# COURSES SCHEME & SYLLABUS

## M.E. (SOFTWARE ENGINEERING)

**ME- SOFTWARE ENGINEERING (2016-18)**

| SEMESTER-I | | | | | | |
|---|---|---|---|---|---|---|
| **S.No.** | **Code** | **Title** | **L** | **T** | **P** | **Cr** |
| 1 | PCL105 | Statistical Methods and Algorithms | 3 | 0 | 2 | 4 |
| 2 | PCS104 | Advance Data Structure and Algorithms | 3 | 0 | 4 | 5 |
| 3 | PSE101 | Software Engineering Concepts and Methodologies | 3 | 0 | 2 | 4 |
| 4 | PSE105 | Software Design and Construction | 3 | 0 | 2 | 4 |
| 5 | PSE104 | Software Project Management | 3 | 0 | 2 | 4 |
| 6 | PCS107 | Technology and Innovation | 1 | 0 | 1 | 2 |
| | | **Total** | **15** | **0** | **12** | **23** |
| **SEMESTER-II** | | | | | | |
| **S.No.** | **Code** | **Title** | **L** | **T** | **P** | **Cr** |
| 1 | PSE202 | Software Verification And Validation Testing | 3 | 0 | 2 | 4 |
| 2 | PSE205 | Software Metrics And Quality Management | 3 | 0 | 2 | 4 |
| 3 | PSE214 | Engineering Design Project (4 Self Effort Hrs) | 1 | 0 | 4 | 5 |
| 4 | | Elective-I | 3 | 0 | 2 | 4 |
| 5 | | Elective-II | 3 | 0 | 2 | 4 |
| 6 | | Elective-III | 3 | 0 | 2 | 4 |
| | | **Total** | **15** | **0** | **14** | **25** |
| **ELECTIVE I** | | | | | | |
| **S.No.** | **Code** | **Title** | **L** | **T** | **P** | **Cr** |
| | PSE204 | Advanced Topics In Software Engineering | 3 | 0 | 2 | 4 |
| | PSE206 | Agile Software Development Approaches | 3 | 0 | 2 | 4 |
| | PSE207 | Component Based Development | 3 | 0 | 2 | 4 |
| | PCS224 | Natural Language Processing | 3 | 0 | 2 | 4 |
| | | | | | | |
| **ELECTIVE II** | | | | | | |
| **S.No.** | **Code** | **Title** | **L** | **T** | **P** | **Cr** |
| | PSE208 | Service Oriented Architecture | 3 | 0 | 2 | 4 |
| | PIS104 | Cryptography | 3 | 0 | 2 | 4 |
| | PCS106 | Parallel And Distributed Computing | 3 | 0 | 2 | 4 |
| | PIS204 | Network Security & Ethical Hacking | 3 | 0 | 2 | 4 |
| | | | | | | |
| **ELECTIVE III** | | | | | | |
| **S.No.** | **Code** | **Title** | **L** | **T** | **P** | **Cr** |
| | PCS204 | Advance Information Management System | 3 | 0 | 2 | 4 |
| | PCS205 | Big Data And Business Intelligence | 3 | 0 | 2 | 4 |
| | PCS206 | Machine Learning | 3 | 0 | 2 | 4 |
| | PSE209 | Secure Software Development and Architecture Design | 3 | 0 | 2 | 4 |
| **SEMESTER-III** | | | | | | |
| **S.No.** | **Code** | **Title** | **L** | **T** | **P** | **Cr** |
| 1 | PSE391 | Seminar | - | - | - | 4 |
| 2 | PSE392 | Capstone Project | - | - | 10 | 12 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | Dissertation (Starts) | - | - | - | - |
| | | **Total** | **-** | **-** | **-** | **16** |
| **SEMESTER-IV** | | | | | | |
| **S.No.** | **Code** | **Title** | **L** | **T** | **P** | **Cr** |
| 1 | PSE091 | Dissertation | - | - | - | 16 |
| | | **Total** | **-** | **-** | **-** | **16** |
| | | **Grand Total-Four Semester Credits** | | | | **80** |

**PCL105: Statistical Methods and Algorithms**

| L | T | P | Cr |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**Course Objective:** To learn, understand and implement different techniques related to probability distributions and statistical models.

**Introduction**: Nature and objectives of research, Study and formulation of research problem. Scope and formulation of hypothesis. Preparation and presentation of research proposal using statistical package.

**Review of Probability**: Appraisal of axiomatic approach of probability, Conditional probability, Bayes' rule, Conditional distributions, and conditional expectations. (CO1)
Markov chains: Basics of Markov chains, Finite state space, Markov chains, Transition and stationary Markov chains. Continuous time Markov process: continuous time branching processes, Kolmogorov, Forward and backward equations, Pure birth, Pure death, Birth and death process.

**Analysis of variance**: One Way Classification: ANOVA for fixed effect model, ANOVA for Random Effect Model, Two-way Classification (one observation per cell): ANOVA for fixed effect model, ANOVA for Random Effect Model.

**Design of Experiments:** Completely Randomised Design, Randomised Block Design, Latin Square Design, their statistical analysis and variance of estimates, Analysis of Covariance.

**Multivariate Data Analysis**: Introduction, multivariate normal distributions, Mean vector, Variance-covariance matrix, Correlation matrix and their estimation for multivariate data. Step wise regression, Selection of best set of variables, Classification and discrimination problems. Factor analysis and principal component analysis.Illustrative examples and Multivariate data analysis using statistical package

**Time Series and forecasting**: Components of time series, Analysis of time series, Measurement of trend, Measurement of seasonal variations, Measurement of cyclic variations, Auto-Regression Analysis, Auto-correlation, Random component in time series.

**Recommended Books**


1. Medhi, J., Stochastic Processes, New Age International
2. Montgomery, Introduction to Statistical Quality Control, John Wiley and Sons
3. Populis, A., Random Variables and Stochastic Processes, Tata McGraw Hill (2002).
4. Bhuyan K.C., Multivariate Analysis and Its Applications, New Central Book Agency (2002).

**COURSE LEARNING OUTCOMES( CLOs )**

| | |
|---|---|
| CLO1 | Basic understanding of probability distributions and statistical data analysis techniques. |
| CLO2 | Know the properties and characteristics of Markov Chain Model |
| CLO3 | Understand data classification techniques using fixed effect and random effect models |
| CLO4 | Understand time series data analysis |

**Evaluation Scheme:**

| **S.No.** | **Evaluation Elements** | **Weightage (%)** |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

| PCS104 ADVANCED DATA STRUCTURES AND ALGORITHMS | | | | |
|---|---|---|---|---|
| | **L** | **T** | **P** | **Cr** |
| | **3** | **0** | **4** | **5** |
| **Course Objective:** To learn the advanced concepts of data structure and algorithms and its implementation .The course has the main ingredients required for a computer science graduate and has all the necessary topics for assessment of data structures and algorithms. | | | | |
| **Introduction to Basic Data Structures**: Importance and need of good data structures and algorithms, Arrays, Linked lists, Stacks, Queues, Priority queues, Heaps; Strategies for choosing the appropriate data structures.<br><br>**Advanced Data Structures**: AVL Trees, Red-Black Trees, Splay Trees, B-trees, Fibonacci heaps, Data Structures for Disjoint Sets, Augmented Data Structures.<br>Algorithms Complexity and Analysis: Probabilistic Analysis, Amortized Analysis, Competitive Analysis, Internal and External Sorting algorithms: Linear Search, Binary Search, Bubble Sort, Selection Sort, Insertion Sort, Shell Sort, Quick Sort, Heap Sort, Merge Sort, Counting Sort, Radix Sort.<br><br>**Graphs & Algorithms:** Representation, Type of Graphs, Paths and Circuits: Euler Graphs, Hamiltonian Paths & Circuits; Cut-sets, Connectivity and Separability, Planar Graphs, Isomorphism,Graph Coloring, Covering and Partitioning,, Depth- and breadth-first traversals, Minimum Spanning Tree: Prim's and Kruskal's algorithms, Shortest-path Algorithms: Dijkstra's and Floyd's algorithm, Topological sort, Max flow: Ford-Fulkerson algorithm, max flow – min cut.<br><br>**String Matching Algorithms**: Suffix arrays, Suffix trees, tries, Rabin-Karp, Knuth-Morris-Pratt, Boyer-Moore algorithm.<br><br>**Approximation algorithms**: Need of approximation algorithms: Introduction to P, NP, NP-Hard and NP-Complete; Deterministic, non-Deterministic Polynomial time algorithms; Knapsack, TSP, Set Cover, Open Problems.<br><br>**Randomized algorithms**: Introduction, Type of Randomized Algorithms, Quick Sort, Min-Cut, 2-SAT; Game Theoretic Techniques, Random Walks.<br>Online Algorithms: Introduction, Online Paging Problem, Adversary Models, k-server Problem. | | | | |

**Laboratory Work**: To Implement in detail all the data structures and algorithms given above in a high level programming language.

