**

**Module-I**

***Introductory concepts of DBMS****:* Introduction and applications of DBMS, Purpose of data base, Data, Independence, Database System architecture- levels, Mappings, Database, users and DBA

***Relational Model****:* Structure of relational databases, Domains, Relations, Relational algebra – fundamental operators and syntax, relational algebra queries, tuple relational calculus

***Entity-Relationship Model****:* Basic concepts, Design process, constraints, Keys, Design issues, E-R diagrams, weak entity sets, extended E-R features – generalization, specialization, aggregation, reduction to E-R database schema.

**Module-II**

***Relational Database design****:* Functional Dependency – definition, trivial and non-trivial FD, closure of FD set, closure of attributes, irreducible set of FD, Normalization – 1NF, 2NF, 3NF, Decomposition using FD- dependency preservation, BCNF, Multi- valued dependency, 4NF, Join dependency and 5NF.

***SQL Concepts****:* Basics of SQL, DDL, DML, DCL, structure – creation, alteration, defining constraints –Primary key, foreign key, unique, not null, check, IN operator, Functions – aggregate functions, Built-in functions –numeric, date, string functions, set operations, sub-queries, correlated sub-queries, Use of group by, having, order by, join and its types, Exist, Any, All , view and its types; transaction control commands – Commit, Rollback, Savepoint; Introduction to PL/SQL Concepts: Cursors, Stored Procedures, Stored Function, Database Triggers

**Module-III**

***Query Processing & Query Optimization****:* Overview, measures of query cost, selection operation, sorting, join, evaluation of expressions, transformation of relational expressions, estimating statistics of expression results, evaluation plans, materialized views

***Transaction Management****:* Transaction concepts, properties of transactions, serializability of transactions, testing for serializability, System recovery, Two- Phase Commit protocol, Recovery and Atomicity, Log-based recovery, concurrent executions of transactions and related problems, Locking mechanism, solution to concurrency related problems, deadlock, two-phase locking protocol, Isolation, Intent locking

***Security****:* Introduction, Discretionary access control, Mandatory Access Control, Data Encryption

**Text Books:**

1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan, “Database Systems Concepts”, McGraw-Hill Education, New Delhi
2. Ramez Elmasri and Shamkant B. Navathe, “Fundamentals of Database Systems”, Pearson Education Inc., New Delhi.

**Reference Books**:

1. Hector Garcia-Molina, Jeffret D. Ullman, JennifferWidom, “Database Systems: A Complete Book”, Pearson Education Inc., New Delhi.
2. C. J. Date “An introduction to Database System”, Pearson Education Inc., New Delhi.
3. Bipin Desai, “An introduction to Database System”, Galgotia Publications.
4. Peter Rob & Carlos Coronel, “Database Systems: Design, Implementation, and Management”, CENGAGE Learning India Pvt. Ltd., New Delhi.
5. Mark L. Gillenson, “Fundamentals of Database Management Systems”, Wiley India Pvt. Ltd., New delhi.
6. Raghu Ramakrishnan, Johannes Gehrke, “Database Management Systems”, McGraw-Hill Education (India), New Delhi.

**Core 9: Software Engineering (PPCCA205)**

**Course Objectives:**

1. To provide understanding of software process, requirement engineering and design engineering.
2. To provide knowledge on UML to develop various models for the software-to-be.
3. To provide understanding of various software testing techniques and strategies.
4. To provide understanding of quality metrics for software product.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Understand software requirements and develop SRS.
2. Analyze requirements and develop various UML models for a software-to-be at hand.
3. Display ability to test software using appropriate techniques.
4. Access quality of software product.

**Course Prerequisites:**

This course requires basic understanding of computer and programming.

**Detailed Syllabus:**

**Module-I**

***Introduction:*** The evolving role of software, changing nature of software, software myths, Software engineering- a layered technology, a process framework.

***Process models:*** The waterfall model, incremental process models, evolutionary process models, specialized process models, the unified process, agile process models.

***Requirements Engineering:*** Requirement engineering tasks, functional and non-functional requirements, the software requirements specification (SRS), IEEE 830 guidelines, requirements validation, requirements management.

**Module-II**

***Building Analysis Models:*** Analysis Modeling Approaches: Structure Analysis, Object-Oriented Analysis; Data Modeling: E-R Diagram; Scenario-Based Modeling: Use-Case Model, Activity Diagram; Flow-Oriented Modeling: Data Flow Model, DFD, Class-Based Modeling: Identifying Analysis Classes, Class Model, CRC Model, Association and Dependency, Object Diagram; Behavior Modeling: State Diagram, Sequence Diagram, and Collaboration Diagram.

***Building Design Models:*** Design process and design quality, Design Concepts, Elements of Design Model, Software Architecture: importance of software architecture, architectural styles and patterns, architectural design; Component-level Design: What is a component, Basic Design Principles, Design guidelines, Cohesion, Coupling, Component Diagram, Deployment diagram; User Interface Design: The Golden Rules, interface design evaluation cycle.

**Module-III**

***Software Testing:*** Verification and Validation, Unit testing and Integration Testing, Validation testing, System testing, the Art of Debugging, Black-box and White-box testing, Basis path testing, Control structure testing, Graph-Based testing, Equivalence partitioning, Boundary value analysis, Object-Oriented testing, Scenario-Based Testing, Radom testing, Partition testing, Inter-class testing, GUI testing, Client Server testing.

***Product Metrics:*** Software quality factors, Attributes of effective software metrics, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

**Text Books:**

1. Roger S. Pressman, Software Engineering, A practitioner’s Approach, 6th Edition, McGraw Hill International Edition.
2. Rajib Mall, Fundamentals of Software Engineering, PHI, 2014.

**Reference Books:**

1. I. Sommerville, Software Engineering, 9th Edition, Pearson Education.
2. *Waman S. Jawadekar Software Engineering: Principles and Practice, Tata McGraw-Hill Education, 2004.*
3. Michael R. Blaha, and James R Rumbaugh, Object-Oriented Modeling and Design with UML, 2nd Edition 2005, Pearson Education.
4. Meilir Page-Jones, Larry L. Constantine, Fundamentals of Object-oriented Design in UML, Pearson Education.

**MC 3: Organizational Behaviour (PMCMH201)**

**Course Objectives:**

1. To develop an understanding of the behavior of individuals and groups inside organizations
2. To enhance skills in understanding and appreciating individuals, interpersonal, and group process for increased effectiveness both within and outside of organizations.
3. To develop theoretical and practical insights and problem-solving capabilities for effectively managing the organizational processes.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Compare and contrast theories of organizational behavior.
2. Analyze management issues as related to organizational behavior.
3. Assess the impact of culture on organizational behavior.
4. Understand organizational change and change management.

**Course Prerequisites:**

This course does not require any prerequisite.

**Detailed Syllabus:**

**Module-I**

***Fundamentals of OB:*** Definition, scope and importance of OB, Relationship between OB and the individual, Evolution of OB, Theoretical framework (cognitive), behavioristic and social cognitive), Limitations of OB.

***Attitude:*** Importance of attitude in an organization, Right Attitude, Components of attitude, Relationship between behavior and attitude, Developing Emotional intelligence at the workplace, Job attitude, Barriers to changing attitudes.

***Personality and values:*** Definition and importance of Personality for performance, The Myers-Briggs Type Indicator and The Big Five personality model, Significant personality traits suitable to the workplace (personality and job – fit theory), Personality Tests and their practical applications.

***Perception:*** Meaning and concept of perception, Factors influencing perception, Selective perception, Attribution theory, Perceptual process, Social perception (stereotyping and halo effect).

***Motivation:*** Definition & Concept of Motive & Motivation, The Content Theories of Motivation (Maslow’s Need Hierarchy & Herzberg’s Two Factor model Theory), The Process Theories (Vroom’s expectancy Theory & Porter Lawler model), Contemporary Theories – Equity Theory of Work Motivation.

**Module-II**

***Foundations of Group Behavior:*** The Meaning of Group & Group behavior & Group Dynamics, Types of Groups, The Five – Stage Model of Group Development.

***Managing Teams:*** Why Work Teams, Work Teams in Organization, Developing Work Teams, Team Effectiveness & Team Building.

***Leadership:*** Concept of Leadership, Styles of Leadership, Trait Approach Contingency Leadership Approach, Contemporary leadership, Meaning and significance of contemporary leadership, Concept of transformations leadership, Contemporary theories of leadership, Success stories of today’s Global and Indian leaders.

**Module-III**

***Organizational Culture:*** Meaning & Definition of Organizational Culture, creating & Sustaining Organizational Culture, Types of Culture (Strong vs. Weak Culture, Soft Vs. Hard Culture & Formal vs. Informal Culture), Creating Positive Organizational Culture, Concept of Workplace Spirituality.

***Organizational Change:*** Meaning, Definition & Nature of Organizational Change, Types of Organizational Change, Forces that acts as stimulants to change.

***Implementing Organizational Change:*** How to overcome the Resistance to Change, Approaches to managing Organizational Change, Kurt Lewin’s-Three step model, Seven Stage model of Change & Kotter’s Eight-Step plan for Implementing Change, Leading the Change Process, Facilitating Change, Dealing with Individual & Group Resistance, Intervention Strategies for Facilitating Organizational Change, Methods of Implementing Organizational Change, Developing a Learning Organization.

**Text Books:**

1. Understanding Organizational Behaviour, Parek, Oxford
2. Organizational Behaviour, Robbins, Judge, Sanghi, Pearson.

**Reference Books:**

1. Organizational Behaviour, K. Awathappa, HPH.
2. Organizational Behaviour, VSP Rao, Excel
3. Introduction to Organizational Behaviour, Moorhead, Griffin, Cengage.
4. Organizational Behaviour, Hitt, Miller, Colella, Wiley

**Lab 5: Data Structures Lab (PLCCA201)**

***Note:*** This course shares the objectives and outcomes of its associated theory course PPCCA201. Suitable programming language preferably object-oriented will be used to carry out laboratory exercises. The exercises suggested below are illustrative in nature. Additional exercises may be suggested by the faculty concerned to meet the course objectives.

**List of Exercises:**

1. Implement Stack and use it for evaluation of post-fix expression.
2. Implement conversion of prefix expression into post-fix form using recursion.
3. Implement circular queue (using array) with menu options like insert, delete, display and exit.
4. Implement a priority queue (using pointers) and use it to organize student records prioritized by marks.
5. Implement doubly linked circular list to hold strings and use it for organizing a sequence of cities constituting at our program.
6. Implement of a binary search tree with menu options: Construct a tree, insert a node, delete a node, traverse and display preorder, in order and post order sequence of its nodes.
7. Implement of di-graphs using adjacency matrix and find the transitive closure using Warshall’s algorithm.
8. Implement a weighted graph and find minimal cost spanning tree using PRIM’s Algorithm.
9. Generate 70 random integers in a given range and sort them using quick sort. Apply both binary search and Interpolation search to locate a given integer and compare the search algorithms based on the number of comparisons / probes required or a successful as well as unsuccessful search.
10. Implement Heap Sort, Merge Sort and other sorting algorithms on the above random numbers.
11. Implement a small Real World Application illustrating DS usage.

**Lab 6: Programming with Java Lab (PLCCA202)**

***Note:*** This course shares the objectives and outcomes of its associated theory course PPCCA202. Suitable IDE will be used to carry out laboratory exercises. The programs will follow proper object-oriented modeling. The exercises suggested below are illustrative in nature. Additionally, suitable exercises may be suggested by the faculty concerned to meet the course objectives.

**List of Exercises:**

1. Develop an Object Oriented Program to find the area and perimeter of a circle.
2. Develop an interest calculator program to find simple interest payable monthly, compound interest payable annually compounded quarterly. Use keyboard inputs for interest rate and principal amount.
3. Define a class to calculate professional tax on a salary amount based on the following tax rate. Use if and switch control structures.

|  |  |
| --- | --- |
| Salary Slab | Tax Rate |
| Up to Rs. 10000.00 | Nil |
| Between Rs. 10001.00 – Rs. 25000.00 | Rs. 100.00 |
| Between Rs. 25001.00 – Rs. 50000.00 | Rs. 200.00 |
| Between Rs. 50001.00 – Rs. 75000.00 | Rs. 300.00 |
| Between Rs. 75001.00 – Rs. 100000.00 | Rs. 450.00 |
| Above Rs. 100000.00 | Rs.650.00 |

1. Develop a text-menu based program to compute area and perimeter of a circle, rectangle, square and a right angle triangle.
2. Develop a menu based program to perform operations in a bank account.
3. Develop a program to find the sum of even numbers and sum of odd numbers in a set of numbers. Define a class with suitable methods to carry out the operations. Use array to store numbers.
4. Modify the class defined in sl-6 to find largest and smallest numbers in a set of numbers.
5. A student scores marks in subjects in a semester. A semester has 5 or 6 subjects depending of MCA course. Define a class called Score that contains subject code, name and marks in that subject. Define a class called Student having an array of objects of Score class in it following object composition. Process result of students in different semesters.
6. Define a class called SimpleMath with overloaded methods to carryout arithmetic operations using it. Use static methods appropriately.
7. Redefine the Circle class in exercise 1 to use value of Pi as a constant and a variable to count number of instances created as you go on creating objects.
8. Develop a class to perform the following tasks on a line of text
   1. Count the number of words in the text
   2. Searches a particular string in the text
   3. Checks if the text is a palindrome
9. A library is a collection of books. Generally, a book is authored by one or more authors. Develop a program to add books and display a list of books when searched by author name. Consider title, ISBN number, publisher name, publication year for book; designation, organization, and country for author.
10. In CET, two types of people are there: students and employees. As per Govt. of India, everyone must have his or her AADHAR number for unique identification. Model these objects appropriately using inheritance and create an array of people with several students and employees in it. Write a program to search a student or an employee based on AADHAR number and print its details.
11. There are different shapes such as circles, rectangles and squares and need to be kept track of if they are painted or not with colors. Create a collection of shapes to be painted. Consider some cost of a color per square unit of the shape area. Write a program to calculate the painting cost of different shapes. Print the list of shapes which are not painted. Also print the list of painted shapes with their color, painting cost and area.
12. A bank account maintains a minimum balance. If the account balance comes down below this level due to some withdrawal, then it raises warning and disallows the operation. Define a custom exception class called “Insufficient Fund Exception” which will be raised when such event occurs. Also use the built-in exception class “Illegal Argument Exception” which is to be raised when you try to either withdraw or deposit an amount less than or equal to zero.
13. Write a multithreaded program to perform following parallel operations on a set of numbers.
    * 1. Find the largest number
      2. Find the sum of the number
      3. Sort the numbers
14. Write program using console I/O.
15. Write Programs using File I/O.
16. Write program using serialization.
17. Write client server programs using Java sockets.
18. Write JDBC programs to perform CRUD operations.
19. Write JDBC program to execute stored procedures.
20. Write GUI programs using basic swing classes
21. Write GUI program involving Menus
22. Write a GUI program using JavaFX

**Lab 7: Operating System Laboratory (PLCCA203)**

***Note:*** This course shares the objectives and outcomes of its associated theory course PPCCA203. Suitable operating system environment and programming language will be used to carry out laboratory exercises. The exercises suggested below are illustrative in nature. Additional exercises may be suggested by the faculty concerned to meet the course objectives.

