M. E. Computer Engineering (FULL TIME) With Specialization in Software Engineering
Curriculum & Syllabus
Batch 2015– 2016 and Onwards

S. No.	Category	No. of Credits			
		SEM I	SEM II	SEM III	SEM IV
1.	Course Compulsory	15	15	-	-
2.	Generic Elective	4	4	-	-
3.	Programme Elective	5	5	-	-
4.	Skill development	2	2	-	-
5.	Seminar/ Workshop/ Research Tool	2	2	-	-
6.	Dissertation Phase	-	-	12	12
Actual Credits per Semester		**28**	**28**	**12**	**12**
Total actual Programme Credits					**80**
7.	Virtual Credited Comprehensive Viva	4	4	4	4
Total Credits per Semester		**32**	**32**	**16**	**16**
Total Programme Credits					**96**

M. E. Computer Engineering (FULL TIME) With Specialization in Software Engineering
Curriculum & Syllabus
Batch 2015– 2016 and Onwards

SEM I				
S.NO	**Sub Code**	**Sub Name**	**Number of Credit L-T-P**	**Sub Type**
1.	SER1C1	Advanced Algorithms	3-1-1	PC1
2.	SER1C2	Object Oriented Analysis & Design	3-1-1	PC2
3.	SER1C3	Software Construction	3-1-1	PC3
4.	SER1Gx	Generic Elective I	3-1-0	GE1
5.	SER1Ex	Elective I	3-1-1	PE1
6.	ASR1S1	Soft Skills -1	2-0-0	
7.	SER1W1	Seminar/ Workshop/Research Tool	0-2-0	
8.	SER1V1	Comprehensive Viva I	0-0-4	
Total Credit for SEM I			**28 actual + 4 Virtual credits**	
		List of Generic Elective I	**L-T-P**	
1.	SER1G1	Soft Computing	3-1-0	
2.	SER1G2	Distributed Operating System	3-1-0	
3.	SER1G3	Advance Computer Architecture	3-1-0	
		List of Elective I	L-T-P	
1.	SER1E1	Database Engineering	3-1-1	
2.	SER1E2	Big Data Analytics	3-1-1	
3.	SER1E3	Secure Software Engineering	3-1-1	
SEM II			**L-T-P**	
1.	SER2C1	Software Project Planning and Management	3-1-1	PC4
2.	SER2C2	Design Pattern	3-1-1	PC5
3.	SER2C3	Software Testing and Quality Assurance	3-1-1	PC6
4.	SER2Gx	Generic Elective II	3-1-0	GE2
5.	SER2Ex	Elective II	3-1-1	PE2
6.	ASR2S2	Soft Skills -2	2-0-0	
7.	SER2W2	Seminar/ Workshop/ Research Tool	0-2-0	
8.	SER2V2	Comprehensive Viva II	0-0-4	
Total Credit for SEM II			**28 actual + 4 Virtual credits**	
		List of Generic Elective II	**L-T-P**	
1.	SER2G1	Data Mining & Warehousing	3-1-0	
2.	SER2G2	Cloud Computing	3-1-0	
3.	SER2G3	Simulation and Modelling	3-1-0	
		List of Elective II	L-T-P	
1.	SER2E1	Speech And Language Processing	3-1-1	
2.	SER2E2	Aspect Oriented Software Engineering	3-1-1	
3.	SER2E3	Machine Learning	3-1-1	

M. E. Computer Engineering (FULL TIME) With Specialization in Software Engineering
Curriculum & Syllabus
Batch 2015– 2016 and Onwards

SEM III			**L-T-P**	
1	SER3D1	Dissertation Phase I	0-0-12	
2	SER3V3	Comprehensive Viva III	0-0-4	
Total Credit for SEM III			**12 actual + 4 Virtual credits**	
SEM IV			**L-T-P**	
1	SER4D2	Dissertation Phase II	0-0-12	
2	SER4V4	Comprehensive Viva IV	0-0-4	
Total Credit for SEM IV			**12 actual + 4 Virtual credits**	
Total Credit			**80 actual + 16 Virtual credits=96**	

Devi Ahilya University, Indore, India Institute of Engineering & Technology				I Year M.E. (Computer Engineering Sp. in Software Engineering) (Full Time)			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
SER1C1	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
Advanced Algorithms	**3**	**1**	**2**	**3**	**1**	**1**	**5**
Duration of Theory Paper: 3 Hours							

Course Objectives: To introduce students a variety of advanced techniques, methods and results from the rapidly-developing field of algorithms to solve problems. To familiarise the state of the art in some areas of algorithmic research, including open problems.

Prerequisites: Data Structures and Algorithms.

COURSE CONTENTS

UNIT - I

Review of basic concepts; Worst case and average case analysis, Asymptotic notation, Solving recurrence equations, Medians and order statistics, Advanced data structures: Binomial Heaps, Fibonacci Heaps, Data Structures for Disjoint Sets – Disjoint-set operations, Linked-list representation of disjoint sets, Disjoint-set forests, analysis of union by rank with path compression.

UNIT - II

Advanced Design and Analysis techniques: Greedy and Dynamic Programming strategies, Backtracking, Branch and Bound. Algorithms for Knapsack problems, Matrix-Chain Multiplication problem, Traveling Salesperson Problem (TSP), etc.
Amortized analysis: the aggregate method, the accounting method, the potential method, Dynamic tables.