***Recommended Books:***


1. *Thomas Coremen, "Introduction to Algorithms", Prentice Hall of India (2009).*
2. *Kleinberg J., Tardos E., "Algorithm Design", Pearson (2012).*
3. *Motwani R., Raghavan P., "Randomized Algorithms", Cambridge University Press, (1995).*
4. ***Vazirani**, Vijay V., "Approximation Algorithms", Springer, (2001).*

**Course Learning Outcomes (CLOs)**

| | |
|---|---|
| CLO1 | Implement the basic data structures, advanced data structures, Internal and External Sorting algorithms and learn the appropriate algorithmic approach to a problem. |
| CLO2 | Demonstrate the ability to evaluate algorithms, to provide justification for that selection, and to implement the algorithm in a particular context |
| CLO3 | Employ graphs to model a variety of real-world problems, synthesise tree and graph algorithms and analyze them. |
| CLO4 | Implement advance algorithmic techniques such as String Matching Algorithms, Approximation algorithms etc. |

**Evaluation Scheme:**

| **S.No.** | **Evaluation Elements** | **Weightage (%)** |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

**PSE101 SOFTWARE ENGINEERING CONCEPTS AND METHODOLOGIES**

| L | T | P | Cr |
|---|---|---|---|
| 3 | 1 | 0 | 3.5 |

**Course Objectives**: To apply principles of software development and evolution. To specify, abstract, verify and validate solutions to large-size problems, to plan, develop and manage large software and learn emerging trends in software engineering.

**Principles and Motivations**: History; definitions; Engineered approach to software development; Software development process models from the points of view of technical development and project management: waterfall, rapid prototyping, incremental development, spiral models, Aspect Software Development, Agile Software Development, Emphasis on computer-assisted environments. Selection of appropriate development process.

**Software Development Methods**: Formal, semi-formal and informal methods; Requirements elicitation, requirements specification; Data, function, and event-based modeling; Popular methodologies such as Yourdon's SAD, SSADM; Managing the Software Projects

**Software Engineering Tools and Environments**: upper and lower CASE tools, evolution of CASE tools-classification, features, strengths and weaknesses; ICASE; CASE standards. Role of the repository for supporting incremental development, software reuse

**Software Quality Assurance**: SQA Tasks, Goals and Metrics, Software Review Techniques: Informal reviews-Formal Technical Reviews, Software Reliability, Software risk management, Case Studies. Real Time Systems

**Configuration Management**: Need, Configuration management functions and activities; Configuration management techniques; Case studies.

**Software Testing Fundamentals**: Basic Terminology, Testing Techniques and strategies. Brief introduction to various standards related to Software Engineering.

***Recommended Books***

1. Pressman, Roger, Software Engineering - A Practitioners Approach, McGraw Hill, (2014).
2. Waman Jawadekar, Software Engineering: Principles & Practices, (2004)
3. Sommerville, Ian, Software Engineering, Addison-Wesley Publishing Company (2006)
4. Jalote, Pankaj, An integrated Approach to Software Engineering, Narosa (2005).

**Course Learning Outcomes (CLOs)**

| | |
|---|---|
| CLO1 | Students should be able to identify the need for engineering approach to software development and various processes of requirements analysis for software engineering problems. |
| CLO2 | Analyse various software engineering models and apply methods for design and development of software projects. |
| CLO3 | Work with various techniques, metrics and strategies for Testing software projects. |
| CLO4 | Identify and apply the principles, processes and main knowledge areas for Software Project Management |
| CLO5 | Proficiently apply standards, CASE tools and techniques for engineering software projects |

**Evaluation Scheme:**

| S.No. | Evaluation Elements | Weightage (%) |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

| PSE 105 SOFTWARE DESIGN AND CONSTRUCTION | | | | |
|---|---|---|---|---|
| | **L** | **T** | **P** | **Cr** |
| | **3** | **1** | **2** | **4.5** |

**Course Objective**: To gain knowledge of software construction fundamentals, managing construction and practical considerations related to the domain of software design and construction.

**Software Design**: Design concepts, design model, software architecture, architectural design, data design, component level design, and user interface design.

**Object Modelling and Design**: OMT, visual modelling, UML, Rational Rose Tool, Classes, objects, relationships, key abstractions, common mechanisms, diagrams, class diagrams, advanced classes, advanced relationships, interfaces, types, roles, packages, instances, object diagrams, interactions, use cases, use case diagrams, interaction diagrams, activity diagrams, events and signals, state machines, processes, threads, state chart diagrams, components, deployment, collaborations, patterns and frameworks, component diagrams, systems and models, code generation and reverse engineering.

**Software Construction**: Object-oriented approach, object-oriented programming and languages, Scope of class members-public, private, protected. Class constructor, destructor, copy constructor, virtual destructor. Derived classes, scope of derivation-public, private, protected. Virtual functions, Function overloading. Friend functions and friend classes, Operator overloading, dynamic memory allocation to classes and class members, new and delete operators. Overloading new and delete operators. Explicit type conversion operators. Input output streams, Stream class hierarchies, standard I/O objects: cin, cout, cerr, overloading <<, >> operators, File Streams, opening, reading, writing to file. File pointers and their manipulators, Introduction to templates and container classes.

**Laboratory Work**: Design and Modelling with Rational Rose, implementation-using Object oriented programming.

***Recommended Books***

1. Object-Oriented Analysis and Design with Applications, Grady Booch (2007)

2. The Unified Modelling Language User Guide, Grady Booch, James Rumbaugh, Ivar Jacobson, Addison-Wesley Professional (2005)

**COURSE LEARNIG OUTCOMES (CLOs)**

| | |
|---|---|
| CLO1 | Specify various elements of object modelling to identify, analyse, visualize, specify, model and design |
| CLO2 | Apply analysis and design principles at various levels and various views in different domains of software systems. |
| CLO3 | Represent engineering problems graphically by drawing all UML diagrams. |
| CLO4 | Identify and apply concepts of software construction like Object Oriented Programming skills |
| CLO5 | Skilful use of Rational Rose tool for drawing all the UML diagrams in order to forward and reverse engineer the complex software engineering problems. |

**Evaluation Scheme:**

| **S.No.** | **Evaluation Elements** | **Weightage (%)** |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

**PSE104 SOFTWARE PROJECT MANAGEMENT**

| L | T | P | Cr |
|---|---|---|---|
| 3 | 0 | 4 | 5 |

**Course Objective**: It gives an in depth knowledge of software project management and project planning. It also covers the Step Wise framework in project planning

**Project Planning**: Characteristics of a software project, Software scope and feasibility, resources, the SPM plan.

**Software Project Estimation**: Size/scope estimation, Decomposition techniques, WBS. Effort estimation: Sizing, Function point, LOC, FP vs LOC. Schedule estimation: GANTT Charts, Activity networks, PERT/CPM networks. Cost estimation: Models: COCOMO I, COCOMO II.

**Quality Planning**: Quality control, Quality assurance, Formal Technical Reviews, The SQA Plan, ISO and CMM standards.

**Risk Management**: Reactive vs proactive Risk strategies, Risk projection, Risk Refinement, Risk Monitoring, Monitoring and management, RMMM plan.

**Measurement and Tracking Planning**: Earned Value Analysis.

**Team Management**: Team structures: hierarchical, Egoless, chief programmer, mixed; Team software Process; Resource levelling, Building a team: Skill sets.

**Configuration Management**: Baselines, Configurable items, SCM repository, SCM process, version control change control, configuration audit.

**Project Monitoring and Control**: Audits and Reviews.

**Laboratory Work**: Implementation of software project management concepts using tools like MS Project, Rational Suite (RequisitePro, Purify, etc.), Advanced Cost Estimation Models. It should also include a micro project involving software project management.