***List of Experiments:***

1. Learn and practice basics of UNIX commands.
2. Learn and write programs using Shell Programming.
3. Implement Shared memory and IPC
4. Implement Threading and Process Synchronization Applications
5. Implement Semaphores
6. Implement the following CPU scheduling algorithms a) Round Robin b) SJF c) FCFS d) Priority
7. Implement Bankers Algorithm for Dead Lock Avoidance
8. Implement an Algorithm for Dead Lock Detection
9. Implement Paging Technique of memory management.
10. Implement page replacement algorithms: FIFO, LRU, LFU
11. Implement file allocation strategies: Sequential, Indexed, Linked
12. Implement File Organization Techniques: Single level directory, Two level, Hierarchical, DAG

**Lab 8: Database Management Systems Lab (PLCCA204)**

This course shares the objectives and outcomes of its associated theory course PPCCA204. Suitable RDBMS such as ORACLE or MySQL will be used to carry out laboratory exercises. The exercises suggested below are illustrative in nature. Additional exercises may be suggested by the faculty concerned to meet the course objectives.

***List of Exercises:***

1. Create ER Models for the following: Inventory Control System, Hospital Management System, Hotel Management System, Timetable Management System, Railway Reservation System
2. Derive Database Schema from the ER Models of the above systems
3. Create a database and set constraints, relationships.
4. Perform Insertion, Deletion, Modifying, Altering, Updating and Viewing records based on conditions.
5. Write SQL queries using aggregate functions.
6. Write SQL queries involving sub-queries
7. Write SQL queries using group by, order by, having etc.
8. Write SQL queries involving joins.
9. Create Views, Synonyms, Sequence, Indexes, savepoint.
10. Write a PL/SQL block to satisfy some conditions by accepting input from the user.
11. Write a PL/SQL block that handles all types of exceptions.
12. Create Stored Procedures.
13. Create database triggers and functions

**Lab 9: Software Engineering Lab (PLCCA205)**

This course shares the objectives and outcomes of its associated theory course PPCCA205.

Suitable UML tools will be used to carry out laboratory exercises.

***List of Exercises:***

1. Analyze the requirements and develop SRS for a software system. The students are to perform in groups and each group will take one application and present the SRS.
2. Draw DFD for the system. Identify various processes, data store, input, output etc. of the system and Use processes at various levels to draw the DFDs.
3. Identify the data persistence requirements and develop E-R diagram for the system
4. Identify various functions that the user will perform with the system. Develop Use-Case Model for the system.
5. Develop specification for each use-case.
6. Develop sequence diagram and collaboration diagram for each use case following the specification.
7. Find Analysis class from the sequence diagrams and Develop class diagram and object diagram
8. Develop activity diagram and state-chart diagram for system
9. Develop component diagram and deployment diagram to show implementation view of the system
10. Use a suitable tool such as Winrunner to perform unit testing and integration testing.

**Semester-3**

**Core 10: Design and Analysis of Algorithms (PPCCA301)**

**Course Objectives:**

1. To understand basics of algorithmic for problem solving.
2. To provide knowledge on algorithm design techniques.
3. To provide knowledge on complexity analysis.
4. To introduce some important algorithms.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Design new and efficient algorithms for specific problems.
2. Learns various algorithm design techniques.
3. Learn complexity analysis.

**Course Prerequisites:**

1. This course does not require any prerequisite.

**Module I**

Introduction, Growth of Functions (Asymptotic notations, standard notations and common functions), Recurrences, solution of recurrences by substitution, recursion tree and Master methods, Divide and conquer algorithms: Merge sort, Quick sort, Strassen’s Matrix Multiplication, Heap sort: Heaps, Building a heap, The heap sort algorithm.

**Module II**

Dynamic programming algorithms (Matrix-chain multiplication, All- pairs shortest paths (Floyd – Warshall), Single source shortest paths (Bellman-Ford Algorithm), Longest common subsequence, Assembly-line scheduling, Greedy Algorithms - (Achivity- selection Problem, Fractional knapsack problem, Huffman codes, Prim’s algorithm- Kruskal’s Algorithm- Dijkstra’s Algorithm).

**Module III**

BRUTE FORCE– Closest-Pair and Convex-Hull Problems-Exhaustive Search – Traveling Salesman Problem, Backtracking – n-Queens problem, Graph coloring, Hamilton Cycle; branch and bound: 15-puzzle problem.

String matching (Rabin-Karp algorithm)

**Module IV**

Linear Programming: The Simplex Method; The Maximum-Flow Problem: The Ford-Fulkerson Method; NP Completeness; Approximate algorithms: Traveling Salesman Problem, vertex cover, set cover.

**Text Books:**

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, “Introduction to Algorithms”, Third Edition, PHI Learning Private Limited, 2012.
2. ELLIS HOROWITZ and SARTAJ SAHNI. Fundamentals of Computer Algorithms. 2nd Edition, Universities Press

**References:**

1. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, “Data Structures and Algorithms”, Pearson Education, Reprint 2006.
2. Donald E. Knuth, “The Art of Computer Programming”, Volumes 1& 3 Pearson Education, 2009. Steven S. Skiena, “The Algorithm Design Manual”, Second Edition, Springer, 2008
3. Anany Levitin, “Introduction to the Design and Analysis of Algorithms”, Third Edition, Pearson Education, 2012.
4. http://nptel.ac.in/

**Core 11: Theory of Computation (PPCCA302**)

**Course Objectives:**

1. To understand how theoretical machine is designed.
2. To provide knowledge on theoretical machine design techniques.
3. To provide knowledge on complexity class of problems.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Acquire knowledge on theoretical machine design.
2. Realize the difference between determinism and non-determinism.
3. Get idea on recursive properties of languages.

**Course Prerequisites:**

1. This course requires data structures as prerequisite.

**Module I**

Alphabet, languages and finite automata (deterministic and nondeterministic), Minimization of finite automata. DFA/NFA to regular expression and vice versa using Arden’s Formula.

grammars: Production rules and derivation of languages. Chomsky’s hierarchy of languages and Grammars. Regular grammars, regular expressions. Closure and decision properties of regular languages. Pumping lemma of regular sets.

**Module II**

Context free grammars and pushdown automata. Chomsky and Griebach normal forms. Parse trees, Cook- Younger- Kasami-parsing algorithms. Ambiguity and properties of context free languages. Pumping lemma, Parikh's theorem. Deterministic pushdown automata, closure properties of deterministic context free languages.

**Module III**

Turing machines and variation of Turing machine model, Turing computability, Type of languages. Linear bounded automata and context sensitive languages. Primitive recursive functions.

Gödel numbering. Ackermann's function, recursiveness of Ackermann and Turing computable functions. Church Turing hypothesis. Recursive and recursively enumerable languages.

**Module IV**

Universal Turing machine and undecidable problems. Undecidability of Post correspondence problem. Valid and invalid computations of Turing machines and some undecidable properties of context free language problems. Time complexity class P, class NP, NP completeness.

**Text Books:**

1. Introduction to the theory of computation: Michael Sipser, Cengage Learning
2. Introduction to Automata Theory, Languages and Computation: J.E. Hopcroft and J.D Ullman, Pearson Education, 3rd Edition.

**Reference Books:**

1. Automata Theory: Nasir and Srimani, Cambridge University Press.
2. Introduction to Computer Theory: Daniel I.A. Cohen, Willey India, 2nd Edition.

**Core 12: Computer Networks (PPCCA303)**

**Course Objectives:**

1. To provide students with broad concepts and fundamentals of computer networks.
2. To familiarize students with the layered approach to computer network.
3. To provide adequate knowledge on issues and protocols involved in different layers of network.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Understand the basic concepts of computer network and data communication.
2. Understand the functions of each layer in the OSI and TCP/IP reference model.
3. Understand the working of essential protocols of computer networks, and how they can be applied in network design and implementation.

**Course Prerequisites:**

This course requires understanding of computer programming and Data structures.

**Detail Syllabus**

**Module I**

Network architecture, Layers, Transmission Media, Data Link Layer: Issues in the data link layer, Framing, Error detection and correction, Link-level Flow Control, Medium access, CSMA, Ethernet, Token ring, FDDI, Wired LAN, Wireless LAN

**Module II**

Connecting Devices, Bridges and Switches, Circuit switching vs. packet switching, Packet switched networks, Network Layer: Design Issues, Logical Addressing, Subnetting, CIDR, IPv4, IPv6, Address Mapping, ARP, RARP, DHCP, ICMP; Delivery, Forwarding, Routing algorithms, RIP, OSPF, BGP –Multicasting – Congestion avoidance in network layer

**Module III**

Transport Layer: Process-to-process delivery, UDP, TCP, Adaptive Flow Control, Adaptive Retransmission, Congestion control, Congestion avoidance and QoS

Application Layer: Email (SMTP, MIME, IMAP, POP3), Remote Logging (Telnet), File Transfer (FTP), WWW and HTTP, Domain Name System (DNS), Network management (SNMP)

**Text Books:**

1. Data Communications and Networking by Behrouz A. Forouzan. Third Edition, TMH.

**Reference Books:**

1. Computer Networks by Andrew S Tanenbaum, David. j. Wetherall, 5th Edition. Pearson Education/PHI
2. Computer Networks: A Systems Approach by Larry L. Peterson, Bruce S. Davie, Morgan Kauffmann Inc., 2003.
3. Computer and Communication Networks by Nader F. Mir, Pearson Education, 2007
4. Data and Computer Communication, William Stallings, Sixth Edition, Pearson Education, 2000

**Core 13: Cryptography and Network Security (PPCCA304)**

**Objective**

1. To provide an understanding of requirement, different approaches and major issues in information security.
2. To develop a basic understanding of cryptography and its use in maintaining security.
3. To provide knowledge of computer security technologies used in computer operating systems, distributed systems, networks and representative applications.
4. To provide an understanding of different issues of overall social, economic and professional contexts, and getting aware of the ethical and legal responsibilities related to computer security.

**Course Outcomes:**

1. To understand about different external and internal threats to an organization.
2. To understand different procedures used to discover, analyze and how to deal with threats to an organization.
3. To understand and apply the cryptography techniques to transmit data securely.
4. To understand network security threats and solutions to handle those.
5. To understand security related legal and regulatory issues.

**Course Prerequisite:**

This course requires understanding of computer organization, operating system, network and discrete mathematics.

**Module-I**

***Introduction to Cryptography:*** Security goals, cryptography attacks, services & mechanisms, techniques.

***Symmetric-Key Encipherment:*** Mathematics of cryptography, Traditional Symmetric-key Ciphers, Data Encryption Standard- Introduction, structure, analysis & security, Advanced Encryption Standard –Introduction, Transformations, key expansion, AES ciphers

**Module-II**

***Asymmetric-Key Encipherment:*** Mathematics of Asymmetric-Key Cryptography: primes, primality testing, factorization, Chinese remainder theorem, quadratic congruence, exponentiation and logarithm; Asymmetric-Key Cryptography: RSA cryptosystem, Rabin cryptosystem

***Integrity, Authentication and Key Management:*** Message Integrity, Message Authentication, Cryptography Hash Functions – MD4 Hash, SHA-512, Digital Signature, Entity Authentication, Key Management: Symmetric key distribution- KDC, Kerberos, Symmetric key agreement – Diffie-Hellman, Public Key Distribution – X.509

**Module-III**

***Network Security:*** Authentication Service – Electronic Mail Security – PGP – S/MIME - IP Security – Web Security.

***System Security:*** Intrusion detection – password management – Viruses and related Threats – Virus Counter measures – Firewall Design Principles – Trusted Systems.

**Text Books:**

1. Behrouz A. Forouzan & Debdeep Mukhopadhayay, “Cryptography & Network Security”, McGraw Hill, New Delhi

**Reference Books:**

1. William **Stallings** & Lawrie **Brown**, “Computer Security: Principles and Practice”, Pearson Education, Inc. New Delhi.
2. Charlie **Kaufman**, Radia **Perlman** & Mike **Speciner**, “Network Security: Private Communication in a Public World”, 2nd Edition, 2003, PHI Learning. New Delhi.