UNIT - III

Graph algorithms: Breadth-first search, Depth-first search, Topological sorting, Minimum Spanning Trees, Single-Source Shortest Paths, All-Pairs Shortest Paths, Maximum Flows:Augmenting Paths and Push-Relabel Methods, Minimum Cost Flows, Bipartite Matching.

UNIT - IV

Introduction to string matching problem, String matching algorithms: Naive algorithm, Rabin-Karp, Knuth-Morris-Pratt, Boyer-Moore, etc. Applications in Bioinformatics.
Computational Geometry:Convex Hull. Line-segment Intersection. Sweep Lines. Voronoi Diagrams. Range Trees. Seidel's Low-dimensional LP Algorithm.

UNIT - V

Theory of NP-Hard and NP-Complete Problems: P, NP and NP-Complete complexity classes; A few NP-Completeness proofs; other complexity classes.

Dealing with intractability: Introduction, Combinatorial Optimization, approximation factor, PTAS, FPTAS, Approximation algorithms for vertex cover, set cover, TSP, knapsack, bin packing, subset-sum problem etc. Analysis of the expected time complexity of the algorithms.

Learning Outcomes:

Upon Completing the Course, Student will have:

1. Skills to analyze algorithms
2. Comparative judgments of different design techniques
3. Ability to solve real world problems
4. Idea about the hardness of some well-known problems including TSP, vertex cover, network flow and combinatorial optimization problems.
5. Familiarity with active research areas in connection with the study of algorithms.

BOOKS RECOMMENDED:

1. T. Cormen, C. Leiserson, R. Rivest, and C. Stein. **Introduction to Algorithms**. (3rd Ed). MIT Press, McGraw-Hill, 2010.
2. M.T. Goodrich, R. Tamassia, “Algorithm design – Foundations, Analysis, and Internet Examples”, John Wiley, Second Edition.
3. V. V. Vazirani, **Approximation Algorithms**, Springer. 2001.
4. Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin, **Network Flows: Theory, Algorithms, and Applications**,
5. E Horrowitz, S salmi, S Rajasekaran, “Fundamentals of Computer Algorithms”, Second Edition, University Press, 2007.
6. Aho, A V Hopcraft Ullman JD, “The Design and analysis of computer Algorithms”, Pearson Education, 2007.

LIST OF PRACTICAL ASSIGNMENTS:

Practical assignments will be based on:

1. Performance analysis
2. Solving problems using design techniques discussed
3. Solution of network flow problems
4. Approximation algorithms
5. String matching algorithms
6. Combinatorial optimization problems
7. TSP problem
8. solving some real world problems using the skill gained in the course
9. study of NP-Complete, NP-Hard problems
10. Any other emerging/ active research problems in the area in consultation with the instructor.

Devi Ahilya University, Indore, India Institute of Engineering & Technology				I Year M.E. (Computer Engineering Sp. in Software Engineering) (Full Time)			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
SE1C2 Object Oriented Analysis and Design	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
	3	**1**	**2**	**3**	**1**	**1**	**5**
Duration of Theory Paper: 3 Hours							

Learning Objectives:

This course offers the opportunity to students to become skilled at the object oriented analysis and design. This is essential as it is the core of the software development process. This course shall help the student to comprehend the principlesof object orientation and apply them as the solution for the real life problems in the form of object oriented design. At the end of the course the students shall be able to design a solution which works and solves software development problems.

Pre requisites:

1. Programming knowledge in any of the object oriented languages like C++, Java.
2. Familiarity and ease with data structures.

Unit I: Introduction to Modelling and UML 2.X

Importance of Modelling, Principles of Modelling, Object Oriented Modelling, Conceptual model of the UML, Architecture, Software Development Life Cycle

Unit 2: Basic and Advanced Structural Modelling

Basic Structural Modelling: Classes, Relationships, Common Mechanisms and Diagrams.
Advanced Structural Modelling: Advanced Classes, Advanced Relationships, Interfaces, Types and Roles, Packages

Unit 3: Class & Object Diagrams

Class & Object Diagrams: Terms, Concepts, Modelling Techniques for Class and Object Diagrams

Unit 4: Basic Behavioural Modelling

Interactions, Interaction diagrams, Use cases, Use case Diagrams, Activity Diagrams, Events and Signals, State Machines, Processes and Threads, Time and Space Diagram, State Chart Diagrams

Unit 5: Architectural Modelling

Architectural Modelling: Component, Deployment, Component Diagrams and Deployment Diagrams, Case Study, Issues in OO Testing

TEXT BOOKS:

1. Grady Booch, James Rumbaugh, IvarJacobson : The Unified Modeling Language User Guide, Pearson Education.

2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, WILEY-Dreamtech India Pvt. Ltd.

REFERENCE BOOKS:

1.Meilir Page-Jones: Fundamentals of Object Oriented Design in UML, Pearson Education.
2. Pascal Roques: Modelling Software Systems Using UML2, WILEY- Dreamtech India Pvt. Ltd.
3.AtulKahate Object Oriented Analysis & Design, The McGraw-Hill Companies.
4. Object-Oriented Analysis and Design with the Unified Process By John W. Satzinger, Robert B Jackson and Stephen D Burd, Cengage Learning.