**Recommended Books**

1. Bob Hughes and Mike Cotterell, Software Project Management, Tata McGraw Hill (2009)
2. Roger Pressman, A practitioner's Guide to Software Engineering, Tata McGraw Hill (2014 )
3. *Head First PMP: A Brain Friendly Guide To Passing The Project Management Professional Exam (2013)*

**Course Learning Outcomes (CLOs)**

| | |
|---|---|
| CLO1 | Apply the basics of Software Project Management in order to manage and deliver qualified product. |
| CLO2 | Identify the Problem Effectively and Efficiently with proper documentation for the use in different software teams and organization. |
| CLO3 | Comprehend and be able to carry on Technical as well as Cost Benefit Analysis and plan the activities within time schedules with CPM and PERT Analysis. |
| CLO4 | Competent to design Communication Plans, Procurement of Resources and Human Resource Management. |

**Evaluation Scheme:**

| **S.No.** | **Evaluation Elements** | **Weightage (%)** |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

| PSE 202 SOFTWARE VERIFICATION AND VALIDATION TESTING | | | | |
|---|---|---|---|---|
| | **L** | **T** | **P** | **Cr** |
| | **3** | **0** | **2** | **4** |
| **Course Objectives**: This course makes students understand the concepts and theory related to software testing. Understand different testing techniques used in designing test plans, developing test suites, and evaluating test suite coverage. Understand how software developers can integrate a testing framework into code development in order to incrementally develop and test code. | | | | |
| **Introduction**: Terminology, evolving nature of area, Errors, Faults and Failures, Correctness and reliability, Testing and debugging, Static and dynamic testing, Exhaustive testing: Theoretical foundations: impracticality of testing all data, impracticality of testing all paths, no absolute proof of correctness.<br><br>**Software Verification and Validation Approaches and their Applicability**: Software technical reviews; Software testing: levels of testing - module, integration, system, regression; Testing techniques and their applicability-functional testing and analysis, structural testing and analysis, error-oriented testing and analysis, hybrid approaches, integration strategies, transaction flow analysis, stress analysis, failure analysis, concurrency analysis, performance analysis; Proof of correctness; simulation and prototyping; Requirement tracing.<br><br>**Test Generation**: Test generations from requirements, Test generation pats, Data flow analysis, Finite State Machines models for flow analysis, Regular expressions based testing, Test Selection, Minimizations and Prioritization, Regression Testing.<br>Program Mutation Testing: Introduction, Mutation and mutants, Mutation operators, Equivalent mutants, Fault detection using mutants, Types of mutants, Mutation operators for C and Java.<br><br>**Laboratory Work**: To Use various verification and validation testing tools and to apply these tools on few examples and case studies | | | | |
| | | | | |
| **Recommended Books**<br>1. Marcus S. Fisher, Software Verification and Validation: An Engineering and Scientific Approach, Springer( 2007)<br>2. Aditya P. Mathur, Foundations of Software Testing, Pearson Education(2008) | | | | |

3. Srinivasan Desikan, Gopalaswamy Ramesh, Software Testing: Principles and Practices, Pearson Education India (2006)

**Course Learning Outcomes (CLOs)**

| | |
|---|---|
| CLO1 | Capable to comprehend the concepts related to theoretical foundations of testing and debugging. |
| CLO2 | Competent to know and demonstrate software verification and validation approaches and their applicability. |
| CLO3 | Proficient to formulate and generate test cases from specifications |
| CLO4 | Able to exemplify program mutation testing strategies using programming language. |
| CLO5 | Proficient to formulate and generate test cases from finite state machine model etc. |

**Evaluation Scheme:**

| S.No. | Evaluation Elements | Weightage (%) |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

## PSE205 SOFTWARE METRICS AND QUALITY MANAGEMENT

| L | T | P | Cr |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**Course Objectives**: This course aims to equip students with the knowledge and techniques of professional practices in software processes and activities.It prepares students to manage the development of high quality software using proven techniques and established standards in software quality management. It will also inculcate knowledge of different metrics associated with Software Development and evaluation.

**Software Metrics**: Measurement in software engineering, software metrics, Metrics data collection and analysis.

**Measuring internal product attributes**: Aspects of software size, length, functionality and complexity, measuring structure, types of structural measures, control-flow structure, and modularity and information flow attributes, data structures.

**Measuring external product attributes**: Modeling software quality, software reliability, software reliability problem, parametric reliability growth models, predictive accuracy, recalibration of software-reliability growth predictions, importance of operational environment, and wider aspects of software reliability.

**Metrics for object-oriented systems and component-based system**: object-oriented metrics and its characteristics various object-oriented, MOOD metrics; component-based metrics and its characteristics and various component-based suites.

**Dynamic Metrics**: Runtime Software Metrics, Extent of Class Usage, Dynamic Coupling, Dynamic Cohesion, and Data Structure Metrics.

**Software Quality:** Concepts of software quality, software quality control and software quality assurance, evolution of SQA, major SQA activities and issues, zero defect software.
Software Quality Assurance: SQA techniques; Management review process, technical review process, walkthrough, software inspection process, configuration audits, and document verification.
**Error Reporting, Trend Analysis and Corrective Action**: Identification,Analysisand Correction of defect, implementation of correction, regression testing; Categorization of defect w.r.t development phases; Error quantity, error frequency, program unit complexity, compilation frequency; Corrective action and documenting the corrective action, periodic review of actions taken.

**Case Studies**: CASE tools, Quality management standards, Quality standards with emphasis on ISO approach, Capability Maturity Models-CMM and CMMI, TQM Models, Bootstrap methodology, The SPICE project, ISO/IEC 15504, Six Sigma Concept for Software Quality.

**Lab Work**: To Work on small projects, build metrics and analyze, check the quality of the projects and do a comparative study with other projects

**Recommended Book*s***

1. Practical Guide to Software Quality Management (Artech House Computing Library)( 2003)
2. Quality Software Management, Volume 1: Systems Thinking, Dorset House Publishing(2011)
3. Metrics and Models in Software Quality Engineering , Pearson, (2003).
4. Applied Software Measurement by Capers Jones, Tata McGraw Hill, (2008*)*

**Course Learning Outcomes (CLOs)**

| | |
|---|---|
| CLO1 | Acquired basic knowledge of Software quality models |
| CLO2 | Exemplify Quality measurement and metrics, Quality plan and implementation |
| CLO3 | Articulate Quality control and reliability of quality process and Quality management system models |
| CLO4 | Articulate Complexity metrics and Customer Satisfaction and International quality standards – ISO, CMM |
| CLO5 | Control and Manage the project and processes, apply configuration management on the basis of collected metrics. |

**Evaluation Scheme:**

| **S.No.** | **Evaluation Elements** | **Weightage (%)** |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

| PSE214 ENGINEERING DESIGN PROJECT | | | | |
|---|---|---|---|---|
| | **L** | **T** | **P** | **Cr** |
| | **1** | **0** | **4** | **5** |

**Course Description**: The design project is a post-graduate level course generally consisting of a team-based semester long project and emphasizes on both technical and managerial skills. The computer science is very fundamental core discipline with having balanced execution of theoretical and practical concepts of all computer subjects and hence there is an emphasis on hands-on development, process, and usage of various fundamental tools in addition to theory and basic concepts. Students will be involved in examining; analysing and reporting different target oriented projects. In this project student can get more exposure to work in various advanced areas of algorithms, database, Software Engineering, Software Design and construction, Project Management, Software Metrics and Quality Management etc.

**Course Learning Outcomes (CLOs):**

| | |
|---|---|
| CLO1 | Learn the Application of advanced programming techniques, tools and methodologies |
| CLO2 | Encourage group working and project planning. |
| CLO3 | Learn the principles of software systems design including user interface design. |
| CLO4 | Analyze the design and optimize it with respect to requirements. |
| CLO5 | To be skilful the requirements of project documentation. |
| CLO6 | Demonstrate, professional report writing skills, communication skills and team skills. |

**Evaluation Scheme:**

- Progress Evaluation :

Every month, there will regular progress evaluation of the project based on various parameters like problem definition, design etc.

