Structure of Post Graduate

(ME Software Engineering)

THAPAR INSTITUTE OF ENGINEERING & TECHNIOLOGY
(DEEMED TO BE UNIVERSITY)

PATIALA, PUNJAB, INDIA

COURSE SCHEME & SYLLABUS

(2020)

M.E. (SOFTWARE ENGINEERING)

ME- SOFTWARE ENGINEERING (2021)

SEMESTER I

S. NO.	CODE	TITLE	L	T	P	Cr
1	PCS112	Applied Statistical Methods	3	0	2	4
2	PCS108	Advanced Data Structures	3	0	2	4
3	PCS111	Software Project Management	3	0	2	4
4	PSE105	Software Design and Construction	3	0	2	4
5	PSE106	Software Engineering Concepts and Methodologies	3	0	2	4
6	PHU004	Research Methodology, Ethics and IPR	2	0	0	2
		TOTAL	**17**	**0**	**10**	**22**

SEMESTER II

1	PMA112	Linear Algebra and Random Processes	3	0	0	3
2	PCS106	Machine Learning	3	0	2	4
3	PSE202	Software Verification And Validation Testing	3	0	2	4
4	PSE205	Software Metrics And Quality Management	3	0	2	4
5		ELECTIVE-I	3	0	2	4
6		ELECTIVE-II	3	0	2	4
		TOTAL	**15**	**0**	**08**	**23**

ELECTIVE I

1	PCS221	Cloud Infrastructure and Services	3	0	2	4
2	PCS224	Natural Language Processing	3	0	2	4
3	PCS205	Computer and Network Security	3	0	2	4
4	PCS215	Continuous Delivery and DevOps	3	0	2	4

ELECTIVE II

1	PSE206	Agile Software Development Approaches	3	0	2	4
2	PSE207	Component Based Development	3	0	2	4
3	PSE208	Service Oriented Architecture	3	0	2	4
4	PSE209	Secure Software Development and Architecture Design	3	0	2	4

SEMESTER III

1		DISSERTATION/INTERNSHIP Interim Report	-	-	-	4
2	PCS391	Seminar	-	-	-	4
		TOTAL				**8**

SEMESTER IV

1	PCS392	PROJECT SEMESTER / DISSERTATION	-	-	-	16
		TOTAL	**-**	**-**	**-**	**16**
		GRAND TOTAL - FOUR SEMESTER CREDITS				**69**

PCS112 Applied Statistical Methods

L	T	P	Cr
3	0	2	4

Course Objective: The course aims to introduce to the students, fundamental principles as well as advanced topics in statistics and sampling techniques. This course underscores the importance of statistical methods to perform scientific and engineering research.

Review of basic probability and statistical principles: Axioms of probability, conditional probability, Bayes' rule, Conditional probability distributions, conditional expectations, law of total probability and law of total expectation, introduction to Bernoulli, binomial, Poisson, geometric, Normal, exponential, distributions, joint and marginal distributions, central limit theorem, probability distribution of functions of random variables.

Hypothesis tests: Introduction to sampling distributions (standard Normal, chi-square, F and t distributions) and their properties, introduction to hypothesis tests (difference between one tailed and two tailed tests), level of significance of test and power of test, two sample test for means using t-distribution.

Analysis of variance: One Way ANOVA, two-way ANOVA with examples.

Time Series Analysis: Autoregressive models: AR(1), AR(p), moving average models: MA(1), MA(q), autoregressive moving average models: ARMA(p,q).

Multivariate Data Analysis and regression: Introduction to linear regression with trends and least squares estimate, definition of Covariance matrix and its application in engineering problems using Principal Component Analysis.

Markov Chains: Introduction to discrete Markov chains in finite state space, multi-step state transition probabilities, stationary (limiting distributions), Chapman-Kolmogorov equations, hitting probabilities, return and exit time distributions for discrete Markov chains, classification of states, detailed balance.

Laboratory Work: Each laboratory experiment will consist of numerical exercises on one of the above topics. Laboratory experiments will be performed using Matlab/SPSS.

Course Learning Outcomes (CLO): Upon the completion of this course, the students will able to:

1. compute probabilities of composite events along with an understanding of random variables and distributions.
2. obtain foundational understanding of discrete Markov processes.
3. make statistical inferences using principles of hypothesis tests and ANOVA.
4. perform analysis of time series data with different time series models.
5. perform multivariate data analysis using Principal Component Analysis and linear regression.