Learning Outcomes:

The aim of the course is to help the student be able to understand the real world problems. The student shall be able to solve the complexity of the problem and also depict the problem with the help of standard UML diagrams. In the process of software design the student shall be able to appreciate the application of diagrams, iterative approach which helps in improving the software quality.

Devi Ahilya University, Indore, India Institute of Engineering & Technology				I Year M.E. (Computer Engineering Sp. in Software Engineering) (Full Time)			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
SER1C3	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
Software Construction	**3**	**1**	**2**	**3**	**1**	**1**	**5**
Duration of Theory Paper: 3 Hours							

Objective: To gain the knowledge for advance programming using JAVA

UNIT I

Object Oriented Concepts, Merits of Object Oriented Technology. Introduction to JAVA and its applications on the Internet. Constructors & constructor overloading, Access modifiers: Class attributes and methods. Introduction to object model of software development.

UNIT II

Introduction to Java classes and objects, Java features, data types data type conversions & control statements Operators and their precedence. Introduction to Class, Instance members and member functions.

UNIT III

String Handling, Wrapper classes Arrays and Vectors Inheritance and Polymorphism

Class relationships Inheritance and its types Merits and Demerits of Inheritance Introduction to Association, inheritance Polymorphism Dynamic method dispatch Runtime polymorphism Abstract classes, Interfaces packages

UNIT IV

Java I/O, Basic concept I/O stream reader-writer Exceptions: Need for exceptions Checked exceptions Unchecked exceptions Creating exceptions Multithreading Introduction, Priorities and scheduling of Threads Thread Synchronization and its life cycle Applet and its Life Cycle

UNIT V

Server side programming:- Java servlets, Java Server pages, Web application development using Java, Database handling with Java.

Text Books:

T1: Herbert Schildt, Java: The Complete Reference, 8th Edition

T2: Programming with java by e balagurusamy

Devi Ahilya University, Indore, India Institute of Engineering & Technology				I Year M.E. (Computer Engineering Sp. in Software Engineering) (Full Time)			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
SER1G1	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
Soft Computing	**3**	**1**	**0**	**3**	**1**	**0**	**4**
Duration of Theory Paper: 3 Hours							

Learning Objectives:

1. To familiarize with neural networks and learning methods for neural networks.
2. To introduce basics of genetic algorithms and their applications in optimization and planning.
3. To introduce the ideas of fuzzy sets, fuzzy logic and fuzzy inference system.
4. To develop skills thorough understanding of the theoretical and practical aspects of Soft Computing.

Pre requisites: Analysis of Algorithm, Artificial Intelligence.

COURSE CONTENTS

UNIT-I

INTRODUCTION TO SOFT COMPUTING AND NEURAL NETWORKS

Evolution of Computing ,Introduction of soft computing, soft computing vs. hard computing, various types of soft computing techniques, applications of soft computing, Machine Learning Basics and Fundamentals of Neural Networks and Application.

UNIT-II

NEURAL NETWORKS

Backpropagation Networks, Architecture: perceptron model, single layer artificial neural network, multilayer perception model; backpropagation learning methods, effect of learning rule coefficient, backpropagation algorithm, factors affecting backpropagation training, Associative memory, Adaptive Resonance Theory.

UNIT-III

GENETIC ALGORITHMS

Genetic Algorithm(GA) Basic concepts, working principle, procedures of GA, flow chart of GA, Genetic representations, (encoding) Initialization and selection, Genetic operators, Mutation, Generational Cycle, probability of crossover and probability of mutation, convergence. The Scheme Theorem – Classification of Genetic Algorithm – Holland Classifier Systems. Simulated annealing

and stochastic models, Boltzmann Machine, Applications of Genetic Algorithm: genetic algorithms in search and optimization, GA based clustering Algorithm, Image processing and pattern Recognition.

UNIT-IV

FUZZY LOGIC

Fuzzy Sets, Operations on Fuzzy Sets, Fuzzy Relations, Membership Functions, Fuzzy Rules and Fuzzy Reasoning, Fuzzy Inference Systems, Fuzzyfications&Defuzzificataions, Fuzzy Controller, Industrial applications Fuzzy Expert Systems, Fuzzy Decision Making.

UNIT-V

NEURO, FUZZY MODELING

Adaptive Neuro, Fuzzy Inference Systems Coactive Neuro, Fuzzy Modeling, Classification and Regression Trees Data Clustering Algorithms, Rulebase Structure, Identification , Neuro Fu zzy Control , Case studies.

Learning Outcomes:

Upon Completing the Course, Student will able to:

1. Identify and describe soft computing techniques and their roles in building intelligent machines.
2. Apply neural networks to pattern classification and regression problems
3. Recognize the feasibility of applying a soft computing methodology for a particular problem
4. Apply fuzzy logic and reasoning to handle uncertainty and solve engineering problems.
5. Apply genetic algorithms to combinatorial optimization problems.

BOOKS RECOMMENDED:

[1] S. Rajasekaran and G.A.VijaylakshmiPai.. Neural Networks Fuzzy Logic, and Genetic Algorithms, Prentice Hall of India.