- Final Evaluation :

1.Project report
2.Presentation (may include demonstration)
3.Demonstration of the project
4. Viva (answers to the queries)

Reflective diary
Poster presentation
Video presentation
Peer review
nal Evaluation : 60 Marks
ntinuous Evaluation: 40 Marks ( At least spread in two evaluations)

| PSE 204 ADVANCED TOPICS IN SOFTWARE ENGINEERING | | | | |
|---|---|---|---|---|
| | **L** | **T** | **P** | **Cr** |
| | **3** | **0** | **2** | **4** |

**Course Objectives**: To apply advance topics in software engineering. To specify, abstract, verify and validate solutions to large-size problems, to plan, develop and manage large software using state-of-the-art methodologies and learn emerging trends

**Formal Methods**: Basic concepts, mathematical preliminaries, Applying mathematical notations for formal specification, formal specification languages, using Z to represent an example software component, the ten commandments of formal methods

**Cleanroom Software Engineering**: approach, functional specification, design and testing. Component-Based Software Engineering: CBSE process, domain engineering, component-based development, classifying and retrieving components, and economics of CBSE. Client/Server Software Engineering: Structure of client/server systems, software engineering for Client/Server systems, analysis modelling issues, design and testing issues.

**Web Engineering**: Attributes of web-based applications, the WebE process, a framework for WebE, formulating, analysing web-based systems, design and testing for web-based applications, Management issues.

**Reengineering**: Business process reengineering, software reengineering, reverse reengineering, restructuring, forward reengineering, Economics of reengineering.

**Computer-Aided Software Engg**: Building blocks and taxonomy for CASE, integrated CASE environments, integration architecture, CASE repository, case Study of tools like TCS Robot.

**Mobile Development Process**: Model View Controller, Presentation Abstraction Control, UML based development, Use cases, Testing: Mobile infrastructure, Validating use cases, Effect of dimensions of mobility on testing, Case study: IT company, design, Implementation.

**Real Time Operating Systems**: Real-time and non-real time applications. Classification of Real-Time Task scheduling algorithms, Event-driven scheduler- Simple priority-based, Rate Monotonic Analysis, Earliest Deadline First, The simplest of Task assignment and scheduling, priority scheduling, characteristics of tasks, task assignment and multi-tasking.

**Software Engineering Issues in Embedded Systems**: Characteristics, I/O, Embedded systems/real time systems. Embedded software architecture, control loop, interrupts control system, co-operating multitasking, pre-emptive multitasking, Domain analysis, Software element analysis, requirement analysis, Specification, Software architecture, Software analysis design, implementation, testing, validation, verification and debugging of embedded systems.

**Laboratory Work**: To implement the advance concepts in the lab using related tools and to develop the project using related technologies

**Recommended Books**

1. Software Engineering a Practitioners Approach, Roger S. Pressman, McGraw-Hill, 8th Edition( 2014)
2. Formal Specification and Documentation using Z - A Case Study Approach, J.Bowan, International Thomson Computer Press (2003)
3. Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications, Robert Oshana, Mark Kraeling, Newnes Publisher (2013)

**COURSE LEARNING OUTCOMES (CLOs):**

| | |
|---|---|
| CO1 | Acquire knowledge on the wider perspective of software engineering and architecture issues |
| CO2 | Implement the mathematical notation of the software systems through formal methods. |
| CO3 | Design and construct the software systems using reusable software "components" by acquiring the knowledge about domain engineering and component based development |
| CO4 | Merge the conventional principles, concepts and methods in software engineering with the elements of object oriented and CBSE to create client/server systems. |
| CO5 | Create high quality web applications by using software engineering concepts and principles like formulation, planning, analysis testing and evaluation. |

**Evaluation Scheme:**

| S.No. | Evaluation Elements | Weightage (%) |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

## PSE206 AGILE SOFTWARE DEVELOPMENT APPROACHES

| L | T | P | Cr |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**Course Objectives**: This course makes student learn the fundamental principles and practices associated with each of the agile development methods. To apply the principles and practices of agile software development on a project of interest and relevance to the student.

**Agile Software Development**: Basics and Fundamentals of Agile Process Methods, Values of Agile, Principles of Agile, stakeholders, Challenges

**Lean Approach**: Waste Management, Kaizen and Kanban, add process and products add value. Roles related to the lifecycle, differences between Agile and traditional plans, differences between Agile plans at different lifecycle phases. Testing plan links between testing, roles and key techniques, principles, understand as a means of assessing the initial status of a project/ How Agile helps to build quality

**Agile and Scrum Principles**: Agile Manifesto, Twelve Practices of XP, Scrum Practices, Applying Scrum. Need of scrum, working of scrum, advanced Scrum Applications, Scrum and the Organization, scrum values

**Agile Product Management**: Communication, Planning, Estimation Managing the Agile approach Monitoring progress, Targeting and motivating the team, Managing business involvement, Escalating issue. Quality, Risk, Metrics and Measurements, Managing the Agile approach Monitoring progress, Targeting and motivating the team, Managing business involvement and Escalating issue

**Agile Requirements**: User Stories, Backlog Management. Agile Architecture: Feature-Driven Development. Agile Risk Management: Risk and Quality Assurance, Agile Tools

**Agile Testing**: Agile Testing Techniques, Test-Driven Development, User Acceptance Test

**Agile Review**: Agile Metrics and Measurements, The Agile approach to estimating and project variables, Agile Measurement, Agile Control: the 7 control parameters. Agile approach to Risk, The Agile approach to Configuration Management, The Atern Principles,Atern Philosophy,The rationale for using Atern, Refactoring, Continuous integration, Automated Build Tools

**Scaling Agile for large projects**: Scrum of Scrums, Team collaborations, Scrum, Estimate a Scrum Project, Track Scrum Projects, Communication in Scrum Projects, Best Practices to Manage Scrum.

**Laboratory Work**: Exploring the tools related to Agile Development and approached and develop small projects using this technology

**Recommended Books**

1. Robert C. Martin ,Agile Software Development, Principles, Patterns, and Practices

| |
|---|
| Alan Apt Series (2011)<br>2. Succeeding with Agile : Software Development Using Scrum, Pearson (2010) |

**Course Learning Outcomes (CLOs)**

| | |
|---|---|
| CLO1 | Analyze existing problems with the team, development process and wider organization |
| CLO2 | Apply a thorough understanding of Agile principles and specific practices |
| CLO3 | Select the most appropriate way to improve results for a specific circumstance or need |
| CLO4 | Judge and craft appropriate adaptations to existing practices or processes depending upon analysis of typical problems |
| CLO5 | Evaluate likely successes and formulate plans to manage likely risks or problems |

**Evaluation Scheme:**

| **S.No.** | **Evaluation Elements** | **Weightage (%)** |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

| PSE207 COMPONENT BASED DEVELOPMENT | | | | |
|---|---|---|---|---|
| | **L** | **T** | **P** | **Cr** |
| | **3** | **0** | **2** | **4** |

**Course Objectives**: It is to gain the knowledge of current component models in terms of their design, management and related issues. The students will be able to assess that how these models measure up to the goals of CBD

**Component Definition**: Definition of Software Component and its Elements. Component Models and Component Services: Concepts and Principles, COTS Myths and Other Lessons Learned in Component-Based Software Development, Roles for Component-Based Development, Common High Risk Mistakes in Component-Based Software Engineering, CBSE Success Factors: Integrating Architecture, Process, and Organization.

**Software Engineering Practices**: The Practice of Software Engineering, From Subroutines to Subsystems: Component-Based Software Development.

**The Design of Software Component Infrastructures**: Software Components and the UML, Component Infrastructures: Placing Software Components in Context, Business Components, Components and Connectors: Catalysis Techniques for Defining Component Infrastructures, An Open Process for Component-Based Development, Designing Models of Modularity and Integration.

**The Management Of Component-Based Software Systems**: Measurement and Metrics for Software Components, The Practical Reuse of Software Components, Selecting the Right COTS Software: Why Requirements are Important, Software Component Project Management Processes, The Trouble with Testing Software Components, configuration Management and Component Libraries, The Evolution, Maintenance and Management of Component-Based Systems.

**Component Technologies**: Overview of the CORBA Component Model, Transactional COM+: Designing Scalable Applications, The Enterprise JavaBeans Component Model, Choosing Between COM+, EJB, and CCM, Software Agents as Next Generation Software Components.