**Core 14: Data Science with Python (PPCCA305)**

**Course Objectives:**

1. To provide introduction to the Python programming language
2. To focus on fundamentals and core concepts of Python
3. To introduce data analysis and visualization using Python

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Get a detailed understanding of Python language
2. Develop skills to write Python programs in broad application areas.
3. Perform data analysis and visualization using Python programs

**Course Prerequisites:**

This course requires understanding of computer and programming in any language.

**Detail Syllabus**

**Module I**

***Introduction:*** What is data Science? Data Science Pipeline, Role of Python in Data Science, Python features, Basic Syntax: variable, data types, expressions, assignment statement, operators, data input and output using input and print functions, formatted output, type conversion, escape sequence, comments in the program, using functions and modules of Python library, Python program format and structure, writing and executing Python program; Control statements: if, if-else, nested if, for loop, while loop, break, continue, pass.

***Strings and text files:*** String manipulations, subscript operator, indexing, slicing a string; strings and number system: converting strings to numbers and vice versa; text files: reading/writing text and numbers from/to a file; creating and reading a formatted file (csv or tab-separated); manipulating files and directories, OS and SYS modules.

***Lists, Sets, Tuples, and Dictionaries:*** basic list operations, replacing, inserting, removing an element; searching and sorting lists; dictionary, adding and removing keys, accessing and replacing values; traversing dictionaries.

***Regular Expressions:*** re module, pattern-string syntax, find, match, search, split, sub functions,

**Module II**

***Functions:*** defining and calling functions, arguments and return values, Recursive functions.

***Classes and OOP:*** defining classes, attributes and methods, constructor, static and class methods, inheritance, polymorphism, operator overloading, exception handling, try, except, raise, assert, finally.

***GUI Programming:*** Event-driven programming, tkinter module, GUI Basics, creating simple GUI, buttons, labels, entry fields, dialogs, widget attributes - sizes, fonts, colors layouts, nested frames.

***Database and Persistence Programming:*** Persistence options in Python, Using DBM Files, Using Object Pickling, Using Shelves, Data Persistence in RDBMS, creating connection, executing queries, processing query results.

**Module III**

***Numerical Python:*** NumPy basics, creating ndarray, data types for ndarray, indexing and slicing, basic operations and manipulations on n-dimensional array.

***Data Analysis:*** introduction to Pandas data structures, Series, Data Frames, indexing, selection, filtering, sorting, ranking, handling missing values, data aggregation, plotting with Matplotlib

**Text Books:**

1. The Fundamentals of Python: First Programs, Kenneth A. Lambert, 2nd Edition, Cengage Learning, 2011
2. Python Data Science Handbook, First Edition, Jake VanderPlas, O’Reilly Media, 2017

**Reference Books:**

1. Programming in Python 3, Second Edition, Mark Summerfield, Addition-Wesley, 2010
2. Learning Python, by Mark Lutz, 5th Edition, O’Reilly Media, 2013
3. Python Programming for the Absolute Beginner, Third Edition, Michael Dawson, Cengage Learning, 2010
4. Python for Data Analysis by Wes McKinney, O’Reilly Media, 2013
5. Python in a Nutshell, by Alex Martelli, Anna Ravenscroft & Steve Holden, 3rd Edition, O’Reilly Media, 2017

**Project 1: Minor Project-1 (PPRCA306)**

**[To be decided by the department]**

**Lab 10: Computer Networks Lab (PLCCA301)**

***Note:*** This course shares the objectives and outcomes of its associated theory course PPCCA303. The exercises suggested below are illustrative in nature. Appropriate exercises may be suggested by the faculty concerned to meet the course objectives.

**List of Exercises:**

1. Implementation of Stop and Wait Protocol and Sliding Window Protocol.
2. Study of Socket Programming and Client – Server model
3. Write a code simulating ARP /RARP protocols.
4. Write a code simulating PING and TRACEROUTE commands
5. Create a socket for HTTP for web page upload and download.
6. Write a program to implement RPC (Remote Procedure Call)
7. Implementation of Sub-netting.
8. Applications using TCP Sockets like

Echo client and echo server, Chat, File Transfer,

1. Applications using TCP and UDP Sockets like DNS, SNMP, File Transfer
2. Study of Network simulator (NS) and Simulation of Congestion Control Algorithms using NS/NetSim
3. Perform a case study about the different routing algorithms to select the network path with its optimum and economical during data transfer: Link State routing, Flooding, Distance vector

**Lab 11: Cryptography and Network Security Laboratory (PLCCA302)**

***Note:*** This course shares the objectives and outcomes of its associated theory course PPCCA304. The exercises suggested below are illustrative in nature. Appropriate exercises may be suggested by the faculty concerned to meet the course objectives.

***List of Exercises***

Perform the following experiments using C/Java language.

1. Write a program to perform encryption and decryption using the following substitution & transposition techniques concepts:
   1. Caesar Cipher
   2. Hill Cipher
2. Write a program to perform encryption and decryption using the following substitution & transposition techniques concepts:
   1. Vigenere Cipher
   2. Rail fence – row & Column Transformation
3. Write a program to implement the DES algorithm.
4. Write a program to implement the Rijndael algorithm.
5. Write a program to implement the RSA algorithm.
6. Write a program to implement the Digital Signature scheme.
7. Write a program to implement the MD5 algorithm.
8. Write a program to implement the SHA-512 algorithm.
9. Write a program to implement the Diffiee-Hellman algorithm.
10. Demonstrate intrusion detection system (ids) using any available software tool (e.g. snort).

**Lab 12: Data Science with Python Lab (PLCCA303)**

***Note:*** This course shares the objectives and outcomes of its associated theory course PPCCA305. Suitable IDE will be used to carry out laboratory exercises. The exercises suggested below are illustrative in nature. Appropriate exercises may be suggested by the faculty concerned to meet the course objectives.

***List of Exercises:***

1. Write program using input and output functions.
2. Write programs using control statements.
3. Write programs involving string manipulations.
4. Write program to read from and write to text file.
5. Write program to create and read from CSV file.
6. Write program involving regular expression.
7. Write programs to perform operations on lists.
8. Write programs to perform operations on dictionary.
9. Write program involving function.
10. Write program involving classes and objects.
11. Write program involving inheritance.
12. Write program involving operator overloading
13. Write program involving exception handling.
14. Write program to perform CRUD operations on SQL database.
15. Write GUI programs using basic UI objects.
16. Write programs to perform operations on n-dimensional array.
17. Write programs to perform operations on series.
18. Write programs to perform operations on data frames.
19. Write programs involving data aggregation.
20. Write programs using plotting functions.

**Semester-4**

**Core 15: Artificial Intelligence (PPCCA401)**

**Course Objectives:**

The primary objective of this course is to introduce the basic principles, techniques, and applications of Artificial Intelligence. More specifically:

1. To introduce AI and its basic principles towards problem solving using various search techniques.
2. To provide understanding on knowledge representation and reasoning techniques used in intelligent systems.
3. To introduce machine learning, natural language processing and perception.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Understand basic principles and techniques of AI towards problem solving
2. Demonstrate their proficiency in knowledge representation in Intelligent systems
3. Demonstrate understanding of basic learning, communication and perception in intelligent system.

**Course Prerequisites:**

This course requires basic knowledge of computer algorithms and data structures.

**Detailed Syllabus**

**Module-I**

Artificial Intelligence: Introduction, Intelligent Agents: Agents & Environments, Concept of Rationality, Nature & Structure of Agents; Problem Solving: Solving Problems by Searching, Classical Search, Adversarial Search, Constraint Satisfaction Problems. Knowledge, Reasoning and Planning: Logical agents, First order logic, Inference in First order logic.

**Module-II**

Classical planning, Knowledge Representation; Uncertain Knowledge and Reasoning: Probabilistic Reasoning, Learning from Examples, Knowledge in Learning; Natural Language Processing: Language models, Text Classification, information retrieval, information extraction

**Module-III**

Natural Language for Communication: Phrase structure Grammars, Syntactic Analysis, Augmented grammars and semantic interpretation, Machine translation, Speech recognition; Perception; Expert Systems: Introduction, Design of Expert systems.

**Text books:**

1. Stuart Russell and Peter Norvig, “Artificial Intelligence: A Modern Approach”, Third Edition, 2010, Pearson Education, New Delhi.

Chapters: 1, 2, 3, 4 (4.1, 4.2), 5 (5.1, 5.2, 5.3), 6, 7, 8, 9, 10 (10.1, 10.2, 10.3, 10.5), 12, 14 (14.1-14.6), 18 (18.1- 18.7), 19 (19.1, 19.2, 19.3), 22, 23, 24 (24.1-24.3, 24.5).

**Reference books;**

1. Elaine A. Rich and Kevin Knight, “Artificial Intelligence”, 3rd Edition, 2009, McGraw-Hill Education (India), New Delhi.
2. Nills J. Nilsson, “Artificial Intelligence: A New Synthesis”, 2nd Edition, 2000, Elsevier India Publications, New Delhi.
3. Michael Negnevitsky, “Artificial Intelligence: A Guide to Intelligent Systems”, Second Edition, 2005, Pearson Education, Inc. New Delhi.
4. Dan W. Patterson, “Introduction to Artificial Intelligence and Expert Systems”, 1st Edition, 1996, PHI Learning Pvt. Ltd., New Delhi.
5. Ben Coppin, “Artificial Intelligence Illuminated”, 2005, Narosa Publication, New Delhi. ISBN: 978-81-7319-671-3
6. Joseph Giarratano and Gary Riley, “Expert Systems: Principles and Programming”, Fourth Edition, CENGAGE Learning India Pvt. Ltd., New Delhi.  
   Chapters: 1 and 6.

**Core 16: Enterprise Java Technologies (PPCCA402)**

**Course Objectives:**

1. To provide adequate knowledge on basic Server-side Java technologies such as Servlet and JSP.
2. To understand Java frameworks such as JSF and JPA and their use.
3. To understand web services, EJBs and how to use them in distributed n-tier applications.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Develop skills to write n-tier web applications for a variety of real-world problems.
2. Develop applications using Servlets, JSP and Java Bean following MVC model
3. Develop web applications using JSF framework
4. Develop distributed applications involving JPA, web service and EJB.

**Course Prerequisites:**

1. This course requires standard Java Programming as prerequisite.

**Detail Syllabus**

**Module I**

Enterprise Application Architecture, Enterprise Java Technologies, Web Applications, Servlet Overview, Servlet API, Writing HelloWorld Program using Servlet, Servlet Life Cycle, Configuring Servlet in web.xml, Retrieving information from Request object, HTML form processing using Servlet, Servlet Initialization, Session tracking, Cookies, Database Access using Servlet, Error Handling, Servlet Collaboration, Forward verses Redirect.

Overview of JSP, JSP Advantages, JSP Application Models: JSP Model1 and Model 2 architectures, Life Cycle of a JSP page, JSP Elements, JSP Comments, Scripting in JSP, Directives, Implicit Objects, Action Tags, JSP and Java Beans, Introduction to JSTL, Introduction to JSP Expression Language.

**Module II**

Introduction to JSF, Features, Benefits of JSF, JSF Architecture, JSF Elements, Request Processing Life Cycles, JSF HTML tags, JSF Core tags, Standard UI components, Managed Beans, Event handling, Page Navigation, convertors, validators, Expression Language, Using AJAX with JSF, sending AJAX Request;

**Module III**

Enterprise JavaBeans Technology: EJB Component Architecture, Role of EJB & its life cycle, Types of Beans, Session Beans, Stateless and Stateful beans, Message Driven Bean, Life Cycle, Managing Transactions in EJB;

Understanding Java Persistence: Object Relational Mapping, Java Persistence API, Benefits, components of JPA, Entity, Entity manager, Persistence unit, Life cycle of Entity, Entity Relationships, querying entities, Java Persistence Query Language, performing CRUD operations using JPA; Introducing Hibernate.

Overview of SOA, Web Services, Types of Web Service, Building Web services with JAX-WS;

**Text Books:**

1. Java Server Programming (Java EE 7) Black Book, by DT Editorial Services, Dreamtech Press, 2015.

**References:**

1. Eric Jendrock, Ricardo Cervera-Navarro, Ian Evans, Kim Haase, William Markito, “The Java EE 7 Tutorial”, 5th Edition, Addison-Wesley Professional, Pearson India, 2014.
2. Advanced Java Technology by MT Savaliya, Dreamtech Press, 2015.
3. David Geary, Cay S. Horstmann, “Core Java Server Faces”, Third Edition, 2010, Pearson Education, Inc. New Delhi.

**Core 17: Compiler Design** **(PPCCA403)**

**Objectives**: the student should be made to:

1. Learn The Design Principles of a Compiler.
2. Learn The Various Parsing Techniques and Different Levels of Translation.
3. Learn How to Optimize and Effectively Generate Machine Code

**Outcomes**:

At the end of the course, the student should be able to:

1. Design and Implement a Prototype Compiler.
2. Apply The Various Optimization Techniques.
3. Use The Different Compiler Construction Tools.