[2] Timothy J. Ross, “Fuzzy Logic with Engineering Applications” Wiley India.

[3] Neural Networks and Fuzzy Systems: Dynamical Systems Application to Machine Intelligence - Bart Kosko, Prentice Hall, 1992.

[4] Mitchell Melanie, “An Introduction to Genetic Algorithm”, Prentice Hall, 1998.

[5] David E. Goldberg, “Genetic Algorithms in search, Optimization & Machine Learning” ,Addison-Wesley, 1997.

Devi Ahilya University, Indore, India Institute of Engineering & Technology				I Year M.E. (Computer Engineering Sp. in Software Engineering) (Full Time)			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
SER1G3	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
ADVANCE COMPUTER ARCHITECTURE	**3**	**1**	**0**	**3**	**1**	**0**	**4**
Duration of Theory Paper: 3 Hours							

Learning Objectives:

1. To familiarize with current trends in high performance computing.
2. To introduce quantitative analysis of computer architectures

Pre requisites: Computer organization

Unit 1 : Introduction to Computer Architeture& Quantitative Analysis

Generations of Computers, Definition of Computer Architecture, General trends in technology, power and cost. Measuring, Reporting, and Summarizing performance. Quantitative principles of computer design.

Unit 2 : Pipelining & Instruction level Parallelism

Need of pipelining, Pipeline Hazards, Implementation issues, Overcoming Pipeline hazards, pipeline extension to support multicycle operations. Concepts and challenges in ILP, Compiler techniques for ILP, Dynamic Scheduling, Hardware based Speculation, Combining dynamic scheduling, multiple issue and speculation.

Unit 3 : Memory Hierarchy Design

Introduction, Optimizations for improving Cache performance, Memory technology and optimizations, Virtual Memory protection and performance issues, Virtual machines protection.

Unit 4 : Data level Parallelism and Thread level Parallelism

Introduction, Vector architecture, Graphics Processing Units, Detecting and enhancing loop-level parallelism, Centralized shared memory architecture, Shared Memory multiprocessors performance, distributed shared memory and directory-based coherence, models of memory consistency.

Unit 5 : Interconnection Networks

Introduction, Connecting two or more devices, network topology, network routing, arbitration, and switching, Examples of interconnection networks, internetworking issues.

Case Study: MIPS Processor

Books Recommended

- J. Hennessy & D. Patterson, Computer Architecture : A Quantitative Approach, Morgan Kaufmann Series, 5th Edition, 2011.
- Kai Hwang, Advance Computer Architecture: Parallelism, Scalability, Programmability, Mcgraw Hill Computer Science Series, 1992.
- D. Sima& T. Fountain & P. Kacsuk, Advance Computer Architectures : A Design Space Approach, 1st Edition, Pearson Education, 2002.
- J. Hayes, Computer Architecture and Organization, Mcgraw Hill Education Series (India), 2012.

Devi Ahilya University, Indore, India Institute of Engineering & Technology				I Year M.E. (Computer Engineering Sp. in Software Engineering) (Full Time)			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
SER1E1	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
DATBASE ENGINEERING	**3**	**1**	**2**	**3**	**1**	**1**	**5**
Duration of Theory Paper: 3 Hours							

Learning Objectives:

- To understand how transactions are executed and concurrency mechanisms are used in practice.
- To understand how DBMS process queries and how it estimates the cost of query optimization.
- To understand how DBMS maintains data records and access paths.
- To understand the need and use of distributed database systems in practice.
- To familiarize with the emerging technologies of databases.

Prerequisites: Introductory knowledge of Database Systems.

COURSE CONTENTS

Unit-I

Transaction Processing & Concurrency Control: Introduction to Transaction Processing, Transaction Properties, Transaction recoverability and serializability, Transaction Support in SQL, Introduction to Concurrency Control, Two-phase locking, Timestamp ordering, Validation and other issues.

Unit-II

Query Processing & Optimization: Introduction, Translating SQL queries, Algorithms – External Sorting, Select, Join and Project operations, Aggregate and Outer Joins, Heuristics for Query optimization, Estimating cost in query optimization, Semantic optimization, Optimization used in practice.

Unit-III

Data Storage and Querying: File organization, Organization of records, Indexing and Hashing – Basic concepts, B+-tree index files, Static and dynamic hashing, comparison of indexing and hashing etc..

Unit-IV

Distributed Databases : Concepts, Techniques for Distributed database design – Data fragmentation, replication, and allocation techniques; Types of Distributed Systems, Query processing in Distributed Databases, Concurrency control & Recovery in Distributed Databases, Distributed Databases in MySQL.

Unit-V

Advance Topics: Information Retrieval and XML data, Spatial data management, NoSQL – Diffrences from Relational Databases, Theory, Key-Value Databases, Graph Databases etc.

Books Recommended:

1. Fundamentals of Database Systems, By R. Elmasri and S. Navathe, 6th Ed. Pearson Education, 2010.
2. Database Management Systems, R. Ramkrishnan and J. Gehrke, 3rd Edition, McGraw Hill Education, 2014.
3. Database System Concepts, By A. Silberschatz, H. Korth and S. Sudarshan, 6th Ed. McGraw Hill Education, 2013.
4. Database Systems, By T. Connolly and C. Begg, 4th Edition, Pearson Education, 2008.