**Legal and Regulatory**: CBSE as a Unique Engineering Discipline, The Future of Software Components: Standards and Certification, Commercial Law Applicable to Component-Based Software, The Effects of UCITA on Software Component Development and Marketing, Future of CBSE.

**Laboratory Work**: Practice, Implementation and working of Component Based Development tools and technologies

**Recommended Books**

1. Addison Wilsey , Component-Based Development: Principles and Planning for Business Systems( 2010)
2. Don Box, Dorling Kingsley, Essential COM (2006.)

**Course Learning Outcomes (CLOs)**

| | |
|---|---|
| CLO1 | Familiarization with Component Based Systems, their Purpose and Scope. |
| CLO2 | AnalyseSoftware Engineering Practices related to CBD. |
| CLO3 | Apply design Of Software Component Infrastructures |
| CLO4 | IdentifyComponent Based Development Technologies |
| CLO5 | Relate the concept of Legal and regulatory framework related to CBD |

**Evaluation Scheme:**

| S.No. | Evaluation Elements | Weightage (%) |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

| PCS-224 NATURAL LANGUAGE PROCESSING | | | | |
|---|---|---|---|---|
| | **L** | **T** | **P** | **Cr** |
| | **3** | **0** | **2** | **4** |

**Course Objectives**: To understand the advanced concepts of Natural Language Processing and to be able to apply the various concepts of NLP in other application areas.

**Introduction**: Origin of Natural Language Processing (NLP), Challenges of NLP, NLP Applications, Processing Indian Languages.

**Words and Word Forms**: Morphology fundamentals; Morphological Diversity of Indian Languages; Morphology Paradigms; Finite State Machine Based Morphology; Automatic Morphology Learning; Shallow Parsing; Named Entities; Maximum Entropy Models; Random Fields, Scope Ambiguity and Attachment Ambiguity resolution.

**Machine Translation**: Need of MT, Problems of Machine Translation, MT Approaches, Direct Machine Translations, Rule-BasedMachine Translation, Knowledge Based MT System, Statistical Machine Translation, UNL Based Machine Translation, Translation involving Indian Languages.

**Meaning**: Lexical Knowledge Networks, WorldNet Theory; Indian Language Word Nets and Multilingual Dictionaries; Semantic Roles; Word Sense Disambiguation; WSD and Multilinguality; Metaphors.

**Speech Recognition**: Signal processing and analysis method, Articulation and acoustics, Phonology and phonetic transcription, Word Boundary Detection; Argmax based computations; HMM and Speech Recognition.

**Other Applications**: Sentiment Analysis; Text Entailment; Question Answering in Multilingual Setting; NLP in Information Retrieval, Cross-Lingual IR.

**Laboratory Work**: To implement Natural language concepts and computational linguistics concepts using popular tools and technologies. To implement key algorithms used in Natural Language Processing.

***Recommended Books:***

1. Siddiqui and Tiwary U.S., Natural Language Processing and Information Retrieval, Oxford University Press (2008).
2. Allen J., Natural Language understanding, Benjamin/Cunnings, (1987).
3. Jensen K., Heidorn G.E., Richardson S.D., Natural LanguageProcessing: The PLNLP Approach, Springer (2013).
4. Roach P., Phonetics, Oxford University Press (2012).

**Course Learning Outcomes (CLOs)**

| | |
|---|---|
| CLO1 | Comprehend the concept of Natural Language Processing (NLP), its challenges and applications. |
| CLO2 | Process words and word forms of the language by considering its morphology, paradigms and named entities. |
| CLO3 | Demonstrate and implement the use of machine translation by using rule-based MT, Knowledge Based MT and Statistical Machine Translation etc. |
| CLO4 | Comprehend the concepts of WorldNet, Semantic Roles and Word Sense Disambiguation |
| CLO5 | Demonstrate the use of NLP in speech recognition and other emerging applications like Sentiment Analysis, Information Retrieval etc. |

**Evaluation Scheme:**

| S.No. | Evaluation Elements | Weightage (%) |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

**PSE 208 SERVICE OIRENTED ARCHITECTURE**

| L | T | P | Cr |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**Course Objectives**: To introduce the concepts and design principles of SOA, Non-technical aspects such as governance, impact on culture and organization, as well as the various interoperability standards, technology infrastructure and security considerations associated with SOA implementations.

**Introduction**: Roots, Characteristics and Anatomy of SOA, Comparing SOA to client-server and distributed internet architectures, SOA component interrelation, Principles of service orientation

**Service Oriented Architecture**: Major components of the architecture SOAP, XML, HTTP, Cookies, WSDL, XML schema, UDDI, Interactions between components.

**Introduction to Web services** : Service descriptions, Messaging with SOAP,Message exchange Patterns, Coordination,Atomic Transactions, Business activities, Orchestration, Choreography,Service layer abstraction, Application Service Layer, Business Service Layer, Orchestration Service Layer

**Analysis**: Service oriented analysis, Business-centric SOA, Deriving business services-service modelling, Service Oriented Design, WSDL basics, SOAP basics, SOA composition guidelines, Entity-centric business service design, Application service design, Task centric business service design

**SOA platform basics**: SOA support in J2EE, Java API for XML-based web services (JAX-WS), Java architecture for XML binding (JAXB), Java API for XML Registries (JAXR), Java API for XML based RPC (JAX-RPC), Web Services Interoperability Technologies (WSIT), SOA support in .NET, Common Language Runtime, ASP.NET web forms, ASP.NET web services, Web Services Enhancements (WSE)

**Security:** WS-BPEL basics, WS-Coordination overview, WS-Choreography, WS-Policy, WS-Security

**Laboratory work**: Installing and configuring web servers, building and implementing Web services using the latest tools (.NET, J2EE).

**Recommended Books**


1. Rick Sweeney, Achieving Service-Oriented Architecture: Applying an Enterprise Architecture Approach, (2010)
2. Thomas Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Pearson Education, (2005)

**Course Learning Outcomes (CLOs)**

| | |
|---|---|
| CLO1 | Analyze functions of Service Oriented Architecture and identify the ways in which they can benefit organizations and study the comparison of web services with other technologies. |
| CLO2 | Evaluate the design of SOA, Major components of the architecture SOAP, XML, HTTP, Cookies, WSDL, XML schema, UDDI and Interactions between various components. |
| CLO3 | Learn some of Semantic Web technologies and applications with knowledge of XML's, Grammar rules, namespace schema. |
| CLO4 | Create web services and web services clients with state-of-the-art tools along |
| CLO5 | Exemplify the web service interoperability, security, and future of web services with the implementation of cloud computing |

**Evaluation Scheme:**

| **S.No.** | **Evaluation Elements** | **Weightage (%)** |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

| PIS 104: CRYPTOGRAPHY | | | | |
|---|---|---|---|---|
| | **L** | **T** | **P** | **Cr** |
| | **3** | **0** | **2** | **4** |

**Course Objectives**: This course provides an introduction to the core cryptographic techniques and the different security services they provide, together with an overview of the key management principles and techniques. It also gives an insight into the design and working of different cryptographic methods.

**Introduction:** History and Overview of Cryptography, Historical Ciphers and Their Cryptanalysis, Definition of Perfect Secrecy, Shanon's Theorem, Basic Principles of Modern Cryptography

**Private Key Cryptography**: Private Key Encryption, Computational Approach to Cryptography, Pseudo Randomness, Constructing Secure Encryption Schemes, Chosen Plaintext Attacks, CPA Secure Encryption Schemes, Chosen Cipher Text Attacks, Security Against CCA, Limitations of Private Key Cryptography

**Message Integrity:** Definition and Applications, Message Authentication Codes, Constructing Secure Message Authentication Codes, Collision Resistant Hashing, Collision Resistant Hashing Functions

**Block Ciphers**: Feistel Networks, Data Encryption Standard (DES): Design and Security, Advanced Encryption Standard: Design and Security, One Way Functions, Construction of Pseudorandom Generators, Construction of Pseudorandom Functions, Construction of Pseudorandom Permutations

**Public Key Cryptography**: Basic Group Theory, Primes, Factoring, Cyclic Groups, Discrete Logarithms, Cryptography Using Arithmetic Modulo Primes, Arithmetic Modulo Composites, RSA Public Key Encryption, Security Against Active Attacks, Attacks on RSA, El Gamal Encryption Schemes, Recent Public Key Encryption Schemes