**Module – I**

***Introduction****:* Overview and phases of compilation. (2-hours)

***Lexical Analysis:*** Non-deterministic and deterministic finite automata (NFA & DFA), regular grammar, regular expressions and regular languages, design of a lexical analyser as a DFA, lexical analyser generator. (3-hours)

***Syntax Analysis:*** Role of a parser, context free grammars and context free languages, parse trees and derivations, ambiguous grammar. Top Down Parsing: Recursive descent parsing, LL (1) grammars, non-recursive predictive parsing, error reporting and recovery. Bottom Up Parsing: Handle pruning and shift reduces parsing, SLR parsers and construction or SLR parsing tables, LR (1) parsers and construction of LR (1) parsing tables, LALR parsers and construction of efficient LALR parsing tables, parsing using ambiguous grammars, error reporting and recovery, parser generator. (8-hours)

**Module – II**

***Syntax Directed Translation:*** Syntax directed definitions (SDD), inherited and synthesized attributes, dependency graphs, evaluation orders for SDD, semantic rules, application of syntax directed translation. (5-hours)

***Symbol Table:*** Structure and features of symbol tables, symbol attributes and scopes. (2-hours)

***Intermediate Code Generation:*** DAG for expressions, three address codes - quadruples and triples, types and declarations, translation of expressions, array references, type checking and conversions, translation of Boolean expressions and control flow statements, back patching, intermediate code generation for procedures. (7-hours)

**Module – III**

***Run Time Environment:*** storage organizations, static and dynamic storage allocations, stack allocation, handlings of activation records for calling sequences. (3-hours)

***Code Generations:*** Factors involved, registers allocation, simple code generation using stack allocation, basic blocks and flow graphs, simple code generation using flow graphs. (3-hours)

***Elements of Code Optimization:*** Objective, peephole optimization, concepts of elimination of local common sub-expressions, redundant and un-reachable codes, basics of flow of control optimization. (2-hours)

**Text Book:**

1. Alfred V Aho, Monica S. Lam, Ravi Sethi and Jeffrey D Ullman, “Compilers – Principles, Techniques and Tools”, 2nd Edition, Pearson Education, 2007.

**References:**

1. Randy Allen, Ken Kennedy, “Optimizing Compilers for Modern Architectures: A Dependence-Based Approach”, Morgan Kaufmann Publishers, 2002.
2. Steven S. Muchnick, “Advanced Compiler Design and Implementation”, Morgan Kaufmann Publishers – Elsevier Science, India, Indian Reprint 2003.
3. Keith D Cooper and Linda Torczon, “Engineering A Compiler”, Morgan Kaufmann Publishers Elsevier Science, 2004.
4. Charles N. Fischer, Richard. J. LeBlanc, “Crafting A Compiler With C”, Pearson  
   Education, 2008.

**Core 18: Optimization Techniques (PPCCA404)**

**Course Objectives:**

The primary objective of this course is to introduce the Mathematical techniques, and its applications for Optimization. Specifically

1. To introduce Operations Research methodologies.
2. To provide understanding on their applications in real life problems.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Understand basic principles and techniques of Operations Research.
2. Demonstrate their applications on real life problems.

**Course Prerequisites:**

This course requires basic knowledge of Mathematics.

**Detailed Syllabus**

**Module I**

Introduction to OR Concepts, Art of modeling, Computations in OR, Phases of OR study Linear Programming Problems(LPP) Concepts, Formulation of model, Graphical solution, Maximization / Minimization – Simplex Algorithm, Use of slack / surplus / artificial variables, Big M and Two phase method ; Dual problem – relation between primal and dual , Dual simplex method – Interpretation of dual variables, Revised Simplex Method, Introduction to Integer programming, Developing software for LP solution methods.

Transportation & Assignment problems

Concepts, formulations of models, Solution procedures, Optimality checks, Balanced/Unbalanced, Maximum/Minimum problems.

**Module II**

Network Analysis: Network Definition, Minimal spanning tree problem, shortest route problem, Maximal flow problem concepts and solution algorithm as applied to problems. Project planning and control by PERT/CPM network, Probability assessment in PERT network.

Queuing Models: Concepts relating to Queuing systems, types of queuing system (use of six-character code), Basic elements of Queuing Model, Role of Poison & Exponential Distribution, Concepts of Birth and Death process, Steady state measures of performance, M/M/1 model.

**Module III**

Computer Modeling& Simulation: Monte Carlo Simulation, Types of simulation, Elements of discrete event simulation, Random number generation, Design of simulation models.

Nonlinear programming algorithms: Direct search method, Quadratic Programming

**Text Books:**

1. Operations Research, An Introduction, by Hamdy A. Taha, 7th Edition-PHI

**References:**

1. Operations Research, Kanti Swaroop
2. Operation Research, V.K. Kapoor
3. Operation Research, Paneer Selvam, PHI
4. Operations Research, Hillier & Lieberman, TMH

**PE 1: Computer Graphics (PPECA401)**

**Course Objectives:**

1. To introduce the use of the components of a graphics system and become familiar with building approach of graphics system components and algorithms related with them.
2. To learn the basic principles of 3- dimensional computer graphics.
3. Provide an understanding of how to scan convert the basic geometrical primitives, how to transform the shapes to fit them as per the picture definition.
4. Provide an understanding of mapping from a world coordinates to device coordinates, clipping, and projections.
5. To be able to discuss the application of computer graphics concepts in the development of computer games, information visualization, and business applications

**Course Outcomes:**

On successful completion of the course, the student will be able:

1. To understand the basic concepts using computer graphics
2. To describe the importance of viewing and projections
3. To implement various algorithms to scan, convert the basic geometrical primitives, transformations, Area filling, clipping

**Module I**

An Introduction Graphics System: Computer Graphics and Its Types, Application of computer graphics, Graphics Systems: Video Display Devices, Raster Scan Systems, Random Scan Systems, Graphics Monitors and Work Stations, Input Devices, Hard Copy Devices, Graphics Software.

**Module II**

Output Primitives and Attributes of Output Primitives: Output Primitive Points and Lines, Line Drawing Algorithms, Circle Generating Algorithms, Scan-Line Polygon Fill Algorithm, Inside-Outside tests, Boundary-Fill Algorithm, Flood Fill Algorithm, Cell Array, Character Generation, Attributes of Output Primitives: Line Attributes, Color and Grayscale Levels, Area fill Attributes, Character Attributes, Bundled Attributes, Antialiasing. Two-dimensional Geometric Transformations: Basic Transformations, Matrix Representation and Homogeneous Coordinates, Composite Transformations, Reflection and Shearing.

**Module III**

Two-Dimension Viewing: The viewing Pipeline, Window to view port coordinate transformation, Clipping Operations, Point Clipping, Line Clipping, Polygon Clipping, Text Clipping, Exterior Clipping Three-Dimensional Concepts: Three Dimensional Display Methods, 3D Transformations, Parallel Projection and Perspective Projection. Three Dimensional Object Representations: Curved Surfaces, Quadratic Surfaces, Spline Representations, Bezier Spline Curves and Surfaces, B-Spline Curves and Surfaces, Octrees, BSP Trees, Fractal Geometry Methods, Shape Grammars. Visible Surface Detection Methods: Classification of Visible-Surface Detection Algorithms, Back-Face Detection, Depth-Buffer method, A-Buffer Method, Scan line and Depth Sorting, Area subdivision Method, Ray Casting Method.

**Text Books:**

1. Donald **Hearn** & M. Pauline **Baker**, “*Computer Graphics with OpenGL*”, Third Edition, 2004, Pearson Education, Inc. New Delhi.
2. Zhigang **Xiang**, Roy A. **Plastock**, “*Computer Graphics*”, Second Edition, 2007, McGraw-Hill Education (India), New Delhi.

**Reference Books:**

1. Plastock: Theory & Problem of Computer Gaphics, Schaum Series.
2. Foley & Van Dam: Fundamentals of Interactive Computer Graphics, Addison-Wesley.
3. Newman: Principles of Interactive Computer Graphics, McGraw Hill.

**PE 1: Data Warehousing and Mining (PPECA402)**

**Course Objectives:**

This course will introduce the concepts, techniques, design and applications of data warehousing and data mining.

1. To understand and implement classical algorithms in data mining and data warehousing.
2. To learn how to analyze the data, identify the problems, and choose the relevant algorithms to apply.
3. To be able to assess the strengths and weaknesses of the algorithms and analyze their behavior on real datasets.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Understand the functionality of the various data mining and data warehousing components.
2. Appreciate the strengths and limitations of various data mining and data warehousing models.
3. Compare the various approaches to data warehousing and data mining implementations.
4. Describe and utilize a range of techniques for designing data warehousing and data mining systems for real-world applications.

**Course Prerequisites:**

1. An introductory course on database systems.
2. Basic concepts in probability and statistics.

**Detail Syllabus**

**Module I**

Introduction to data warehousing, evolution of decision support systems, Modeling a data warehouse, granularity in the data warehouse, Data warehouse life cycle, building a data warehouse, Data Warehousing Components, Data Warehousing Architecture, On Line Analytical Processing, Categorization of OLAP Tools, Introduction to Data mining and knowledge discovery, Relation to Statistics, Databases, Data Mining Functionalities, Steps In Data Mining Process, Architecture of a Typical Data Mining Systems.

**Module II**

Classification of Data Mining Systems, Overview of Data Mining Techniques, Data Preprocessing, Data Cleaning, Data Integration, Data Transformation and Data Reduction, Data Generalization and Summarization Based Characterization, Mining Association Rules in Large Databases, Classification and Prediction, Issues Regarding Classification and Prediction, Classification by Decision Tree Induction, Bayesian Classification.

**Module III**

Other Classification Methods, Prediction, Clusters Analysis, Types of Data in Cluster Analysis, Categorization of Major Clustering Methods, Partitioning Methods, Hierarchical Methods, Applications of Data Mining, Social Impacts of Data Mining and Case Studies.

**Text Books**

1. Adriaans, P., Data mining. Addison- Wesley
2. Margaret Dunham, Data Mining: Introductory and Advanced Topics, Published by Prentice Hall
3. Sholom M. Weiss, Nitin Indurkhy, Predictive data mining: a practical guide, Morgan Kaufmann Publishers, 1998

**PE 1: Internet of Things (PPECA403)**

**Course Objectives:**

1. To comprehend fundamental concepts of IoT
2. To understand working of various IoT related devices
3. To learn principles of various IoT related protocols
4. To understand broad application areas of IoT

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Get a detailed understanding of IoT and related devices.
2. Analyze different IoT protocols in broad application areas.
3. Develop prototype applications of IoT

**Course Prerequisites:**

This course requires understanding of computer network and programming in any language.

**Detail Syllabus**

**Module I**

***Introduction to IoT:*** Definition, Evolution of IoT, IoT and related terms, Characteristics of IoT, Enabling Technologies, Applications of IoT.

***IoT Network Architecture:*** Standard Architectures, oneM2M, IoT World Forum (IoTWF) and Alternative IoT reference models, Simplified IoT Architecture and Core IoT Functional Stack, Fog, Edge and Cloud in IoT, Common Security Challenges in IoT.

***Smart Objects in IoT:*** Sensors, Actuators, classification of sensors, actuators, Smart Objects, Sensor Networks, Introduction to Arduino UNO, RaspberryPi, NodeMCU.

**Module II**

***Connecting Smart Objects:*** Communication Criteria, IoT Access Technologies, Bluetooth, Wifi, Zigbee, LoRaWAN, RFID, etc. IP and IoT: Constraint Nodes and Networks, Optimizing IP for IoT Network.

***Application Protocols for IoT:*** MQTT, XMPP, DDS, AMQP, CoAP, REST, and Comparison of protocols

**Module III**

***Applications of IoT:*** Connected Manufacturing, Architecture for the Connected Factory,  
Industrial Automation and Control Systems Reference Model; IoT and the Oil and Gas Industry, IoT Architectures for Oil and Gas; Smart and Connected Cities, Smart City Use-Case, Smart City IoT Architecture; Transportation Challenges,     IoT Use Cases for Transportation, IoT Architecture for Transportation; Challenges in Mining, IoT Strategy for Mining, Architecture for IoT in Mining; Overview of Public Safety, IoT Blueprint for Public Safety, Emergency Response IoT Architecture, School Bus Safety Network Architecture

**Text Books:**

1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things”, 1st Edition, Pearson Education (Cisco Press Indian Reprint), 2017.
2. Srinivasa K G, Siddesh G.M. Hanumantha Raju R. “Internet of Things”, CENGAGE Leaning India, 2017.

**Reference Books:**

1. Vijay Madisetti and Arshdeep Bahga, “Internet of Things: A Hands-on-Approach”, 1st Edition, University Press, 2015.
2. Raj Kamal, “Internet of Things: Architecture and Design Principles”, 1st Edition, McGraw Hill Education, 2017.

**Lab 13: Artificial Intelligence Laboratory (PLCCA401)**

***Note:*** This course shares the objectives and outcomes of its associated theory course PPCCA401. The exercises suggested below are illustrative in nature. Additional suitable exercises may be suggested by the faculty concerned to meet the course objectives. Prolog Programming will be used to carry out the exercises.

***List of Exercises:***

1. Loading, editing program files and getting familiarity with SWI-Prolog environment.
2. Developing a knowledge base of family relations, writing queries and executing queries.
3. Writing prolog program involving input and output.
4. Write menu-based interactive program.
5. Write program involving arithmetic
6. Write program involving structure
7. Writing program involving backtracking using fail predicate.
8. Write program using List manipulation
9. Write program using cut predicate
10. Write program using repeat predicate

**Lab 14: Enterprise Java Technologies Laboratory (PLCCA402)**

***Note:*** This course shares the objectives and outcomes of its associated theory course PPCCA402. Suitable IDE will be used to carry out laboratory exercises. The programs will follow proper object-oriented modeling. The exercises suggested below are illustrative in nature. Additional suitable exercises may be suggested by the faculty concerned to meet the course objectives.