List of Assignments:

During the learning of course, students need to do assignments:

1. Solving intermediate SQL queries involving join expressions, views and transaction support.
2. Using PL/SQL constructs involving procedures, triggers, recursive queries etc.
5. Assignment on Query processing and indexing.
4. Using concurrency and transactions
6. Distributed database support in MySQL or PostgreSQL.

Devi Ahilya University, Indore, India Institute of Engineering & Technology				I Year M.E. (Computer Engineering Sp. in Software Engineering) (Full Time)			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
SER1E2	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
Big data analytics	**3**	**1**	**2**	**3**	**1**	**1**	**5**
Duration of Theory Paper: 3 Hours							

LEARNING OBJECTIVES:

1. To increase knowledge of the Big Data landscape.
2. Develop a comprehensive knowledge R and to use R for effective data analysis.
3. Develop skills in independent managing Big data projects and related issues.
4. Develop ability to carry out research in area of Big Data.

Pre requisites: Some programming experience (in any language) is recommended.

COURSE CONTENTS

UNIT-I

Introduction to Big Data : Overview of Big Data, Characteristics of Big Data, Sources of Big data, Five V's of Big Data, Examples of Big Data, Advantagesof Big Data, Big Data Applications, Strategies of Big Data, challenges Process of Data Analysis.

UNIT II

Introduction R : Overview and History of R, R Console Input and Evaluation, Data Types - R Objects and Attributes, Vectors and Lists, Matrices, Factors, Missing Values, Data Frames, Names Attribute, Reading Tabular Data, Reading Large Tables, Textual Data Formats, Connections: Interfaces to the Outside World.

Unit III

Programming with R: Subsetting – Basics, Lists, Matrices, Partial Matching, Removing Missing Values, Vectorized Operations. Control Structures - If-else, For loops, While loops, Repeat, Next,

Break, Functions, Scoping Rules - Symbol Binding, R Scoping Rules, Coding Standards, Dates and Times

Unit IV

Loop Functions and Debugging : Loop Functions – lapply, apply, mapply, tapply, split, Debugging Tools - Diagnosing the Problem, Basic Tools, Using the Tools

Unit V

Developing Data Products : Introduction to Data Products, Intro to rCharts and GoogleVis, rCharts introduction, rCharts examples, rCharts mapping, GoogleVis, plotly, Interactive graphics

LEARNING OUTCOMES:

Upon completing the course, students will be able to:

1. Apply Knowledge of Big Data to solve real world big data problems.
2. Understand the fundamentals of 'R' programming
3. Work on a real life Project, implementing R Analytics to create Business Insights.
4. Apply Data Visualization to create fancy plots
5. Undergo into further research in Big Data.

BOOKS RECOMMENDED:

1. Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Todays Businesses, Michael Minelli, Michele Chambers, AmbigaDhira, Wiley India Pvt Ltd, 2013.

2. R for Everyone: Advanced Analytics and Graphics, 1st Ed., Jared P. Lander, Pearson Education, Inc., 2014.

3.Big Data Analytics with R and Hadoop, Vignesh Prajapati, Packt Publishing Ltd, 2013.

4. Big Data Analytics: Turning Big Data into Big Money, Frank J. Ohlhorst, Wiley, 2012.

5. Creating Value with Big Data Analytics: Making Smarter Marketing Decisions, Peter C. Verhoef, Edwin Kooge, Natasha Walk, Taylor & Francis, 2016.

List of Practical/ Programming Assignments: (if applicable)

During the learning of course, students need to do assignments:

5. To learn the R Programming language.
6. To explore Rstudio for solving the Big data problems.

Devi Ahilya University, Indore, India Institute of Engineering & Technology				I Year M.E. (Computer Engineering Sp. in Software Engineering) (Full Time)			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
SER2C4	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
Software Project Planning and Management	**3**	**1**	**2**	**3**	**1**	**1**	**5**
Duration of Theory Paper: 3 Hours							

Learning Objectives:

This course provides the knowledge for correct software development life cycle, create realistic project plans, and manage a software development team through each phase of the project.

The purpose of software project planning and management is to forecast many hazards and risks and problems as possible. The student shall be to plan, organize and control activities so that the project is completed as successfully as possible in spite of all the risks.

Pre-requisites:

A course on software engineering and practical experience of handling college projects.

COURSE CONTENTS

UNIT-I: Introduction to Software Project Management

Project Definition, Contract Management, Activities Covered by Software ProjectManagement, Overview of Project Planning, Stepwise Project Planning

UNIT-II: Project Evaluation

Strategic Assessment, Technical Assessment, Cost Benefit Analysis, Cash FlowForecasting, Cost Benefit Evaluation Techniques, Risk Evaluation

UNIT-III: Activity Planning

Objectives, Project Schedule, Sequencing and Scheduling Activities, NetworkPlanning Models, Forward Pass, Backward Pass , Activity Float, Shortening ProjectDuration , Activity on Arrow Networks , Risk Management , Nature of Risk , Types ofRisk , Managing Risk , Hazard Identification , Hazard Analysis , Risk Planning andControl.