Recommended Books:

1. Medhi, J., Stochastic Processes, New Age International (2005)
2. Paul L. Meyer, Introductory probability and statistical applications, Addison-Wesley Publishing Company, 1970
3. Durrett, R., Essentials of Stochastic Processes, Springer (2016).
4. Ross, Sheldon, Stochastic Processes, John Wiley and Sons (1996).
5. Hogg, R., McKean, J. and Craig, A. Introduction to Mathematical Statistics, Pearson (2013).
6. Hamilton, James, Time Series Analysis, Princeton University Press (2012).

PCS108 ADVANCED DATA STRUCTURES

L	T	P	Cr
3	0	2	4

Course Objective: To learn the advanced concepts of data structure and their implementation. The course has the main ingredients required for a computer science graduate and has all the necessary topics for assessment of data structures.

Introduction: Overview of various linear and non-linear data structures.

Complexity Analysis: Introduction to asymptotic complexity, Complexity of recursive algorithms, Amortized complexity, Complexity analysis of various sorting and searching techniques, Sorting in linear time.

Tree Structures: AVL Trees, Red-Black Trees, Splay Trees, B-trees, B+ Trees, Fibonacci heaps, Data Structures for Disjoint Sets, Augmented Data Structures, Self-Adjusting Data Structures, Temporal data structures, Succinct data structures, Dictionaries and cuckoo hashing.

Data Structures for Graphs and Related Algorithms: Representation, Type of graphs, Paths and circuits:, Euler graphs, Hamiltonian paths and circuits, Cut-sets, Connectivity and separability, Planar graphs, Isomorphism, Graph colouring, Covering and partitioning, Depth- and breadth-first traversals, Minimum spanning tree: Primôs and Kruskalôs algorithms, Shortest-path Algorithms: Dijkstraôs and Floydôs algorithm, Topological sort, Max flow: Ford-Fulkerson algorithm, Max flow-min cut.

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

Recommended Books:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, The MIT Press.
2. Y. Langsam, M. J. Augenstein, and A. O. Tenenbaum, Data Structures Using C and C++, Pearson Education India.
3. Peter Brass, Advanced Data Structures, Cambridge University Press.
4. J. Kleinberg and E. Tardos, Algorithm Design, Pearson Education India.
5. E. Horowitz, S Sahni, & S. Rajasekaran, Computer Algorithms, Computer Science Press.

Course Learning Outcomes (CLOs): After the completion of this course the student will be able to:

CLO1	Analyse the algorithms associated with advanced data structures.
CLO2	Implement basic data structures and analyse them to solve fundamental problems.
CLO3	Implement different tree data structures and differentiate themwith respect to their applications.
CLO4	Identify properties of graphs and employ them to model a variety of real-world problems.

PCS111 SOFTWARE PROJECT MANAGEMENT

L	T	P	Cr
3	0	2	4.0

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 leveling, 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.

Recommended Books

1. Software Project Management, Bob Hughes and Mike Cotterell, Tata McGraw Hill 5th edition, 2009
2. A practitionerôs Guide to Software Engineering, Roger Pressman, Tata McGraw Hill 2014 8thedition
3. Head First PMP: A Brain Friendly Guide To Passing The Project Management Professional Exam,2013

PSE106 SOFTWARE ENGINEERING CONCEPTS AND METHODOLOGIES

L	T	P	Cr
3	0	2	4.0

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 Yourdonôs 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

PSE 105 SOFTWARE DESIGN AND CONSTRUCTION

L	T	P	Cr
3	0	2	4

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.

PHU004: RESEARCH METHODOLOGY, ETHICS AND IPR

L	T	P	Cr
2	0	0	2

Course Objectives:

The course aims to:

- equip the students to analyse research related information.
- sensitize the students to ethical research practices.
- equip them to write technical reports and research paper.
- equip them with the process of patent filing
- create awareness about the consequences of IPR Infringement

Unit 1: Meaning of Research Problem, Sources of Research Problem, Criteria and Characteristics of good Research Problem, Errors in selecting a research Problem, scope and objectives of research problem.

Unit 2:Effective Literature studies, approaches and analysis.

Unit 3: Effective Technical Writing, How to write report and Research paper; developing a research proposal.

Unit 4: Non Parametric Tests: When to use a Nonparametric Tests**;** Mann Whitney U Test; Sign Test; Wilcoxon Signed Rank Test andKruskal-Wallis Test.