***List of Exercises:***

1. Write a page counter Servlet to display how many times the webpage is accessed.
2. Write a Servlet program to display the date and time the webpage was last accessed by any user.
3. Write a Servlet program that displays the details of a student like Name, Roll No, Sex, Semester, Hobbies, Programming Languages known when user submits a form using suitable controls such as text field, combo box, list box, radio button, checkbox etc.
4. Write a Servlet program that validates user login. When a user submits user id and password using a login form, a Servlet retrieves the same and validates against the stored values. Store the values of user id and password as init parameters in web.xml file.
5. Write a program to develop a contact database with name, mobile number and email of a person. Use a HTML form to add a contact and a Servlet will retrieve the form data and store in the database table using JDBC.
6. Write a JSP program that will search the contact database (developed in Sl. No.5) using the person’s mobile number and display it.
7. Write a JSP program that will accept student details such as name, roll no, gender, and semester using HTML form. When the form is submitted it will display the details of the student just submitted. Use a Java Bean to hold the data.
8. Develop a web application following MVC model (combine Servlet, JSP and Java Bean) to develop an employee database. Perform operations such as insert employee records, search employee by Employee ID.
9. Using Java Server Faces (JSF) technology, develop an application to accept student details and display it. Use appropriate components such as text field, radio button, check box, combo box, list box to accept student name, gender, semester, languages known, and hobbies.
10. Develop a JSF application for login. Hardcode the userid and password values in the managed bean. If the login is success, it displays a success page and if the login fails it will display the login form with the error message. Define page navigation in the faces-config.xml for page navigation.
11. Develop a JSF application for simple arithmetic operations using two numbers. The webpage will have two input text fields to accept the numbers, one output text field to display the result and four command buttons to perform the operations. Use number converter as required by the application.
12. Develop a JSF application to check if a student scores a pass or fail in a subject. The webpage will have one input text field to accept the score, one output text field to display the result and one command button to submit data. Use range validator as required by the application to validate a student’s score between 0 and 100.
13. Develop JSF application using AJAX
14. Develop application using Sateless Session Bean
15. Develop application using Sateful Session Bean
16. Develop application using Message Driven Bean
17. Develop application involving Entity Bean and JPA
18. Develop application to perform CRUD operation using JPA
19. Develop application using Hibernate
20. Develop application using web service

**Lab 15: Compiler Design Lab (PLCCA403)**

Practice of LEX and YACC in windows/Linux OS. Practice of writing of programs either in C/C++/JAVA for implementation.

***List of Experiments:***

1. Design a lexical analyzer for given language and the lexical analyzer should ignore redundant spaces, tabs and new lines. It should also ignore comments. Although the syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value. Simulate the same in C/LEX language.
2. Write a program to identify whether a given line is a comment or not.
3. Write a program to recognize strings under 'a', 'a\*b+', 'abb'.
4. Write a program to test whether a given identifier is valid or not.
5. Write a program to simulate lexical analyzer for validating operators.
6. Implement the lexical analyzer using JLex, flex or other lexical analyzer generating Tools.
7. Write a program for implementing the functionalities of predictive parser for the mini Languages specified in **Note 1.**
8. Write a program for constructing of LL (1) parsing
9. Write a program for constructing recursive descent parsing.
10. Write a program to implement LALR parsing.
11. Write a program to implement operator precedence parsing
12. Write a program to implement Program semantic rules to calculate the expression that takes an expression with digits, + and \* and computes the value.
13. Convert the BNF rules into Yacc form and write code to generate abstract syntax tree for the mini language
14. Write a program to generate machine code from abstract syntax tree generated by the parser. The instruction set specified in **Note 2** may be considered as the target code.

**Note 1:**

A simple language written in this language is

{int a[3],t1,t2;

T1=2;

A[0]=1;a[1]=2;a[t]=3;

T2= -(a[2]+t1\*6)/(a[2]-t1);

If t2>5then

Print(t2)

Else{

Int t3;

T3=99;

T2=25;

Print(-t1+t2\*t3);/\*this is a comment on two lines\*/

}endif

}

Comments (zero or more characters enclosed between the standard C/JAVA Style comment brackets/\*…\*/)can be inserted .The language has rudimentary support for1- dimensional array, the declaration int a[3] declares an array of three elements, referenced as a[0],a[1] and a[2].

Note: You should worry about the scoping of names.

**Note 2:**

Consider the following mini language, a simple procedural high –level language, only operating on integer data, with syntax looking vaguely like a simple C crossed with Pascal. The syntax of the language is defined by the following grammar.

<program>::=<block>

<block>::={<variable definition><slist>}

|{<slist>}

<variabledefinition>::=int <vardeflist>

<vardec>::=<identifier>|<identifier>[<constant>]

<slist>::=<statement>|<statement>;<slist>

<statement>::=<assignment>|<ifstatement>|<whilestatement>

|<block>|<printstatement>|<empty>

<assignment>::=<identifier>=<expression>

|<identifier>[<expression>]=<expression>

<if statement>::=if<bexpression>then<slist>else<slist>endif

| if<bexpression>then<slist> endif

<whilestatement>::=while<bexpression>do<slist>enddo

<printstatement>::=print(<expression>)

<expression>::=<expression>::=<expression><addingop><term>|<term>|<addingop>

<term>

<bexpression>::=<expression><relop><expression>

<relop>::=<|<=|==|>=|>|!=

<addingop>::=+|-

<term>::=<term><multop><factor>|<factor>

<Multop>::=\*|/

<factor>::=<constant>|<identifier>|<identifier>[<expression>]

|(<expression>)

<constant>::=<digit>|<digit><constant>

<identifier>::=<identifier><letter or digit>|<letter>

<letter or digit>::=<letter>|<digit>

<letter>::=a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z

<digit>::=0|1|2|3|4|5|6|7|8|9

<empty>::=has the obvious meaning

**Lab 16 (PE Lab 1): Computer Graphics Lab (PLCCA404)**

***List of Experiments***

1. Program using OpenGL library functions, to implement the basic primitives such
2. as POINT, LINES, QUAD, TRIANGLES and POLYGON etc.
3. Program using OpenGL library functions, to implement the line chart as per user
4. Input. Input monthly data for period of one year.
5. Program to draw hard wired house by using basic primitives of OpenGL library
6. functions.
7. Program by using OpenGL library functions, to implement the Digital Differential
8. Analyzer line drawing algorithm.
9. Program by using OpenGL library functions, to implement the Bresenham’s Line
10. Drawing, Circle drawing, Mid-point Circle drawing algorithm.
11. Program by using OpenGL library functions, to implement the Cohen-Sutherland
12. Line clipping algorithm.
13. Program by using OpenGL library functions, to implement the Liang-Barsky Line
14. clipping algorithm.
15. Program to demonstrate 2D transformations.
16. Program to demonstrate 3D transformations.
17. Window to View Port Transformation

**Lab 16 (PE Lab 1): Data Warehousing and Mining Lab (PLCCA405)**

WEKA may be used to perform the exercises.

***List of Exercises:***

1. Execute multi-dimensional data model using SQL queries.
2. Implement various OLAP operations such as slice, dice, roll up, drill up, pivot etc.
3. Implementation of Text Mining on the data warehouse
4. Perform statistical analysis on a dataset
5. Perform dimensionality reduction of attributes on a dataset
6. Perform data cleaning on a dataset
7. Evaluate Information Gain of an attribute
8. Perform classification on data sets
9. Perform clustering on data sets
10. Perform Regression on data sets
11. Perform Association Rule on datasets

**Lab 16 (PE Lab 1): Internet of Things Lab (PLCCA406)**

***Note:*** The exercises suggested below are illustrative in nature. Appropriate exercises may be suggested by the faculty concerned to meet the course objectives.

**List of Exercises:**

1. Programming Arduino to control LEDs with a potentiometer
2. Programming Arduino to measure distance using ultrasonic sensor
3. Programming Arduino to record temperature in LCD Display using temperature sensor.
4. Programming Arduino to read and display RFID tag on LCD display
5. Programming Arduino to log temperature and pressure data onto SD card.
6. Programming Raspberry Pi to control a buzzer
7. Programming Raspberry Pi to control a servo motor
8. Programming Raspberry Pi to control a camera module
9. Programming Raspberry Pi to interface GPS module
10. Programming Raspberry Pi to measure distance using ultrasonic sensor.

**Semester-5**

**Core 19: Models and Patterns for Web Development (PPCCA501)**

**Course Objectives:**

1. To provide elaborate understanding of web development life cycle activities.
2. To provide introductory knowledge of Semantic Web
3. To provide understanding of selected design patterns used in web applications.

**Course Outcomes:**

On successful completion of the course, the students will be able to:

1. Understand web applications requirements and develop SRS.
2. Analyze requirements and develop various models for the web application.
3. Display ability to test software using appropriate techniques.
4. Understand usability, performance and security aspects of web applications.
5. Understand semantic web and technologies for developing semantic web.

**Course Prerequisites:**

1. Basic understanding of computer, programming and software engineering.

**Detailed Syllabus:**

**Module-I**

Web Engineering: Introduction, characteristics of web application; Requirement engineering for web applications; Modeling web applications: Modeling Specifics, Content Modeling, Hypertext or Navigation Modeling, Presentation Modeling, Customization Modeling, Modeling Methods and Tools; Web application architecture: Architecture Fundamentals, Components of a Generic Web Application Architecture, Layered Architectures, Data-aspect Architectures; Testing Web Applications: Testing Fundamentals, Test Approaches, Test schemes, methods and techniques, Test Automation;

**Module-II**

Operation and Maintenance of Web Applications: Challenges, Promoting a Web Application, Content Management, Usage Analysis; Usability of Web Applications: What is Usability? Design Guidelines, Web Usability Engineering Methods; Performance of Web Applications: What Is Performance, System Definition and Indicators, Analytical Techniques, Representing and Interpreting Results, Performance Optimization Methods; Security for Web Applications: Aspects of Security, Encryption, Digital Signatures and Certificates, Secure Client/Server-Interaction, Client Security Issues, Service Provider Security Issues;

The Semantic Web: Fundamentals of the Semantic Web, Technologies for semantic web, Ontology, Introduction to RDF, RDF Schema, OWL and SPARQL.

**Module-III**

Design Patterns: Introduction to design pattern, pattern categories, benefits of using patterns; Presentation Tier Patterns: Intercepting Filter, Front Controller, View Helper, Dispatcher view; Business Tier Patterns: Business Delegate, Session Façade, Business Object, Data Transfer Object; Integration Tier Patterns: Data Access Object, Web Service Broker; Messaging Patterns: Publish-Subscribe Pattern.

**Text Books:**

1. Web Engineering: The Discipline of Systematic Development of Web Applications by Gerti Kappel, Birgit Proll, Siegfried Reich, Werner Retschitzegger, John Wiley.
2. Core J2EE Patterns: Best Practices and Design Strategies, by Deepak Alur, John Crupi and Dan Malks, 2nd Edition, Prentice Hall 2003

**Reference Books:**

1. Web Engineering: A practitioner's approach by Roger Pressman and David Loue
2. Foundations of Semantic Web Technologies, by Pascal Hitzler, Markus Krotzsch, Sebastian Rudolph, CRC Press, 2009.
3. J2EE Design Patterns, by William Crawford and Jonathan Kaplan, O’Reilly, 2004

**PE 2: Multimedia and Animation (PPECA501)**

**Course Objectives**

* 1. To understand various file formats for audio, video and text media.
  2. To learn and understand technical aspect of Multimedia Systems.
  3. To strengthen their skills with key-frame animation, the graph editor, object hierarchies, forward and morph targets.

**Course Outcome**

* 1. To achieve a basic understanding of multimedia systems.
  2. To be able to evaluate more advanced or future multimedia systems.
  3. To be able to create a complete scene from photographic reference using all acquired 3D modeling techniques.
  4. To be able to create a seamless texture map to apply to a 3D surface.
  5. To be able to light a 3D object using three-point lighting.

**Module – I**

***Multimedia Fundamentals:*** Introduction, Multimedia & Hypermedia, WWW, Multimedia software tools, Multimedia Authoring and Tools: Multimedia Production, Multimedia Presentation, Graphics and Image Data Representation, Color Models in images & video, Fundamental Concepts in Video, Basics of digital Audio, Graphics and Image Editing.

**Module-II**

***Multimedia Data Compression:*** Lossless Compression Algorithms (Basics of Information Theory, Run length coding, variable length coding, Huffman Coding, lossless image compression, lossless JPEG), Lossy Compression Algorithms (distortion measure, quantization, Discrete Cosine transform), Basic Image Compression standard-JPEG, Main Steps in JPEG Image Compression, Basic Video Compression standard-MPEG (MPEG-1&2).

**Module-III**

***3D Animation Overview, Animation techniques; Exploring Animation:*** Using Principles of Fine Art and Traditional Animation, building a Good Story; Understanding Modeling and Texturing: Modeling, Texturing; Rigging and Animation: Rigging, Animation; Understanding Visual Effects, Lighting, and Rendering: Creating Visual Effects, Lighting, Lighting; Hardware and Software Tools of the Trade

**Text Books:**

* 1. Ze-Nian **Li** and Mark S. **Drew**, “*Fundamentals of Multimedia*”, First Edition, 2004, PHI Learning Pvt. Ltd., New Delhi.
  2. Andy Beane, “3D Animation Essentials”

**Reference Books:**

1. S. Gokul: Multimedia Magic, BPB Publication.
2. Bufford: Multimedia Systems, Addison Wesley
3. Richard Williams: “The Animator’s Survival Kit”, Faber & Faber

**PE 2: Machine Learning (PPECA502)**

**Course Objectives**:

Machine learning uses interdisciplinary techniques such as statistics, linear algebra, optimization, and computer science to create automated systems that can sift through large volumes of data at high speed to make predictions or decisions without human intervention. Machine learning as a field is now incredibly pervasive, with applications spanning from business intelligence to homeland security, from analyzing biochemical interactions to structural monitoring of aging bridges, and from emissions to astrophysics, etc. This class will familiarize students with a broad cross-section of models and algorithms for machine learning, and prepare students for research or industry application of machine learning techniques.