UNIT-IV: Monitoring and Control

Creating Framework, Collecting the Data, Visualizing Progress, Cost Monitoring, Earned Value, Prioritizing Monitoring, Getting the Project Back to Target, Change Control, Managing Contracts, Introduction, Types of Contract, Stages in Contract Placement, Typical Terms of a Contract, Contract Management , Acceptance**.**

UNIT-V: Managing People and Organizing Teams

Introduction, Understanding Behaviour, Organizational Behaviour: A Background, Selecting the Right Person for the Job, Instruction in the Best Methods, Motivation,The Oldman, Hackman Job Characteristics Model, Working In Groups, Becominga Team, Decision Making, Leadership, Organizational Structures, Stress, Healthand safety, case studies.

Text Books:

1. Bob Hughes, Mikecotterell, “Software Project Management”, Third Edition, Tata

McGraw Hill, 2004.

REFERENCE Books

1. Ramesh, Gopalaswamy, "Managing Global Projects", Tata McGraw Hill, 2001.

2. Royce, “Software Project Management”, Pearson Education, 1999.

3. Jalote, “Software Project Management in Practice”, Pearson Education, 2002.

Learning Outcomes:

The aim of the course is to help the student to be a responsible member of the software development team. The student after completion of the course shall be able to comprehend project problems and apply the knowledge on projects and software development. The student also shall be aware of the conditions and constraints such as resources, time, cost and quality.

Devi Ahilya University, Indore, India Institute of Engineering & Technology				I Year M.E. (Computer Engineering Sp. in Software Engineering) (Full Time)			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
SER2C5	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
Design Pattern	**3**	**1**	**2**	**3**	**1**	**1**	**5**
Duration of Theory Paper: 3 Hours							

Learning Objectives: To strengthen the knowledge of Object Oriented Design and Development by understanding various design patterns.

Pre requisites: Knowledge of object oriented system concepts, object oriented analysis and modeling and object oriented programming.

COURSE CONTENTS

UNIT-I

Introduction to Software Patterns, Overview of UML, Class Diagrams, Collaboration Diagrams, State chart Diagram, Deployment Diagram, Fundamental Design Patterns: Delegation, Interface, Abstract Super-class, Interface and Abstract class, Immutable, Marker Interface.

UNIT-II

Simple Factory pattern, Factory Method, Abstract Factory, Builder, Prototype, Singleton

UNIT-III

Adaptor, Bridge, Composite, Façade, Flyweight, Decorator, Proxy Pattern

UNIT-IV

Chain of Responsibility, Command, Interpreter, Mediator, Memento Pattern

UNIT-V

Observer, State, Strategy, Template Method, Visitor, Iterator Pattern.

Learning Outcomes: To learn Various Design Patterns and learn their application in real software development..

BOOKS RECOMMENDED:

[1]. Gamma, Helm, Johnson, Vlissides, Design Patterns. Elements of Reusable Software., Pearson Education 2006
[2]. Cooper, J. W., Java Design Patterns, A Tutorial, Pearson Education, 2000.
[3]. Freeman, Freeman, Head First Design Patterns, O'Reilly Pub. 2007
[4]. Mark Grand, Patterns in Java Vol. 1, Wiley 2002
[5]. Mark Grand, Patterns in Java Vol. 2, Wiley 2002
[6]. Mark Grand, Patterns in Java Vol. 3, Wiley 2002
[7]. Douglas Schmidt, Pattern Oriented Software Architecture Vol1, John Wiley 2000, also called as POSA

Devi Ahilya University, Indore, India Institute of Engineering & Technology				I Year M.E. (Computer Engineering Sp. in Software Engineering) (Full Time)			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
SER2C6	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
Software Testing & QA	**3**	**1**	**2**	**3**	**1**	**1**	**5**
Duration of Theory Paper: 3 Hours							

Learning Objectives: To develop a skill in developing good quality in the software product.

Pre requisites: Basic knowledge of software Engineering and programming.

COURSE CONTENTS

UNIT-I

SOFTWARE TESTING PRINCIPLES: Need for testing - Psychology of testing - Testing economics – Various software development Life cycles (SDLC) –Principles of testing.

UNIT-II

WHITE BOX TESTING White box testing techniques - Statement coverage - Branch Coverage - Condition coverage - Decision/Condition coverage - Multiple condition coverage - Dataflow coverage - Mutation testing - Automated code coverage analysis

UNIT-III

BLACK BOX TESTING: Black box testing techniques - Boundary value analysis - Robustness testing - Equivalence partitioning -Syntax testing - Finite state testing - Levels of testing – Unit testing- Integration Testing

UNIT-IV

TESTING STRETAGIES: System testing - Functional testing-non-Functional testing-acceptance testing-performance testing –Factors and Methodology for Performance testing, Regression testing-Methodology for Regression-testing.

UNIT-V

ADVANCE SOFTWARE TESTING METHOD (OBJECT ORIENTED TESTING): Syntax testing - Finite state testing - Levels of testing - Unit, Integration and System Testing. Challenges - Differences from testing non-OO Software - Class testing strategies - State-based Testing Software quality Assurance: ISO 9000; CMM and Test Management Issues; Quality Assurance personnel Issues.

Learning Outcomes: To learn to Software Testing & QA concepts and its approaches to software Testing and QA.

.