Unit 5: Ethics: Need for Ethics in Professional Life; Kohlbergôs Theory of Moral Development and Its Applicability to Engineers.Professional Ethics: Values in Work Life; Professional Ethics and Ethos; Codes of Conduct.

Research Ethics, Plagiarism, Case Studies on Ethics.

Unit 6: Introduction to IPR: Nature of Intellectual Property Rights: Patents; Designs; Trademarks; Copyright; Trade Secrets; Industrial Design; Geographical Indicators; Integrated Circuits. International Character of IPRs, Role of IPRs in Economic Development.

Patents: Introduction to Patents, Inventions not Patentable, Procedure for grant of Patents,Rights and Obligations of a Patentee; IPR Infringement.

Case studies on IPRs.

Course Outcome

At the end of this course the student will be able to:

- analyse research related Information.
- Indulge in ethical research practices
- equipped to write technical reports and research paper.
- Equipped with the process of patent filing
- possess awareness about consequences of IPR Infringement

Text Books:

1. Geoffrey R. Marczyk. Essentials of Research Design and Methodology, Wiley; 2008.
2. Wayne Goddard, Stuart Melville. Research methodology: An Introduction, Juta, 2004.
3. Thomas, C. George. Research Methodology & Scientific Writing, Ane Books Pvt. Ltd, 2016.
4. Menell, Peter S, Lemley, Mark A, Merges, Robert P. Intellectual Property in the New Technological Age, Vol. I Aspen Law & Business, 2019.
5. Menell, Peter S, Lemley, Mark A, Merges, Robert P. Intellectual Property in the New Technological Age, Vol. II Aspen Law & Business, 2019.
6. Narayanan, P., Intellectual Property Law, Eastern Law House, 2008.

PMA112 Linear Algebra and Random Process

L	T	P	Cr
3	0	0	3.0

Course Objective: The course aims to shape the attitudes of learners regarding the field of linear algebra and random process. Specifically, the course aims to (i) develop maturity in linear algebraic structure that appear in various areas of computer science (ii) motivate students towards an intrinsic interest in statistical thinking (iii) instil the belief that statistics is important for scientific research.

Linear Algebra

Matrices: Matrix multiplication, Transposes, Inverses, Gaussian elimination, factorization $A = L U$, rank of matrix.

Vector Spaces: Column ad row spaces, Solving $AX = 0$ and $AX = B$, Linear Independence/Dependence, Basis, Dimension and Linear Transformation.

Orthogonality: Orthogonal Vectors and subspaces, projection, and least squares, Gram ï Schmidt orthogonalization.

Determinants: Determinant formula, cofactors, inverses and volume.

Eigenvalues and Eigenvectors: Characteristic polynomial, Diogonalization, Hermitian and Unitary matrices, Spectral theorem, Change of basis.

Positive definite matrices and Singular Value Decomposition, Applications to Optimization problems and Graph Theory.

Random Processes

Basic topics: Event, Probability, Conditional probability, Independence, Product spaces

Random Variables: Distributions, Laws of average, discrete and continuous random variables, random vectors, Monte Carlo simulation.

Discrete Random Variables: Probability mass function, Independence, Expectation, Sums of random variables.

Continuous Random Variables: Probability density function, Independence, Expectation, Conditional expectations, Functions of random variables, Sums of random variables, Multivariate normal distributions.

Recommended Books:

1. Gilbert Strang, Linear algebra and Its Applications, Cengage Learning, Fourth edition, 2006.
2. Kenneth Hoffman and Ray Kunze, Linear algebra, Prentice Hall of India, second edition, 2013
3. W. B. Davenport, Probability and Random Process- an introduction for application scientists and engineers, McGraw Hill, 1970
4. Johnson, R., Miller, I. and Freudôs, J., Miller and Freundôs Probability and Statistics for Engineers, Pearson Education (2005) 7th Ed.
5. Walpole, Ronald E., Myers, Raymond H., Myers, Sharon L. and, Keying Ye, Probability and Statistics for Engineers and Scientists, Pearson Education (2007) 8thed.

Course Learning Outcomes (CLO): Upon successful completion of the course, the students will be able to

1. Identify and comprehend linear algebraic structures that appear in computer science.
2. Use linear algebraic methods to perform computational task.
3. Apply properties of eigenvalues and orthogonality to analyse computational problems occurring in various areas of computer science.
4. Understand and apply various concepts of probability theory.
5. Comprehend and apply the properties of random processes in real life problems.

PCS 106 MACHINE LEARNING

L	T	P	Cr
3	0	2	4.0

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.