**Course Outcomes**: By the end of the course, students should be able to:

1. Develop an appreciation for what is involved in learning models from data.
2. Understand a wide variety of learning algorithms.
3. Understand how to evaluate models generated from data.
4. Apply the algorithms to a real-world problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models.

**Course Prerequisites**:

The student must have studied courses on Statistics and Computer Algorithms

**Detailed Syllabus:**

**Module-I**

Introduction to Machine Learning: Types of Learning, Unsupervised, Supervised, Reinforcement Learning

Linear Models for Regression: Linear Basis Function Models, Least Squares, Bayesian Linear Regression, Evidence Approximation.

Linear Models for Classification: Discriminant Functions, Fisher’s Discriminant, Probabilistic Discriminant Models, Fixed Basis Functions, Logistic Regression, Multiclass Logistic Regression

**Module-II**

Kernel Methods: Radial Basis Function, Gaussian Processes for Classification, SVM for Regression and Classification,

Mixture Models and EM: K-means Clustering, Mixtures of Gaussian, EM Algorithm.

Dimensionality Reduction: Principal Component Analysis, PCA for High Dimensional Data

**Module-III**

Neural Network: Introduction, Feed forward NN, Training Neural Network, Gradient Descent, Error Back Propagation, Regularization in Neural Network, Bayesian Neural Network.

Deep Learning: Introduction, Deep Feed Forward Network, Regularization for Deep Learning, Convolutional Network.

**Text Books:**

1. Christopher Bishop, Pattern Recognition and Machine Learning, Springer
2. Ian Good Fellow, Yoshua Dengio and Aaron Courville, Deep Learning, MIT Press

**Reference Books:**

1. Shai Shalev-Shwartz and Shai Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press.

**PE 2: Mobile Computing (PPECA503)**

**Course Objectives:**

The student should be made to:

1. Understand the basic concepts of mobile computing
2. Be familiar with the network protocol stack
3. Learn the basics of mobile telecommunication system
4. Be exposed to Ad-Hoc networks
5. Gain knowledge about different mobile platforms and application development

**Course Outcomes:**

At the end of the course, the student should be able to:

1. Explain the basics of mobile telecommunication system
2. Choose the required functionality at each layer for given application
3. Identify solution for each functionality at each layer
4. Use simulator tools and design Ad hoc networks
5. Develop a mobile application.

**Module- I**

Introduction: Applications, Wireless transmission: Frequencies for radio transmission, Regulations Signals Antennas Signal propagation: Path loss of radio signals, Additional signal propagation effects, Multi-path propagation

Multiplexing: Space division multiplexing, Frequency division multiplexing, Time division multiplexing, Code division multiplexing Modulation

Spread spectrum: Direct sequence spread spectrum, Frequency hopping spread spectrum

Medium access control: Motivation for a specialized MAC, SDMA, FDMA, TDMA, CDMA, Comparison of S/T/F/CDMA, GSM

**Module- II**

Satellite systems: Applications, Basics, Routing, Localization, Handover

Wireless LAN: Infrared vs radio transmission, Infrastructure and ad-hoc network, IEEE 802.11, HIPERLAN, Bluetooth

Mobile network layer: Mobile IP, Dynamic host configuration protocol, Mobile ad-hoc networks: Routing, Destination sequence distance vector, Dynamic source routing, Alternative metrics, Overview of ad-hoc routing protocols

**Module- III**

Mobile transport layer: Traditional TCP, Congestion control, Slow start, Classical TCP improvements,: Indirect TCP , Snooping TCP , Mobile TCP , Fast retransmit/fast recovery , Transmission/time-out freezing , Selective retransmission , Transaction-oriented TCP ,Wireless application protocol (version 1.x): Architecture , Wireless datagram protocol , Wireless transport layer security , Wireless transaction protocol ,Wireless session protocol , Wireless application environment , Wireless markup language , WML Script

**Text Book:**

1. MOBILE COMMUNICATIONS, Jochen Schiller ,2nd edition, Pearson Education
2. Prasant Kumar Pattnaik, Rajib Mall, “Fundamentals of Mobile Computing”, PHI Learning Pvt. Ltd, New Delhi – 2012

**Reference Books:**

1. Mobile Computing, Talukdar, Ahmed, Yavagal, 2nd edition, TMH Publication.
2. Jochen H. Schller, “Mobile Communications”, Second Edition, Pearson Education, New Delhi, 2007.
3. Dharma Prakash Agarval, Qing and An Zeng, “Introduction to Wireless and Mobile systems”, Thomson Asia Pvt Ltd, 2005.
4. Uwe Hansmann, Lothar Merk, Martin S. Nicklons and Thomas Stober, “Principles of Mobile Computing”, Springer, 2003.
5. William.C.Y.Lee,“Mobile Cellular Telecommunications-Analog and Digital Systems”, Second Edition,Tata Mc Graw Hill Edition ,2006.
6. C.K.Toh, “AdHoc Mobile Wireless Networks”, First Edition, Pearson Education, 2002.
7. Android Developers : <http://developer.android.com/index.html>
8. Apple Developer : <https://developer.apple.com/>
9. Windows Phone Dev Center: http://developer.windowsphone.com
10. BlackBerry Developer: http://developer.blackberry.com/

**PE 2: Software Project Management (PPECA504)**

**Course Objectives:**

1. To outline the need for Software Project Management
2. To highlight different techniques for software cost estimation and activity planning.

**Course Outcomes:**

* At the end of the course the students will be able to practice Project Management principles while developing a software.

**Module- I**

***Project Evaluation and Project Planning***

Importance of Software Project Management – Activities Methodologies – Categorization of Software Projects – Setting objectives – Management Principles – Management Control – Project portfolio Management – Cost-benefit evaluation technology – Risk evaluation – Strategic program Management – Stepwise Project Planning.

**Module- II**

***Project Life Cycle and Effort Estimation***

Software process and Process Models – Choice of Process models – mental delivery – Rapid Application development – Agile methods – Extreme Programming – SCRUM – Managing interactive processes – Basics of Software estimation – Effort and Cost estimation techniques – COSMIC Full function points – COCOMO II A Parametric Productivity Model – Staffing Pattern.

**Module- III**

***Activity Planning and Risk Management***

Objectives of Activity planning – Project schedules – Activities – Sequencing and scheduling – Network Planning models – Forward Pass & Backward Pass techniques – Critical path (CRM) method – Risk identification – Assessment – Monitoring – PERT technique – Monte Carlo simulation – Resource Allocation – Creation of critical patterns – Cost schedules.

**Text Book:**

1. Bob Hughes, Mike Cotterell and Rajib Mall: Software Project Management – Fifth Edition, Tata McGraw Hill, New Delhi, 2012.

**References:**

1. Robert K. Wysocki “Effective Software Project Management” – Wiley Publication,2011.
2. Walker Royce: “Software Project Management”- Addison-Wesley, 1998.
3. Gopalaswamy Ramesh, “Managing Global Software Projects” – McGraw Hill Education (India), Fourteenth Reprint 2013.

**PE 3: Virtual Reality (PPECA505)**

**Course objectives**

1. To make students know the basic concept and framework of virtual reality
2. To teach students the principles and multidisciplinary features of virtual reality.
3. To teach students the technology for multimodal user interaction and perception in VR, in particular the visual, audial and haptic interface and behavior.
4. To teach students the technology for managing large scale VR environment in real time.
5. To provide students with an introduction to the VR system framework and development tools.

**Course Outcomes:**

1. Ability to develop 3D virtual environments.
2. Ability to develop 3D interaction techniques.
3. Ability to develop immersive virtual reality applications.

**Module I**

Introduction of Virtual Reality: Fundamental Concept and Components of Virtual Reality. Primary Features and Present Development on Virtual Reality

Multiple Modals of Input and Output Interface in Virtual Reality: Input devices-- Tracker, Sensor, Digital Glove, gesture interfaces, Movement Capture, Video-based Input, 3D Menus & 3D Scanner etc. Output: Visual, Auditory, Phantom Haptic Devices, Haptic displays, Graphics displays.

Visual Computation in Virtual Reality: Fundamentals of Computer Graphics. Software and Hardware Technology on Stereoscopic Display. Computing Architectures for Virtual reality: rendering pipeline, Work station based Architectures.

**Module II**

Environment Modeling in VirtualReality: Geometric Modeling, Behavior Simulation, Physically Based Simulation, model management

Interactive Techniques in Virtual Reality: Body Track, Hand Gesture, 3D Manus, Object Grasp

**Module III**

Application of virtual reality: Medical applications, Military application, Robotics application

Introduction of Augmented Reality: System Structure of Augmented Reality. Key Technology in AR.

Development Tools and Frameworks in Virtual Reality: Frameworks of Software, Development Tools in virtual reality.

**Text Books**

1. Virtual Reality Technology, 2nd Edition Grigore C. Burdea, Philippe Coiffet, Wiley –IEEE press
2. Understanding Virtual reality, Interface, Application & design, William R. Sherman, Alan Craig, Elsevier

**Reference Books**

1. Virtual Reality Systems, Jhon Vince, Pearson Education
2. Virtual & Augmented Reality Paul Mealy, Learning Made Easy
3. Virtual Reality for Beginners: How to Understand, Use & Create with VR (Virtual Reality, Augmented Reality, Machine Learning) Murray Ramirez Kindle Edition,

**PE 3: Statistical Computing (PPECA506)**

**Course Objectives:**

1. To introduce students to state-of-the-art methods and modern programming tools for data analysis.
2. To learn the principles and methods of statistical analysis and also put them into practice using a range of real-world data sets.
3. To provide a basic understanding of data analysis using statistics and to use computational tools on problems of applied nature.
4. To investigate and evaluate relative efficiency of different methods

**Course Outcomes**: -After the successful completion of this course the students will be able to:

1. understand concepts of Large Numbers and different distributions in statistics and their limitations;
2. understand modern notions in data analysis-oriented computing;
3. be capable of confidently applying common Supervised & Unsupervised Learning algorithms in practice and implementing their own;
4. be capable of performing statistical computations;

**Detailed Syllabus**

**Module-I:**

Basic Probability Theory: Conditional probability, Independence; Random variables: Discrete and Continuous random variables, function of random variables; Expectation : Mean, variance, covariance; Random processes, convergence of random processes: The Law of Large Numbers, Central Limit Theorem, , t and F distributions Monte-Carlo Simulation;

**Module-II:**

Markov chains: Time-homogeneous discrete-time Markov chains

Linear methods for Regression and Classification: Overview of supervised Learning, Linear regression models and least squares, Multiple Regression, Subset selection, Ridge regression, least angle regression and Lasso, Linear discriminant analysis, Logistic regression.

Additive Models, Trees and Boosting: Generalized additive models, Regression and Classification trees, Boosting Methods- exponential loss and AdaBoost, Random forests and analysis.

**Module-III:**

Support Vector Machines (SVM), and K-nearest Neighbor: Basis expansion and regularization, Kernel smoothing methods, SVM for classification, Reproducing Kernels, SVM for regression, K-nearest Neighbor classifiers.

Unsupervised Learning: Cluster analysis, Principal Components, Gaussian mixtures and selection.

**Text Books**

1. Trevor Hastie, Robert Tibshirani , Jerome Friedman, The Elements of Statistical Learning-Data Mining, Inference and Prediction, Second Edition, Springer Verlag, 2009.
2. Carlos Fernandez-Granda,” Probability and Statistics for Data Science”, A note developed at Center for Data Science in New York University-2017.

**References**

1. 1.C.M. Bishop- Pattern Recognition and Machine Learning, Springer,2006.
2. L. Wasserman- All of statistics
3. John A. Rice, “Mathematical Statistics and Data Analysis” third edition, Cengage Learning.

**PE 3: Cloud Computing (PPECA507)**

**Course Objectives:**

1. To understand the concept of cloud and utility computing.
2. To understand the various issues in cloud computing.
3. To familiarize themselves with the lead players in cloud.
4. To appreciate the emergence of cloud as the next generation computing paradigm.
5. To be able to set up a private cloud.

**Course Outcomes:**

1. Upon completion of the course, the students will be able to:
2. Articulate the main concepts, key technologies, strengths and limitations of cloud computing.
3. Identify the architecture, infrastructure and delivery models of cloud computing.
4. Explain the core issues of cloud computing such as security, privacy and interoperability.
5. Choose the appropriate technologies, algorithms and approaches for the related issues.

**Module I**

***Evolution of Computing Paradigms*** - Overview of Existing Hosting Platforms, Grid Computing, Utility Computing, Autonomic Computing, Dynamic Datacenter Alliance, Hosting / Outsourcing, Introduction to Cloud Computing, Workload Patterns for the Cloud, “Big Data”, IT as a Service, Technology Behind Cloud Computing,

**Module II**

***A Classification of Cloud Implementations***- Amazon Web Services - IaaS, The Elastic Compute Cloud (EC2), The Simple Storage Service (S3), The Simple Queuing Services (SQS), VMware vCloud - IaaS, vCloud Express, Google AppEngine - PaaS, The Java Runtime Environment,

**Module III**

***The Python Runtime Environment***- The Datastore, Development Workflow, Windows Azure Platform - PaaS, Windows Azure, SQL Azure, Windows Azure AppFabric, Salesforce.com - SaaS / PaaS, Force.com, Force Database - the persistency layer, Data Security, Microsoft Office Live - SaaS, LiveMesh.com, Google Apps - SaaS, A Comparison of Cloud Computing Platforms, Common Building Blocks. Infrastructure security – Data security – Identity and access.