BOOKS RECOMMENDED:

[1]. Srinivasan Desikan&Gopalswamy Ramesh “Software testing Principles and Practices” Pearson education, 2006

[2]. R. Patton; Software Testing; Techmedia (SAMS) 2000

[3]. GlenfordJ.Myers, " The Art of Software Testing ", John Wiley & Sons, 1979.

[4]. Boris Beizer, Black-Box Testing: “Techniques for Functional Testing of Software and Systems ", John Wiley & Sons, 1995.

[5]. P.C.Jorgensen, “Software Testing - A Craftsman’s Approach ", CRC Press, 1995.

[6]. Robert V.Binder, " Testing Object-Oriented Systems: Models Patterns and Tools ", Addison Wesley, 2000.

Devi Ahilya University, Indore, India Institute of Engineering & Technology				I Year M.E. (Computer Engineering Sp. in Software Engineering) (Full Time)			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
SER2E4	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
Speech andLanguageProcessing	**3**	**1**	**2**	**3**	**1**	**1**	**5**
Duration of Theory Paper: 3 Hours							

Objective: To gain the knowledge for developing advanced technology of computer systems like speech recognition and machine translation.

Prerequisite: Discrete structures, Finite automata, information retrieval and Context-free Grammar

SPEECH ANDLANGUAGEPROCESSING

UNITI
Natural Language Processing, Applications, Ambiguity, Morphology, Parsing with Finite State Transducers, Regular Expressions, Stemmer, Spelling errors.

UNITII
Computational Phonology: speech sound, phonetic transcription, text to speech; Pronunciation Variations, Bayesian Method to spelling and pronunciations, Minimum Edit Distance, Weighted Automata, N-grams.

UNITIII
HMM and speech recognition, Viterbi algorithm, Acoustic processing of speech, Feature Extraction, Speech Synthesis; Part-of-Speech Tagging: rule based, stochastic, transformation based.

UNITIV
Syntax Processing: Parsing with CFG, CKY parsing and the Earley parser, Probabilistic parsing; Semantic Processing: Meaning representation, First Order Predicate Calculus. Lexical Semantics: Internal structure of words, thematic roles, Primitive decomposition, WordNet.

UNITV
Word sense disambiguation; Information Retrieval: Vector space model, Improving user queries; Pragmatic Processing: Discourse; Natural Language Generation, Machine Translation.

TEXTBOOKS:
[1] D. Jurafsky and J.H. Martin; Speech and Language Processing; Processing; Prentice Hall; 2000.
[2] 2. C. Manning and H. Schutze, “Foundations of Statistical Natural Language Processing”,
[3] JamesAllen.“NaturalLanguage Understanding”,AddisonWesley,1994.

List of Assignment in NLP Lab:

- Problem based on Stemming Algorithm.
- Problem based on Part of Tagging.
- Problem based on Parsing.
- Problem based on Information Retrieval.
- Case study on Different NLP Techniques
- Cricket Game Prediction.
- Machine Translation from English-Hindi.
- Query Expansion for Information Retrieval.
- Emotion detection for texts.
- Any other problem based on emerging trends in speech & language processing.

Institute of Engineering & Technology				Sp. in Software Engineering) (Full Time)			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
SER2E6 **Machine Learning**	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
	3	**1**	**2**	**3**	**1**	**1**	**5**
Duration of Theory Paper: 3 Hours							

Learning Objectives:

The field of machine learning is concerned with the question of how to build computer programs able to construct new knowledge or to improve already possessed knowledge by using input information. The goal of this course is to introduce the theoretical foundations of machine learning, to provide practical experience of applying machine learning techniques and to investigate new problems where machine learning techniques can do better.

Pre requisites: Basic knowledge of a programming language and Basic knowledge of probabilities and statistics is required.

COURSE OF CONTENTS

Unit-I

Introduction: Definition, Applications of machine learning, Importance of machine learning, Aspects of developing a learning system: training data and test data, Issues in machine learning, Types of learning: supervised, unsupervised and Reinforcement learning, Concept learning, General-to-specific ordering of hypotheses. Version spaces and the candidate elimination algorithm.

Unit-II

Supervised Learning: Classification and Regression learning methods**,** Decision Tree Learning**:** Representing concepts as decision trees, ID3 algorithm. Picking the best splitting attribute, searching for simple trees and computational complexity. Regression and function approximation, linear regression and best fit, Order of polynomial, Polynomial regression, Cross validation.

Unit-III

Unsupervised Learning: Introduction to unsupervised learning -Clustering -Classification of clustering algorithms: K-Means and EM -Factor Analysis: PCA (Principal Components Analysis) and ICA (Independent Component Analysis) -Self-Organized Maps (SOM) and Multi-dimensional Scaling.

Unit-IV

Computational Learning theory, Introduction, PAC Learning, VC dimension, Support Vector Machines (SVM), Genetic Algorithm (GA), illustrative examples for SVM and GA.

Unit-V

Artificial Neural Networks Learning, Introduction, Neural Network Representation, Perceptron, Backpropagation algorithm, Examples of Neural Network Learning.