**Digital Signatures**: Definitions and Applications, Lamport and Merkle Schemes. Overview of Signatures Based on Discrete-Log Certificates and Trust Management., SSL/TLS and Ipsec, Privacy Mechanisms

**Advanced Topics**: ECC, DNA Cryptography, Quantum Cryptography, Digital Watermarking and Steganography etc.

**Recommended Books**

1. J. Katz and Y. Lindell, Introduction to modern cryptography, Chapman and Hall/CRC, (2014)
2. A. Menezes, P. Vanoorschot, S. Vanstone, Handbook of applied cryptography by CRC Press ( 1996)
3. William Stalling, Cryptography and network security: principles and practice, Prentice Hall (2013)

**Course Learning Outcomes (CLOs)**

| | |
|---|---|
| CLO1 | Compare and contrast a range of different cryptosystems from an applied viewpoint. |
| CLO2 | Identify the different approaches to quantifying secrecy |
| CLO3 | Recognize the different modes of operation for block ciphers and their applications and understand the role of hash functions in information security. |
| CLO4 | Account for the cryptographic theories, principles and techniques that are used to establish security properties |
| CLO5 | Analyze and use methods for cryptography and reflect about limits and applicability of methods |

**Evaluation Scheme:**

| S.No. | Evaluation Elements | Weightage (%) |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

## PCS106 PARALLEL AND DISTRIBUTED COMPUTING

| L | T | P | Cr |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**Course Objective**: To learn the advanced concepts of Parallel and Distributed Computing and its implementation for assessment of understanding the course by the students

**Introduction**: Scope, issues, applications and challenges of Parallel and Distributed Computing

**Parallel Programming Platforms**: Implicit Parallelism: Trends in Microprocessor Architectures, Dichotomy of Parallel Computing Platforms, Physical Organization, Communication Costs in Parallel Machines, Routing Mechanisms for Interconnection Networks, GPU, co-processing.

**Principles of Parallel Algorithm Design**: Decomposition Techniques,Characteristics of Tasks and Interactions,Mapping Techniques for Load Balancing.

J**DA programming model**: Overview of CUDA, Isolating data to be used by parallelized code, API function to allocate memory on parallel computing device, to transfer data, Concepts of Threads, Blocks, Grids, Developing a kernel function to be executed by individual threads, Execution of kernel function by parallel threads, transferring data back to host processor with API function.
Analytical Modelling of Parallel Programs:Sources of Overhead in Parallel Programs,Performance Metrics for Parallel Systems,The Effect of Granularity on Performance,Scalability of Parallel Systems,Minimum Execution Time and Minimum Cost-Optimal Execution Time

**Dense Matrix Algorithms**: Matrix-Vector Multiplication,Matrix-Matrix Multiplication, Issues in Sorting on Parallel Computers, Bubble Sort and Variants, Quick Sort,Other Sorting Algorithms

**Graph Algorithms**: Minimum Spanning Tree: Prim's Algorithm,Single-Source Shortest Paths: Dijkstra's Algorithm,All-Pairs Shortest Paths,Transitive Closure,Connected Components,Algorithms for Sparse Graph

**Search Algorithms for Discrete Optimization Problems**: Sequential Search Algorithms,Parallel Depth-First Search,Parallel Best-First Search,Speedup Anomalies in Parallel Search Algorithms

**Laboratory Work** : To Implement the algorithms with the help of CUDA programming using parallel and distributed programming techniques

**Recommended Books**:

1. A Grama, A Gupra, G Karypis, V Kumar. Introduction to Parallel Computing (2nd Ed.). Addison Wesley(2003).
2. C Lin, L Snyder. Principles of Parallel Programming. USA: Addison-Wesley Publishing Company(2008).

3. J Jeffers, J Reinders. Intel Xeon Phi Coprocessor High-Performance Programming. Morgan Kaufmann Publishing and Elsevier(2013).
4. T Mattson, B Sanders, B Massingill. Patterns for Parallel Programming. Addison-Wesley Professional(2004).

**Course Learning Outcomes (CLOs)**

| | |
|---|---|
| CLO1 | Learn the concepts, issues and tasks in parallel and distributed computing along with different parallel architectures |
| CLO2 | Demonstrate the principles for Parallel Algorithm Design. |
| CLO3 | Explore the parallel programming models and algorithms for common operations. |
| CLO4 | Analyze the application of parallel algorithms to solve the complex computational problems. |
| CLO5 | Implement various parallel algorithms with CUDA. |

**Evaluation Scheme:**

| **S.No.** | **Evaluation Elements** | **Weightage (%)** |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

| PIS204:NETWORK SECURITY AND ETHICAL HACKING | | | | |
|---|---|---|---|---|
| | **L** | **T** | **P** | **Cr** |
| | **3** | **0** | **2** | **4** |

**Course Objectives**: This course is designed to impart a critical theoretical and detailed practical knowledge of a range of computer network security technologies as well as network security tools and services related to ethical hacking.

**Introduction**: Security, Functionality and ease of use Triangle, Essential Terminology, Elements of Security, Difference between Penetration Testing and Ethical Hacking, Deliverables ethics and legality, Computer Crimes and Implications.

**Reconnaissance:** Information Gathering Methodology, Locate the Network Range, Active and Passive reconnaissance

**Scanning**: Scanning, Elaboration phase, active scanning, scanning tools NMAP, hping2. Enumeration, DNS Zone transfer. Detecting live systems on the network, Discovering services running /listening on target systems, Understandingport scanning techniques, Identifying TCP and UDP services running on the network, Active and passive fingerprinting

**Trojans and Backdoors**: Effect on Business, Trojan, Overt and Covert Channels, Working of Trojans, Different Types of Trojans, Different ways a Trojan can get into a system, Indications of a Trojan Attack, Some famous Trojans and ports used by them

**Sniffers**: Definition of sniffing, Sniffer working, Passive Sniffing, Active Sniffing, Ethreal tool, Man-in-the-Middle Attacks, Spoofing and Sniffing Attacks, ARP Poisoning and countermeasures. Denial of Service:Goal of DoS (Denial of Service), Impact and Modes of Attack.

**Social Engineering**: Social Engineering, Art of Manipulation, Human Weakness, Common Types of Social Engineering, Human Based Impersonation, Example of Social Engineering, Computer Based Social Engineering, Reverse Social Engineering, Policies and Procedures, Security Policies-checklist

**Session Hijacking**: Understanding Session Hijacking, Spoofing vs Hijacking, Steps in Session Hijacking, Types of Session Hijacking, TCP Concepts 3 Way and shake, Sequence numbers

**Ethical Hacking**: System Hacking and Hacking Wireless Networks: Aspect of remote password guessing, Role of eavesdropping, Various methods of password cracking, Keystroke Loggers, Understanding Sniffers, Comprehending Active and Passive Sniffing, ARP Spoofing and Redirection, DNS and IP Sniffing, HTTPS Sniffing. Introduction to 802.11, Role of WEP, Cracking WEP Keys, Sniffing Traffic, Wireless DOS attacks, WLAN Scanners, WLAN Sniffers, Hacking Tools, Securing Wireless Networks.

**Laboratory work**: deals with launching different types of attacks and creating a network blueprint of an organization.

**Recommended Books**

1. *Eric Core, Hackers Beware, EC-Council Press, (2003)*
2. *William Stallings, Network Security Essentials, Prentice Hall, ( 2013)*
3. *William R. Cheswick and Steven M. Bellovin, Firewalls and Internet Security, Addison-Wesley Professional, ( 2003.)*
4. W. Stallings, Cryptography and Network Security, Prentice Hall (2010)

**Course Learning Outcomes (CLOs)**

| | |
|---|---|
| CLO1 | Demonstrate knowledge of various vulnerabilities in network applications. |
| CLO2 | Practice awareness of various malicious content and guiding ways for protection against the same. |
| CLO3 | Demonstrate knowledge of various forms of attacks. |
| CLO4 | Recall judicious and ethical use of various tools. |
| CLO5 | Expertise in the techniques of system hacking and hacking over a wireless network. |

**Evaluation Scheme:**

| S.No. | Evaluation Elements | Weightage (%) |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

## PCS204 ADVANCED INFORMATION MANAGEMENT SYSTEMS

| L | T | P | Cr |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**Course Objective**: To learn the advanced concepts of database information and management and its implementation for assessment of understanding the course by the students

**Transaction Processing and Concurrency Control Techniques**: Introduction to Transaction Processing, Properties and states of Transactions, Scheduling of transactions, Serializability of Schedules, Locking Techniques for Concurrency Control, Two phase locking techniques.