**Text Book:**

* 1. Kai Hwang, Geoffrey C. Fox and Jack J. Dongarra, “Distributed and Cloud Computing from Parallel Processing to the Internet of Things”, Morgan Kaufmann, Elsevier, 2012

**Reference Books**

* 1. Barrie Sosinsky, “Cloud Computing Bible” John Wiley & Sons, 2010
  2. Tim Mather, Subra Kumaraswamy, and Shahed Latif, “Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance”, O'Reilly 2000

**PE 3: Software Testing (PPECA508)**

**Course Objectives:**

1. To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
2. To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
3. To learn how to planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.
4. To expose the advanced software testing topics, such as object-oriented software testing methods, and component-based software testing issues, challenges, and solutions.
5. To gain software testing experience by applying software testing knowledge and methods to practice-oriented software testing projects.
6. To understand software test automation problems and solutions.
7. To learn how to write software testing documents, and communicate with engineers in various forms.
8. To gain the techniques and skills on how to use modern software testing tools to support software testing projects.

**Course Outcomes**:

By the end of the course, the student should:

1. Have an ability to apply software testing knowledge and engineering methods.
2. Have an ability to design and conduct a software test process for a software testing project.
3. Have an ability to identify the needs of software test automation, and define and develop a test tool to support test automation.
4. Have an ability understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.
5. Have an ability to use various communication methods and skills to communicate with their teammates to conduct their practice-oriented software testing projects.
6. Have basic understanding and knowledge of contemporary issues in software testing, such as component-based software testing problems
7. Have an ability to use software testing methods and modern software testing tools for their testing projects.

**Detailed Syllabus:**

**Module I**

Software Testing: Introduction, Evolution, Myths & Facts, Goals, Psychology, Definition, Model for testing, Effective Vs Exhaustive Software Testing. Software Testing Terminology and Methodology: Software Testing Terminology, Software Testing Life Cycle, relating test life cycle to development life cycle Software Testing Methodology.

Verification and Validation: Verification & Validation Activities, Verification, Verification of Requirements, High level and low level designs, How to verify code, Validation  
Dynamic Testing I: Black Box testing techniques: Boundary Value Analysis, Equivalence class Testing, State Table based testing, Decision table based testing, Cause-Effect Graphing based testing, Error guessing

**Module II**

Dynamic Testing II: White-Box Testing: need, Logic coverage criteria, Basis path testing, Graph matrices, Loop testing, data flow testing, mutation testing Static Testing: inspections, Structured Walk through, Technical reviews.

Validation activities: Unit testing, Integration Testing, Function testing, system testing, acceptance testing Regression testing: Progressives Vs regressive testing, Regression test ability, Objectives of regression testing, When regression testing done? Regression testing types, Regression testing techniques

**Module III**

Efficient Test Suite Management: Test case design Why does a test suite grow, Minimizing the test suite and its benefits, test suite prioritization, Types of test case prioritization, prioritization techniques, measuring the effectiveness of a prioritized test suite  
Software Quality Management: Software Quality metrics, SQA models Debugging: process, techniques, correcting bugs, Basics of testing management tools, test link and Jira

Automation and Testing Tools: need for automation, categorization of testing tools, selection of testing tools, Cost incurred, Guidelines for automated testing, overview of some commercial testing tools. Testing Object Oriented Software: basics, Object oriented testing  
Testing Web based Systems: Challenges in testing for web based software, quality aspects, web engineering, testing of web based systems, Testing mobile systems

**Text Books**

1. Software Testing, Principles and Practices, Naresh Chauhan, Oxford
2. Foundations of Software testing, Aditya P Mathur, 2ed, Pearson
3. Software Testing- Yogesh Singh, CAMBRIDGE

**Reference books**

1. Software testing techniques – Baris Beizer, International Thomson computer press, second edition.
2. Software Testing, Principles, techniques and Tools, M G Limaye, TMH
3. Effective Methods for Software testing, Willian E Perry, 3ed, Wiley

**OE 4: Digital Image Processing (POECA501)**

**Course Objectives:** The Student Should Be Made To:

1. Learn Digital Image Fundamentals.
2. Be Exposed to Simple Image Processing Techniques.
3. Be Familiar with Image Compression And Segmentation Techniques.
4. Learn to Represent Image In Form Of Features.

**Course Outcomes:**

Upon completion of this course, the students should be able to

1. Apply suitable image processing techniques for various applications.
2. Integrate various suitable image processing techniques for complex problems.

**Detailed Syllabus**

**Module I**

Introduction – Origin – Steps In Digital Image Processing – Components – Elements Of Visual Perception – Image Sensing And Acquisition – Image Sampling And Quantization – Relationships Between Pixels – Color Models.

***Image Enhancement:*** Spatial Domain: Gray Level Transformations – Histogram Processing – Basics Of Spatial Filtering–Smoothing And Sharpening Spatial Filtering – Frequency Domain: Introduction To Fourier Transform – Smoothing And Sharpening Frequency Domain Filters – Ideal, Butterworth And Gaussian Filters.

**Module II**

***Image Restoration and Segmentation*** Noise Models – Mean Filters – Order Statistics – Adaptive Filters – Band Reject Filters – Band Pass Filters – Notch Filters – Optimum Notch Filtering – Inverse Filtering – Wiener Filtering Segmentation: Detection Of Discontinuities–Edge Linking And Boundary Detection – Region Based Segmentation- Morphological Processing- Erosion And Dilation.

**Module III**

***Wavelets and Image Compression***

Wavelets – Subband Coding – Multiresolution Expansions – Compression: Fundamentals – Image Compression Models – Error Free Compression – Variable Length Coding – Bit-Plane Coding – Lossless Predictive Coding – Lossy Compression – Lossy Predictive Coding – Compression Standards.

***Image Representation and Recognition*** Boundary Representation – Chain Code – Polygonal Approximation, Signature, Boundary Segments – Boundary Description – Shape Number – Fourier Descriptor, Moments- Regional Descriptors –Topological Feature, Texture – Patterns And Pattern Classes – Recognition Based On Matching.

**Text Books:**

1. R.C. Gonzalez, R.E. Woods, Digital Image Processing, 3rd Edition, Pearson Education
2. R C Gonzalez, Woods and Eddins, Digital Image Processing using Matlab, 2nd Edition, Tata McGraw Hill

**Reference Books:**

1. S. Sridhar, Digital Image Processing, Oxford University Press, 2011
2. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, “Digital Image Processing Using MATLAB”, Third Edition Tata Mc Graw Hill Pvt. Ltd., 2011.
3. Anil Jain K. “Fundamentals Of Digital Image Processing”, PHI Learning Pvt. Ltd., 2011.
4. William K Pratt, “Digital Image Processing”, John Willey, 2002.
5. Malay K. Pakhira, “Digital Image Processing And Pattern Recognition”, First Edition, PHI Learning Pvt. Ltd., 2011.

**OE 4: Big Data Analytics** (**POECA502)**

**Course Objectives:**

1. To know the fundamental concepts of big data and analytics.
2. To explore tools and practices for working with big data
3. To learn about stream computing.
4. To know about the research that requires the integration of large amounts of data.

**Course Outcomes:**

Students will be able to:

1. Apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer based systems of varying complexity.
2. Critically analyze a problem, identify, formulate and solve problems in the field of Computer Science and Engineering considering current and future trends.
3. Analyze the local and global impact of computing on individuals, organizations, and society.
4. Use current techniques, skills, and tools necessary for computing practice.
5. Demonstrate advanced knowledge of a selected area within the computer science discipline.
6. Critically analyze existing systems in an area of specialization and develop innovative solutions.

**Course Prerequisites:** Familiarity with Intermediate Python or Java is advised. Most assignments could easily be done in Python, Scala, Java or R. We will assume familiarity with Linux commands.

**Detailed Syllabus**

**Module-I**

***Overview of Big Data:*** History of Data Management-Evolution of Big Data, Elements of Big Data: volume, velocity, variety, veracity, Big data Analytics, Advantages of Big data Analytics.

***Use of Big Data:*** Social Networking, Business Intelligence, Marketing, Product Design and Development, Preventing Fraud, Use of RFID in Retail.

***Handling Big Data:*** Introduction to Hadoop, How does Hadoop Function? Hadoop Ecosystem, Hadoop Distributed File system, HDFS Architecture, Features of HDFS, MapReduce, Features of MapReduce, Hadoop YARN, HBase, Hive, Pig and Pig Latin, ZooKeeper.

***The MapReduce Framework:*** Exploring the features of MapReduce, Working of MapReduce, Exploring Map and Reduce Functions, Optimize MapReduce Jobs, Uses of MapReduce, Role of HBase in Big Data Processing, Characteristics of HBase.

**Module-II**

***Big Data Technology Foundations:*** Exploring the Big Data stack, Data Sources Layer, Ingestion Layer, Storage Layer, Physical Infrastructure Layer, Platform Management Layer, Security Layer, Monitoring layer, Analytics Engine, Visualization Layer.

***RDBMS and Big Data:*** Issues with Relational Model, Non–Relational Data Base, Issues with the Non- Relational Model, Polyglot Per Persistence, Integrating Big data with Traditional Data Warehouses, Big data Analysis and Data Warehouse, Changing Deployment Models in Big Data Era.

***Storing Data in Hadoop:*** Introducing HDFS, HDFS Architecture, using hdfs files, hadoop Specific File System Types, HDFS Commands, Introducing HBase: HBase Architecture, Storing Big data with HBase, interacting with Hadoop Ecosystem, Base in Operation-Programming with HBase.

**Module-III**

***YARN:*** Background of YARN, Advantages of YARN, YARN Architecture, YARN Schedulers.

***Hive:*** Getting started with Hive: Hive Variables, Hive Properties, Hive Queries, Data types in Hive, Hive DDL, Creating databases, Viewing a database, Dropping a Data Base, Altering Databases, Creating Tables, Data Manipulations in Hive, Data Retrieval queries, Using JOINS in Hives.

NoSQL Data management, Introduction to NoSQL,Characteristics of NoSQL,Evoluation of Databases, Aggregate Data Models, Key Value Data Models, Document Databases, Relationships, Graph Data Base, Schema-Less Data Bases, Materialized Views

***Big Data Analytics:*** Comparing Reporting and analysis, Reporting, analysis, The Analytic Process, Types of analysis, Basic Analytics, Advanced Analytics, Operationalized Analytics, Monetized analytics, Characteristics of Big Data Analysis.

***Data Visualization:*** Overview, Techniques, Types, Applications, Big Data Visualization, Data Visualization Tools.

***Social Media Analytics and Text Mining:*** Introducing Key Elements of Social Media, Introducing Text Mining, Understanding Text Mining Processing, Sentiment Analysis,

***Introducing Mobile Analytics:*** Mobile Analytics and Web Analytics, Types of Results from Mobile Analytics, Types of Applications for Mobile Analytics, Introducing Mobile Analytics tools, Location- Based Tracking Tools, Real-Time Analytics Tools, User Behaviour Tracking Tools.

**Text Books:**

1. DT Editorial Services, BIG DATA, Black Book, Dreamtech Press
2. Anand Rajaraman and Jeffrey David Ullman, "Mining of Massive Datasets", Cambridge University Press, 2012.
3. David Loshin, "Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph", 2013.

**References:**

1. EMC Education Services, "Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data", Wiley publishers, 2015.
2. Bart Baesens, "Analytics in a Big Data World: The Essential Guide to Data Science and its Applications", Wiley Publishers, 2015.
3. Dietmar Jannach and Markus Zanker, "Recommender Systems: An Introduction", Cambridge University Press, 2010.
4. Kim H. Pries and Robert Dunnigan, "Big Data Analytics: A Practical Guide for Managers " CRC Press, 2015.
5. Jimmy Lin and Chris Dyer, "Data-Intensive Text Processing with MapReduce", Synthesis Lectures on Human Language Technologies, Vol. 3, No. 1, Pages 1-177, Morgan Claypool publishers, 2010

**OE 4: Natural Language Processing (POECA503)**

**Course Objectives:**

1. To learn the fundamentals of natural language processing
2. To understand the use of CFG and PCFG in NLP
3. To understand the role of semantics of sentences and pragmatics
4. To apply the NLP techniques to IR applications

**Course Outcomes:**

Upon completion of the course, the students will be able to:

1. To tag a given text with basic Language features
2. To design an innovative application using NLP components
3. To implement a rule-based system to tackle morphology/syntax of a language
4. To design a tag set to be used for statistical processing for real-time applications
5. To compare and contrast the use of different statistical approaches for different types of NLP applications.

**Module I**

***Introduction***

Origins and challenges of NLP – Language Modeling: Grammar-based LM, Statistical LM – Regular Expressions, Finite-State Automata – English Morphology, Transducers for lexicon and rules, Tokenization, Detecting and Correcting Spelling Errors, Minimum Edit Distance

**Module II**

***Word Level Analysis***

Unsmoothed N-grams, Evaluating N-grams, Smoothing, Interpolation and Backoff – Word Classes, Part-of-Speech Tagging, Rule-based, Stochastic and Transformation-based tagging, Issues in PoS tagging – Hidden Markov and Maximum Entropy models. Context-Free Grammars, Grammar rules for English, Treebanks, Normal Forms for grammar.

***Syntactic Analysis***

Dependency Grammar – Syntactic Parsing, Ambiguity, Dynamic Programming parsing – Shallow parsing – Probabilistic CFG, Probabilistic CYK, Probabilistic Lexicalized CFGs – Feature structures, Unification of feature structures. Requirements for representation, First-Order Logic, Description Logics.