Introduction: Basic concepts, Designing a learning system, Issues in machine learning. Types of machine learning: Learning associations, Supervised learning, Unsupervised learning, Reinforcement learning, Model Evaluation Parameters, Feature Selection and Extraction. Linear Regression, Multi Regression, Gradient Descent, SVM.

Clustering: K-Means, k-Medoids, Agglomerative versus Divisive Hierarchical Clustering Distance Measures in Algorithmic Methods, Mean-shift Clustering.

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.

Genetic Algorithms: Basic concepts, Hypothesis space search, Genetic programming, Models of evolution and learning, Parallelizing Genetic Algorithms.

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

1. Mitchell T.M., Machine Learning, McGraw Hill (1997) 2nd ed.
2. Alpaydin E., Introduction to Machine Learning, MIT Press (2010) 2nd ed.
3. Bishop C., Pattern Recognition and Machine Learning, Springer-Verlag (2006) 2nd ed.
4. Michie D., Spiegelhalter D. J., Taylor C. C., Machine Learning, Neural and Statistical Classification. Overseas Press (2009) 1st ed.

COURSE LEARNING OUTCOMES (CLOs)

After the completion of this course the student will be able to:

CLO1	Understand basic principles, techniques, and applications of machine learning modules.
CLO2	Understand and use regression techniques for real time applications
CLO3	Apply decision tree learning, Bayesian learning, Artificial Neural Network and SVM in real world problems.
CLO4	Critically evaluate and compare different learning models and algorithms and be able to adapt or combine some of the key elements of existing machine learning algorithms to design new algorithms
CLO5	Understand the use of genetic algorithms and genetic programming.

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.

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.

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.
Program Mutation Testing: Introduction, Mutation and mutants, Mutation operators, Equivalent mutants, Fault detection using mutants, Types of mutants, Mutation operators for C and Java.

Laboratory Work: To Use various verification and validation testing tools and to apply these tools on few examples and case studies

Recommended Books

1. Marcus S. Fisher, Software Verification and Validation: An Engineering and Scientific Approach, Springer(2007)
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.

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

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). 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 ï ISO, CMM
CLO5	Control and Manage the project and processes, apply configuration management on the basis of collected metrics.

PCS221 CLOUD INFRASTRUCTURE AND SERVICES

L	T	P	Cr
3	0	2	4

Course Objective: To learn the advanced concepts of cloud infrastructure and services and its implementation for assessment of understanding the course by the students.

Introduction: Cloud Computing, History and evolution, Overview of Types of Computing: Cluster, Grid, Utility and Autonomic Computing, Applications of cloud computing for various industries, economics and benefits of cloud computing.

Cloud Computing Architecture: Cloud Architecture, Types of Clouds: Public, Private & Hybrid Clouds, Cloud based services: Iaas, PaaS and SaaS.

Cloud Computing Issues and Challenges: Security, Elasticity, Resource management and scheduling, QoS (Quality of Service) and Resource Allocation, Cost Management, Big Data.

Data Center and Warehousing : Classic Data Center, Warehousing, Virtualized Data Center (Compute, Storage, Networking and Application), Design Principles.

Cloud Implementations and Environments: Amazon Web Services, The Elastic Compute Cloud (EC2), The Simple Storage Service (S3), The Simple Queuing Services (SQS), Google AppEngine - PaaS, Windows Azure, Aneka, A Comparison of Cloud Computing Platforms.

Virtualization: Virtualization, Advantages and Disadvantages, Types of Virtualization: Resource Virtualization i.e. Server, Storage and Network virtualization, Migration of processes, VMware cloud ï IaaS.

Cloud based Data Storage: Introduction to Hadoop and Map Reduce for Simplified data processing on Large clusters, Distributed File system, Data Replication, Shared access to data stores, introduction to Python, Design of data applications based on Map Reduce, Task Partitioning, Data partitioning, Data Synchronization.

Laboratory Work: To implement Cloud, Apache and Hadoop framework and related services. To understand various concepts practically about virtualization, data storage. To implement few algorithms with the help of MapReduce and some high level language.