**Database Recovery Techniques**: Recovery Concepts, Recovery Techniques Based on Deferred Update, Techniques Based on Immediate Update.

**Distributed DBMS**: Introduction, functions and architecture of a DDBMS, distributed relational database design, Transparencies in a DDBMS, Distributed transaction management, distributed concurrency control, distributed deadlock management, distributed database recovery.

**Object-Oriented DBMS and NoSQL**: Advanced database applications, weakness of RDBMS, next-generation database systems, OODBMS perspectives, persistence, advantages and disadvantages of OODBMS, Object-oriented database design, Object oriented extensions in Oracle, Comparison of ORDBMS and OODBMS.

**Need of NoSQL and Its Data Models**: Key- value data model, Document data model, Column family data model, Graph data models, CAP Theorem

**Data Warehousing Concepts, OLAP and Data mining**: Evolution of data warehousing, data warehousing concepts, benefits and problems of data warehousing, comparison of OLTP systems and data warehousing, On-Line Analytical Processing, Introduction to data mining.

**Laboratory Work**: To Implement Different concepts of Advanced Information Management Systems through sample programs and small projects to understand the techniques in a practical manner.

**Recommended Books**:


1. Thomas Connolly, Carolyn Begg, “Database Systems”, Pearson Education, (2005)
2. Pramod J Sadalage and Martin Fowler, “NoSQL Distilled”, Pearson, (2012)
3. Hoffer, Prescott, Mcfadden, “Modern Database Management”, Pearson Education Asia, (2007)
4. Ivan Bayross, “SQL and PL/SQL”, BPB Publication, ( 2010)

**Course Learning Outcomes (CLOs)**

| | |
|---|---|
| CLO1 | Comprehend techniques of Transaction Processing, Concurrency Control and Database Recovery Technique. |
| CLO2 | Design Distributed database and apply concurrency control and recovery of data on distributed database. |
| CLO3 | Comprehend the concept of Object-Oriented DBMS and NoSQL data models. |
| CLO4 | Comprehend the need of Data Warehousing Concepts, OLAP and Data mining. |
| CLO5 | Demonstrate use of PL/SQL to develop database centric applications. |

**Evaluation Scheme:**

| **S.No.** | **Evaluation Elements** | **Weightage (%)** |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

| PCS 205 BIG DATA ANALYTICS AND BUSINESS INTELLIGENCE | | | | |
|---|---|---|---|---|
| | **L** | **T** | **P** | **Cr** |
| | **3** | **0** | **2** | **4** |

**Course Objective**: To have an advanced level of understanding of most recent advancements in Big Data and using insights, statistical models, visualization techniques for its effective application in Business intelligence.

**Introduction to Data Analytics**: Data and Relations, Data Visualization, Correlation, Regression, Forecasting, Classification, Clustering.

**Big Data Technology Landscape**: Fundamentals of Big Data Types, Big data Technology Components, Big Data Architecture, Big Data Warehouses, Functional vs. Procedural Programming Models for Big Data.

**Introduction to Business Intelligence**: Business View of IT Applications, Digital Data, OLTP vs. OLAP, BI Concepts, BI Roles and Responsibilities, BI Framework and components, BI Project Life Cycle, Business Intelligence vs. Business Analytics.
Big Data Analytics: Big Data Analytics, Framework for Big Data Analysis, Approaches for Analysis of Big Data, ETL in Big Data, Introduction to Hadoop Ecosystem, HDFS, Map-Reduce Programming, Understanding Text Analytics and Big Data, Predictive analysis on Big Data, Role of Data analyst.

**Business implementation of Big Data**: Big Data Implementation, Big Data workflow, Operational Databases, Graph Databases in a Big Data Environment, Real-Time Data Streams and Complex Event Processing, Applying Big Data in a business scenario, Security and Governance for Big Data, Big Data on Cloud, Best practices in Big Data implementation, Latest trends in Big Data, Latest trends in Big Data, Big Data Computation, More on Big Data Storage, Big Data Computational Limitations.

**Laboratory Work**: Introduction, use and assessment of most recent advancements in Big Data technology along with their usage and implementation with relevant tools and technologies.

**Recommended books:**


1. Minelli M., Chambers M., Dhiraj A., Big Data, Big Analytics: Emerging Business
2. Intelligence and Analytic Trends for Today's Businesses, Wiley CIO Series (2013),
3. White T., Hadoop: The Definitive Guide, O' Reilly Media (2012).

**Course Learning Outcomes (CLOs)**

| | |
|---|---|
| CLO1 | Comprehend the concepts of big data, architecture and environment, digital data types, structure and its implementation. |
| CLO2 | Explore the advanced level of understanding of the usage of Big Data in present World. |
| CLO3 | Comprehend the concepts of Map-Reduce, HDFS command and Hadoop services and its implementation. |
| CLO4 | Analyze big data, create statistical models, and identify insights that can lead to actionable results |
| CLO5 | Use software tools such as R and Hadoop, in text analytics. |

**Evaluation Scheme:**

| S.No. | Evaluation Elements | Weightage (%) |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

**PCS 206 MACHINE LEARNING**

| L | T | P | Cr |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**Course Objectives**: This course provides an advanced level of understanding to machine learning and statistical pattern recognition. It offers some of the most cost-effective approaches to automated knowledge acquisition in emerging data-rich disciplines and focuses on the theoretical understanding of these methods, as well as their computational implications.

**troduction**: Well-Posed learning problems, Basic concepts, Designing a learning system, Issues in machine learning. Types of machine learning: Learning associations, Supervised learning (Classification and Regression Trees, Support vector machines), Unsupervised learning (Clustering), Instance-based learning (K-nearest Neighbor, Locally weighted regression, Radial Basis Function), Reinforcement learning (Learning Task, Q-learning, Value function approximation, Temporal difference learning).

**Decision Tree Learning**: Decision tree representation, appropriate problems for decision tree learning, Univariate Trees (Classification and Regression), Multivariate Trees, Basic Decision Tree Learning algorithms, Hypothesis space search in decision tree learning, Inductive bias in decision tree learning, Issues in decision tree learning.

**Bayesian Learning**: Bayes theorem and concept learning, Bayes optimal classifier, Gibbs algorithms, Naive Bayes Classifier, Bayesian belief networks, The EM algorithm.

**Artificial Neural Network**: Neural network representation, Neural Networks as a paradigm for parallel processing, Linear discrimination, Pairwise separation, Gradient Descent, Logistic discrimination, Perceptron, Training a perceptron, Multilayer perceptron, Back propagation Algorithm. Recurrent Networks, Dynamically modifying network structure.

**Genetic Algorithms**: Basic concepts, Hypothesis space search, Genetic programming, Models of evolution and learning, Parallelizing Genetic Algorithms.
Inductive and Analytical Learning: Learning rule sets, Comparison between inductive and analytical learning, Analytical learning with perfect domain theories: Prolog-EBG. Inductive-Analytical approaches to learning, Using prior knowledge to initialize hypothesis (KBANN Algorithm), to alter search objective (Tangent Prop and EBNN Algorithm), to augment search operators (FOCL Algorithm).