**Module III**

***Semantics and Pragmatics***

Syntax-Driven Semantic analysis, Semantic attachments – Word Senses, Relations between Senses, Thematic Roles, selectional restrictions – Word Sense Disambiguation, WSD using Supervised, Dictionary & Thesaurus, Bootstrapping methods – Word Similarity using Thesaurus and Distributional methods.

***Discourse Analysis and Lexical Resources***

Discourse segmentation, Coherence – Reference Phenomena, Anaphora Resolution using Hobbs and Centering Algorithm – Coreference Resolution – Resources: Porter Stemmer, Lemmatizer, Penn Treebank, Brill’s Tagger, WordNet, PropBank, FrameNet, Brown Corpus, British National Corpus (BNC).

**Text Books:**

1. Daniel Jurafsky, James H. Martin―Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, Pearson Publication, 2014.
2. Steven Bird, Ewan Klein and Edward Loper, ―Natural Language Processing with Python, First Edition, OReilly Media, 2009.

**References:**

1. Breck Baldwin, ―Language Processing with Java and LingPipe Cookbook, Atlantic Publisher, 2015.
2. Richard M Reese, ―Natural Language Processing with Java, OReilly Media, 2015.
3. Nitin Indurkhya and Fred J. Damerau, ―Handbook of Natural Language Processing, Second Edition, Chapman and Hall/CRC Press, 2010.
4. Tanveer Siddiqui, U.S. Tiwary, ―Natural Language Processing and Information Retrieval, Oxford University Press, 2008.

**OE 4: Wireless Sensor Networks (POECA504)**

**Course Objectives:**

1. This course deals with the comprehensive knowledge about wireless sensor networks.
2. It provides an insight into different layers and their design considerations.
3. A thorough knowledge of infrastructure establishment and sensor network platform is provided.
4. To provide an overview about sensor networks and emerging technologies.

**Course Outcomes:** Students will be able to

1. Analyze modeling and simulation of various communication networks
2. Generate test and estimate parameters.
3. Apply this knowledge for detection estimation and simulation of various communication networks.

**Detailed Syllabus**

**Module I**

***MAC & Routing In Ad Hoc Networks:*** Introduction – Issues and challenges in ad hoc networks – MAC Layer Protocols for wireless ad hoc networks – Contention-Based MAC protocols – MAC Protocols Using Directional Antennas – Multiple-Channel MAC Protocols – Power-Aware MAC Protocols – Routing in Ad hoc Networks – Design Issues – Proactive, Reactive and Hybrid Routing Protocols

**Module II**

***Transport & QoS In Ad Hoc Networks***

TCP challenges and Design Issues in Ad Hoc Networks – Transport protocols for ad hoc networks – Issues and Challenges in providing QoS – MAC Layer QoS solutions – Network Layer QoS solutions – QoS Model; Introduction – Applications – Challenges – Sensor network architecture – MAC Protocols for wireless sensor networks – Low duty cycle protocols and wakeup concepts.

**Module III**

***MAC & Routing in Wireless Sensor Networks***

Contention-Based protocols – Schedule-Based protocols – IEEE 802.15.4 Zigbee – Topology Control – Routing Protocols; Data-Centric and Contention-Based Networking – Transport Layer and QoS in Wireless Sensor Networks – Congestion Control in network processing – Operating systems for wireless sensor networks – Examples

***Transport & QoS in Wireless Sensor Networks***

Security Attacks – Key Distribution and Management – Intrusion Detection – Software based Anti-tamper techniques – Water marking techniques – Defense against routing attacks – Secure Ad hoc routing protocols – Broadcast authentication WSN protocols – TESLA – Biba – Sensor Network Security Protocols – SPINS

**Text Books:**

1. C. Siva Ram Murthy and B.S. Manoj, ―Ad Hoc Wireless Networks – Architectures and Protocols, Pearson Education, 2006.
2. Holger Karl, Andreas Willing, ―Protocols and Architectures for Wireless Sensor Networks, John Wiley & Sons, Inc., 2005.

**References**

1. Subir Kumar Sarkar, T G Basavaraju, C Puttamadappa, ―Ad Hoc Mobile Wireless Networks, Auerbach Publications, 2008.
2. Carlos De Morais Cordeiro, Dharma Prakash Agrawal, ―Ad Hoc and Sensor Networks: Theory and Applications (2nd Edition), World Scientific Publishing, 2011.
3. Waltenegus Dargie, Christian Poellabauer, ―Fundamentals of Wireless Sensor Networks Theory and Practice, John Wiley and Sons, 2010
4. Xiang-Yang Li, “Wireless Ad Hoc and Sensor Networks: Theory and Applications, 1227 th edition, Cambridge university Press,2008

**OE 4: Parallel & Distributed Systems (POECA505)**

**Prerequisite:** Systems Programming OR Operating system

**Course Objective:**

The objective of this course is:

1. To introduce the fundamentals of parallel and distributed processing
2. Introduction to system architecture, programming model, and performance analysis
3. The specific topics include multithreaded programming, message passing interface

**Course Outcomes:**

On successful completion of this course students will be able to:

1. Develop and apply knowledge of parallel and distributed computing techniques and methodologies.
2. Apply design, development, and performance analysis of parallel and distributed applications.
3. Use the application of fundamental Computer Science methods and algorithms in the development of parallel applications.
4. Explain the design, testing, and performance analysis of a software system, and to be able to communicate that design to others.

**Detailed Syllabus**

**Module-I**

Introduction to Parallel Computing; Motivating Parallelism, Scope of Parallel Computing; Parallel

Programming; Platforms: Implicit parallelism, Limitation of Memory System Performance, Dichotomy of Parallel Computing Platforms, Physical Organization of Parallel Platforms: PRAM Model, Interconnection network, network topology, Evaluation of interconnection network.

**Module-II**

Communication Costs of Parallel Machines, Routing Mechanism for Interconnection Networks, Impact of Process- processor Mapping and Mapping Techniques.

Principles of Parallel Algorithm Design: Preliminaries, Decomposition Techniques, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing, Methods for containing interaction Overheads.

**Module-III**

Parallel Algorithm Models: Basic Communication Operations: One-to-All Broadcast and All-to-One Reduction, All-to-All Broadcast and Reduction, Scatter and Gather, All-to-All Personalized Communication, Circular Shift. All reduce and prefix sum.

Analytical Modeling of Parallel Programs: Sources of Overhead, Performance metrics. Effect of Granularity on Performance, Scalability of Parallel Systems, minimum Execution time and minimum cost-optimal Execution time, Asymptotic Analysis of Parallel Programs, Introduction to MPI Principles of Message: The Building Blocks (Send and Receive Operations), Message Passing Interface, Collective Communication and Computation Operations.

**Text Books:**

1. Ananth Grama, George Karypis, Vipin Kumar, Anshul Gupta, “Introduction to Parallel Computing”, 2nd Edition, 2004, Pearson Education, Inc. New Delhi.

**Reference Books:**

1. Michael J. Quinn, “Parallel Computing: Theory and Practice”, 1994, McGraw-Hill Education (India), New Delhi.
2. Calvin Lin, Larry Snyder, “Principles of Parallel Programming”, 1st Edition, 2009, Pearson Education, Inc. New Delhi.
3. Michael J. Quinn, “Parallel Programming in C with MPI and OpenMP”, 2004, McGraw-Hill Education (India), New Delhi.
4. Barry Wilkinson, “Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers”, 2nd Edition, 2005, Pearson Education, Inc. New Delhi.
5. Yves Robert, Henri Casanova, Armand Legrand, “Parallel Algorithms”, 1st Edition, 2009, CRC Press. ISBN-13:9781584889458.

**OE 4: Soft Computing (POECA506)**

**Course Objectives:**

1. To learn the basic concepts of Soft Computing
2. To become familiar with various techniques like neural networks, genetic algorithms and fuzzy systems.
3. To apply soft computing techniques to solve problems.

**Course Outcomes:**

Upon completion of this course, the students should be able to

1. Apply suitable soft computing techniques for various applications.
2. Integrate various soft computing techniques for complex problems.

**Module - I**

Introduction to Soft Computing, Artificial Neural Network(ANN) : Fundamentals of ANN, Basic Models of an artificial Neuron, Neural Network Architecture, Learning methods, Terminologies of ANN, Hebb network, Supervised Learning Networks: Perceptron, MLP, Architecture of a Back propagation Network : back propagation, Learning Effect of Tunning parameters of the Back propagation, Adaline, Madaline, RBF Network, Associative memory: Auto, hetero and linear associative memory, network, Adaptive Resonance Theory ART1, ART2, Applications

**Module –II**

***FUZZY LOGIC*** Fuzzy set theory: crisp sets, fuzzy sets, crisp relations, fuzzy relations, Fuzzy Systems: Crisp logic predicate logic, fuzzy logic, fuzzy Rule based system, Defuzzification Methods, Fuzzy rule-based reasoning

**Module –III**

***GENETIC ALGORITHMS*** Fundamentals of genetic algorithms: Encoding, Fitness functions, Reproduction. Genetic Modeling: Cross cover, Inversion and deletion, Mutation operator, Bit-wise operators, Bitwise operators used in GA. Convergence of Genetic algorithm. Applications, Real life Problems

Hybrid Soft Computing Techniques Hybrid system, neural Networks, fuzzy logic and Genetic algorithms hybrids. Genetic Algorithm based Back propagation Networks: GA based weight determination applications: Fuzzy logic controlled genetic Algorithms soft computing tools, Applications.

**Text Book:**

1. Principles of Soft Computing- S.N.Sivanandan and S.N.Deepa, Wiley India, 2nd Edition,2011

**Reference Book:**

1. Neuro Fuzzy and Soft Computing, J. S. R. Jang, C.T. Sun, E. Mitzutani, PHI
2. Neural Networks, Fuzzy Logic, and Genetic Algorithm (synthesis and Application) S. Rajasekaran, G.A. Vijayalakshmi Pai, PHI

**MC 6: Professional Ethics (PMCMH501)**

**Course Objectives:**

To enable the students to create an awareness on Engineering Ethics and Human Values, to instill Moral and Social Values and Loyalty and to appreciate the rights of others.

**Course Outcomes:**

Upon completion of the course, the student should be able to apply ethics in society, discuss the ethical issues related to engineering and realize the responsibilities and rights in the society

**Module I**

***Human Values***

Morals, values and Ethics – Integrity – Work ethic – Service learning – Civic virtue – Respect for others – Living peacefully – Caring – Sharing – Honesty – Courage – Valuing time – Cooperation – Commitment – Empathy – Self confidence – Character – Spirituality – Introduction to Yoga and meditation for professional excellence and stress management.

**Module II**

***Engineering Ethics***

Senses of ‘Engineering Ethics’ – Variety of moral issues – Types of inquiry – Moral dilemmas – Moral Autonomy – Kohlberg’s theory – Gilligan’s theory – Consensus and Controversy – Models of professional roles – Theories about right action – Self-interest – Customs and Religion – Uses of Ethical Theories

**Module III**

***Engineering as Social Experimentation***

Engineering as Experimentation – Engineers as responsible Experimenters – Codes of Ethics – A Balanced Outlook on Law.

**Text Books:**

1. Mike W. Martin and Roland Schinzinger, “Ethics in Engineering”, Tata McGraw Hill, New Delhi, 2003.
2. Govindarajan M, Natarajan S, Senthil Kumar V. S, “Engineering Ethics”, Prentice Hall of India, New Delhi, 2004.

**References:**

1. Charles B. Fleddermann, “Engineering Ethics”, Pearson Prentice Hall, New Jersey, 2004.
2. Charles E. Harris, Michael S. Pritchard and Michael J. Rabins, “Engineering Ethics – Concepts and Cases”, Cengage Learning, 2009
3. John R Boatright, “Ethics and the Conduct of Business”, Pearson Education, New Delhi, 2003
4. Edmund G Seebauer and Robert L Barry, “Fundametals of Ethics for Scientists and Engineers”, Oxford University Press, Oxford, 2001
5. Laura P. Hartman and Joe Desjardins, “Business Ethics: Decision Making for Personal Integrity and Social Responsibility” Mc Graw Hill education, India Pvt. Ltd.,New Delhi 2013.\
6. World Community Service Centre, ‘Value Education’, Vethathiri publications, Erode, 2011

**Project 2: Minor Project-II (PPRCA506)**

**[To be decided by the department]**

**Project 3: Summer Internship Evaluation (PPRCA507)**

**[To be decided by the department]**

**Lab 17: Models and Patterns for Web Development Lab (PLCCA501)**

This course shares the objectives and outcomes of its associated theory course PPCCA501.

Suitable tools will be used to carry out laboratory exercises.

***List of Exercises:***

1. Analyze the requirements and develop SRS for a web application. The students are to perform in groups and each group will take one application and present the SRS.
2. Identify information and functional requirements, and develop a content model.
3. Identify various functions that the user will perform with the system. Develop Use-Case Model for the system.
4. Develop specification for each use-case.
5. Develop a navigation model for the system
6. Develop sequence diagram for each usecase following the specification.
7. Develop presentation model for the web application.
8. Develop n-tier architectural model for system.
9. Use a suitable tool to perform different testing such as link, load, stress and compatibility for web application.
10. Identify and use suitable patterns for the web application

**Lab 18: Elective-II Lab (PLCCA502)**

At least 10 experiments based on real life applications are to be conducted by the concerned teacher.

**Lab 19: Elective-III Lab (PLCCA503)**

At least 10 experiments based on real life applications are to be conducted by the concerned teacher.

**Semester-6**

**Project 4: Industrial Training cum Project (PPRCA601)**

**[To be decided by the department]**