Recommended Books:

1. *Raj Kumar Buyya, James Broberg, Andrezei M.Goscinski, Cloud Computing: Principles and paradigms (2011)*
2. *Michael Miller, Cloud Computing, Que Publishing (2008).*
3. *Cloud Computing: A practical Approach Anthony Velte, Toby Velte and RobertElsenpeter by Tata McGrawHill*
4. *Judith Hurwitz, Robin Bllor, Marcia Kaufman, Fern Halper, Cloud Computing for dummies (2009).*
5. *T. white, Hadoop: The Definitive Guide, O' Reilly Media (2012), 3rd ed.*

Course Learning Outcomes (CLOs)

CLO1	Understand the existing hosting platforms and computing paradigms currently being used in industry and academia.
CLO2	Comprehend data centre needs, its virtualization techniques and types of clouds.
CLO3	Apply virtualization in Amazon Web Services, Azure, Aneka etc.
CLO4	Learn to use cloud based data storage.
CLO5	Learn Hadoop file system and MapReduce Programing.

PCS224 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-Based Machine 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 Language Processing: The PLNLP Approach, Springer (2013).*
4. *Roach P., Phonetics, Oxford University Press (2012).*

Course Learning Outcomes (CLOs)

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

PCS205 COMPUTER AND NETWORK SECURITY

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.

PCS215 Continuous Delivery and DevOps

L	T	P	Cr
3	0	2	4.0

Course Objectives: This course makes student learn the fundamental principles and practices associated with DevOPs. To apply the principles and practices of DevOps and automation on a project of interest and relevance to the student.

Introduction to DevOps: History of DevOps, DevOps Ecosystem, DevOps Objectives, DevOps Market Trends, Infrastructure As A Code, IaaS Overview, Paas Overview, DevOps on the Cloud, DevOps Production Model, Tool pipelining

DevOps and Automation: Version Control, Continuous Integration, Continuous Testing, Configuration Management, Continuous Deployment, Containerization, Continuous Monitoring, Tool pipelining

Version Control: Introduction to version control, Introduction toGit, importance of Git for an organization, Common commands in Git, Working with Remote Repositories, Branching and Merging in Git, Git workflows, Git cheat sheet.

Continuous Integration: Introduction to Jenkins and its Architecture, Jenkins Management, Build Setup, Git and Jenkins Integration

Continuous Testing: Agile Testing Techniques, Test-Driven Development (TDD), Behaviour Driven Development (BDD), Acceptance Test Driven Development (ATDD) Life Cycle, User Acceptance Test, Definition of Done (DoD), fit test, early testing and traditional testing techniques, Introduction to Selenium,Selenium ï Webdriver, X-Path, Creating Test Cases in Selenium WebDriver (Waits), Handling different controls on Webpage

.

Containerization: Benefits and use cases for containerized environments, Shipping Transportation Challenges, Introduction to Docker, Understanding images and containers, Introduction to Container, Container Life Cycle, Sharing and Copying, Base Image, Docker File, Working with containers, Publishing Image on Docker Hub, Install Docker on a local machine, Define a container environment using a Dockerfile, Store and share a docker Deployment, Container Deployment, Container orchestration, Kubernetes and container clusters, Continuous Delivery (CD) and Continuous Integration (CI) with AWS CodePipeline and AWS CodeBuild

Self Learning Content: Linux Commands,Introduction to Cloud,IaaC, IaaS, PaaS and SaaS, AWS,Virtualisation, REST API, SQL, Introduction to SQLAlchemy and Postgresql,HTTP and Flask Basics, ELK, Enabling tools for DevOps: Software configuration tools, Orchestration tools and Automated QA tools, Chef, Puppet, Docker,Vagrant, and Selenium, Maven, Ansible, Nagios

Laboratory Work: Exploring and installing the DevOps enabling tools. Students will be given small project deploying a Flask-based web application to the cloud using Docker and Kubernetes

CLOs

1. Comprehend the basic concepts of DevOps and automation.
2. Apply version control using Git for remote repositories
3. Apply agile testing techniques
4. Continuous Delivery (CD) and Continuous Integration (CI) with AWS CodePipeline and AWS CodeBuild

Recommended Books

1. Sharma S., The DevOps Adoption Playbook: A Guide to Adopting DevOps in a Multi-Speed IT Enterprise Wiley; 1st Ed., 2017
2. Relan K, Building REST APIs with Flask: Create Python Web Services with MySQL, Apress,1st Ed., 2019

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)
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

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

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 XMLôs, 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

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.

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

PCS392 DISSERTATION

Cr 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	Design and implementation of identified research problem or industrial projects.
CLO2	Develop acumen for higher education and research.
CLO3	Write technical reports and publish the research work in referred journals, national and international conferences of repute.
CLO4	Foresee how their current and future work will influence/impact the economy, society and the environment.

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