**Design and Analysis of Machine Learning Experiments**: Guidelines for machine learning experiments,Factors, Response, and Strategy of experimentation, Cross-Validation and Resampling methods, measuring classifier performance, Hypothesis testing, Assessing a classification algorithm's performance, Comparing two classification algorithms, Comparing multiple algorithms: Analysis of variance, Comparison over multiple datasets.

| |
|---|
| **Laboratory Work**: It is concerned with the design, analysis, implementation, and applications of programs that learn from experience. Learning algorithms can also be used to model aspects of human and animal learning. |
| **Recommended Books**<br>1. Mitchell T.M., Machine Learning, McGraw Hill (1997).<br>2. Alpaydin E., Introduction to Machine Learning, MIT Press (2010).<br>3. Bishop C., Pattern Recognition and Machine Learning, Springer-Verlag (2006).<br>4. Michie D., Spiegelhalter D. J., Taylor C. C., Machine Learning, Neural and Statistical Classification. Overseas Press (2009). |

**Course Learning Outcomes (CLOs)**

| | |
|---|---|
| CLO1 | Demonstrate in-depth knowledge of methods and theories in the field of machine learning. |
| CLO2 | Demonstrate the use Bayesian perspective on machine learning, Artificial neural networks, back propagation algorithm |
| CLO3 | Assess the learning algorithms modelled after biological evolution, including genetic algorithms and genetic programming. |
| CLO4 | Demonstrate the ability to critically evaluate and compare different learning models and learning algorithms. |
| CLO5 | Design new algorithms after combining some of the key elements of existing machine learning algorithms. |

**Evaluation Scheme:**

| S.No. | Evaluation Elements | Weightage (%) |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

**PSE209: SECURE SOFTWARE DEVELOPMENT AND ARCHITECTURE DESIGN**

| L | T | P | Cr |
|---|---|---|---|
| 3 | 0 | 2 | 4 |

**Course Objectives**: Students will learn that how the security aspects of software development are embedded into the system to be developed. It includes secure architecture design, secure coding, secure deployment and secure software development methodologies.

**Introduction & Motivation**: Hacker vs. Cracker, Historical Background, Mode of Ethical Hacking, Hacker Motive, Gathering Information, Secure Software, Compliance Requirements, C-Level Language, Assets, Threats and Risks, Security Requirements, Confidentiality, Integrity, Availability

**Secure Software Development Methodologies**: Secure Software Development Lifecycle (SSDLC), Guidelines for Secure Software, SD-3 Principles, Security Practices, Secure coding standards, OWASP, ISO15408, Common Criteria (CC), build-insecurity

**Requirements Engineering**: Availability, Authenticity, Confidentiality, Efficiency, Integrity, Maintainability, Portability, Reliability, Requirements Engineering, Trustworthiness, Threat Analysis and Risk Management

**Secure Architectural Design**: Threat Modelling, Asset, Threat, Attack, Dataflow Diagram (DFD), Threat Tree (Attack Tree), STRIDE, DREAD. Security Architecture, Software Attack Surface, Secure, Mandatory Access Control (MAC), Discretionary Access Control (DAC), Role-based Access Control (RBAC), Access Matrix

**Secure Coding and Security Testing**: Introduction to Vulnerabilities, Vulnerability Patterns, Secure Coding Practices, Code Checking, Tools, Cross Site Scripting, Injection Flaws, Cross Site Request Forgery, Denial of Service, Test Cases, Security Test Plan, White Box Test, Black Box Test, Penetration Testing, Code Review, Test Report

**Secure Deployment**: Secure Default Configuration, Product Life Cycle, Automated Deployment Process, Secure Target Environment, Secure Delivery of Code, Trusted Origin, Code Signing, Least Privilege Permissions, ITIL Release and Deployment Management

**Security Response:** Security Response, Security Bulletins, Vulnerabilities, Security Patches, Disclosure, Responsible Disclosure, Patch Tuesday, Security Response Policy, Security Response Process, Common Vulnerability Scoring System, CVSS

**Code & Resource Protection**: Introduction to Back Door, Time Bomb, Four-Eyes Principle, Confidentiality Classification, Background Screening, Security Clearance, Offline and Online Licensing, Mechanisms, Code Obfuscation

**Recommended Books**

1. Julia H. Allen, Sean Barnum, Robert J. Ellison, Gary McGraw and Nancy Mead Software Security Engineering: A Guide for Project Managers by. Addison-Wesley, (2004)
2. Gary McGraw ,Software Security: Building Security, Addison-Wesley (2006)
3. Threat Modelling: Designing for Security by Adam Shostack, John Wiley and Sons Inc,

(2014).
4. Mano Paul ,7 Qualities of Highly secure Software Taylor and Francis, CRC Press (2012)

**Course Learning Outcomes (CLOs)**

| | |
|---|---|
| CLO1 | Analyze issues related secure software development methodologies |
| CLO2 | Apply a thorough understanding of secure coding principles |
| CLO3 | Select the most appropriate approach to secure software development |
| CLO4 | Judge and craft appropriate adaptations to the development process to make sure a secure deployment |
| CLO5 | Evaluate the implications and impact of secure architecture design |

**Evaluation Scheme:**

| S.No. | Evaluation Elements | Weightage (%) |
|---|---|---|
| 1. | MST | 20 |
| 2. | EST | 40 |
| 3. | Sessionals (May include Assignments/Projects/Tutorials/Quizzes/Lab Evaluations) | 40 |

| PSE391: SEMINAR | | | | |
|---|---|---|---|---|
| | **L** | **T** | **P** | **Cr** |
| | **0** | **0** | **0** | **4** |
| **Course Objectives**: This course is designed to help the student obtain skills to discuss or present something within a group. Seminar Course is an outcome of six months of study, exploration, survey and analysis of a particular topic. It is designed to test the skills of the candidate in making a good presentation, Audience Engagement, Communication Skills. It also helps in building lifelong learning as a skill in the candidate. | | | | |

## Course Learning Outcomes (CLOs)

| | |
|---|---|
| CLO1 | Identification of a domain specific scholarly topic |
| CLO2 | Investigate and tabulate details and history about the selected topic |
| CLO3 | Application of the selected topic in domain or real life |
| CLO4 | Technical report writing |
| CLO5 | Demonstrating the communication skills by good presentation and engaging the audience. |

**Evaluation Scheme**:

- Presenting a topic to an audience in a given time with a professionally prepared content.
- Literature Survey/Content: This includes the depth knowledge of the related work done by others related to Seminar Topic
- Viva(answer to the queries)
- Report Writing
- Reflective Diary
- Poster Presentation
- Video Presentation
- Peer Review

Final Seminar Assessment : 70 Marks
Continuous Assessment : 30 Marks

| PSE392: CAPSTONE PROJECT | | | | |
|---|---|---|---|---|
| | **L** | **T** | **P** | **Cr** |
| | **0** | **0** | **10** | **12** |
| **Course Objectives**: This course is designed to encourage experiential and wholesome projects where students take what they've learned throughout the program and apply it to examine a specific idea. It aims to provide the students an exposure to gain proficiency in modelling, implementing and testing nontrivial software applications. It must include a design component, User interface and usefulness for the society or the profession. | | | | |

## Course Learning Outcomes (CLOs)

| | |
|---|---|
| CLO1 | Investigate and identify real world problems |
| CLO2 | Design, develop and implement a domain specific project |
| CLO3 | Application of different skills learned in the program |
| CLO4 | Technical report writing |
| CLO5 | Demonstrating and communicating the working and impact of the project |

**Evaluation Scheme:**

- Progress Evaluation :
  Every month, there will regular progress evaluation of the project based on various parameters like problem definition, design etc.
- Final Evaluation :

1.Project report
2.Presentation (may include demonstration)
3.Demonstration of the project
4. Viva (answers to the queries)
5. Reflective diary
6. Poster presentation
7. Video presentation
8. Peer review

Final Evaluation : 60 Marks
Continuous Evaluation: 40 Marks ( At least spread in two evaluations)

| PSE091: Dissertation | | | | |
|---|---|---|---|---|
| | **L** | **T** | **P** | **Cr** |
| | **0** | **0** | **16** | **16** |
| **Course Objectives:** This course is designed to help the student obtain research skills which includes a thorough survey of a particular domain, finding a research problem and presenting a methodology to resolve the problem; with adequate experimental results to strengthen the contribution. The students are also given an exposure where they learn to write research papers and presenting the work in the conferences. Students are also supposed to learn about communicating the impact of their work by different tools which includes video, poster and presentation. | | | | |

## Course Learning Outcomes (CLOs)

| | |
|---|---|
| CLO1 | Identification, formulation and analysis of domain specific scholarly research problems |
| CLO2 | Design and implementation of identified research problem. |
| CLO3 | Technical report writing and Publication of research work in referred journals, National and international conferences of Repute |
| CLO4 | Ability to foresee how their current and future work will influence/impact the economy, society and the environment. |
| CLO5 | Ability to communicate and present the work to the relevant audience |

**Evaluation Scheme:**

- Subject matter of Presentation
- Literature Review
- Discussion of Results and Inferences drawn
- Presentation Structuring
- Response to Questions
- Usefulness/Contribution to the profession
- Overall Perception
- Reflective Diary
- Publication
- Poster
- Video Presentation