RECOMMENDED BOOKS

[1] Tom Mitchell, *Machine Learning*, McGraw-Hill, 1997.
[2] Richard O. Duda, Peter E. Hart & David G. Stork, *Pattern Classification*, Wiley & Sons, 2001.
[3] EthemAlpaydin, *Introduction to Machine Learning*, MIT Press, 2004.
[4] David E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Kluwer Academic Publishers, Boston, MA, 1989.
[5] ZbigniewMichalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, 1999.

Learning Outcomes:

Upon Completing the Course, students will have knowledge of various machine learning techniques useful for solving the real world problems.

List of Assignment in Machine Learning Lab:

- **Problem based on different machine Learning algorithm**
- **Works on different machine learning Tools**
- **Case Study on different data sets**

Devi Ahilya University, Indore, India Institute of Engineering & Technology				I Year M.E. (Computer Engineering Sp. in Software Engineering) (Full Time)			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
SER2E5 Aspect Oriented Software Engineering	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
	3	**1**	**2**	**3**	**1**	**1**	**5**
Duration of Theory Paper: 3 Hours							

Learning Objectives:

The objective of this course is to master basics of aspect-oriented software development, which enables a higher degree of the separation of concerns through crosscutting concern modularization. The course provides an overview of aspect-oriented approaches to software development throughout all of its stages, as well as programming languages connected with these approaches. The course also covers the relationship of aspect-oriented software development and software product lines. Students will gain experience with AspectJ, which is the most important aspect-oriented programming language of today.

Pre-requisite:

Familiarity with Object oriented programming, Object oriented design, UML is essential.

COURSE CONTENTS

Unit 1: Introduction to AOSD

This module provides a broad overview of aspect orientation. It introduces students to the aspect-oriented paradigm's origins and foundations, providing a solid basis and a common terminology to be used in subsequent modules. The fundamental concepts include all major elements of the paradigm: separation of concerns, crosscutting concerns, modularization, aspects, join points, point cuts, advice, and aspectual composition. Module 1's only prerequisite is knowledge of software engineering in some existing, well known paradigm.

Unit 2: Aspect-oriented analysis and design

This module covers a broad spectrum of software development activities, from initial requirements definition to architecture derivation and detailed design production. Each of these life-cycle stages can be realized using various aspect-oriented approaches. This module underlines the problems of tangling and scattering caused by crosscutting concerns in non aspect-oriented analysis and design approaches. It also presents aspect-oriented approaches for aspect identification, modularization, and composition, using several case studies for illustration. An in-depth experience with a particular analysis and design technique and its related tools is a final important goal of the module. Students achieve hands-on experience of aspect-oriented analysis and design through exercises. The

prerequisites are Module 1 and familiarity with some requirements engineering, architecture, and design approaches. Knowledge of object-oriented (OO) analysis and design techniques (including UML) is desirable.

Unit 3: Aspect-oriented programming

Several AOP languages exist today, and most are extensions of existing languages. This module focuses on hands-on experience, giving special care to programming practices in AOP. The module covers various aspect languages, highlighting their differences and commonalities to teach students to abstract from concrete languages and understand aspect orientation's essential mechanisms. It also touches on implementing aspect language execution models to help students better understand the impact of aspects on program execution (for example, in terms of performance).The prerequisites are Module 1 and experiencein or knowledge about software implementation by means of contemporary languages, preferably in the OO paradigm.

Unit 4: Aspect-oriented applications

Module 5 illustrates the practical use of various aspect-oriented technologies, such as programming languages, aspect-oriented analysis and design, and more generally, any software engineering methodology that embraces aspects. It presents case studies of applications that benefit from AOSD, covering system level elements (such as middleware) and end-to-end user applications (such as e-banking or e-government applications). The module's main subject isn't technologies that create system-level elements and end-to-end user applications; Modules 2 and 3 will have covered these. The prerequisites are Module 1 and, depending on the instantiation, Module 2 and/or Module 3.

Unit 5: AOSD and other paradigms

Aspects are always used in a context. Therefore, to develop applications using aspect-oriented techniques, it's important to relate AOSD to the development paradigms, methodologies, and programming languages that you used to implement the underlying base application. AOSD's context is almost always class-based object orientation. However, as AOSD spreads to other contexts, this relationship will diversify and become more important. Also, other advanced development paradigms have been developed that can be related to AOSD, either because they're complementary or because they target the same problems as AOSD (albeit differently). This module provides insights into the relationship between AOSD and these other advanced development paradigms (for example, development methodologies other than the OO paradigm), different general-purpose programming languages for the base code, and component-oriented software engineering. The prerequisites are Module 1 and at least one other module.

Text Books:

1. Ivar Jacobson and Pan-Wei Ng. Aspect-Oriented Software Development with Use Cases. Addison-Wesley, 2004.
2. RamnivasLaddad. AspectJ in Action: Enterprise AOP with Spring Applications. Second edition, Manning, 2009.
3. Mastering AspectJ: Aspect-Oriented Programming in JavaBy Joseph D. Gradecki, Nicholas Lesiecki, Wiley 2003

Reference Books:

1. Robert E. Filman et al. Aspect-Oriented Software Development. Addison-Wesley, 2004.
2. Siobhan Clarke and Elisa Baniassad. Aspect-Oriented Analysis and Design: The Theme Approach. Addison-Wesley, 2005.

3. Aspect-oriented Programming with AspectJ. Ivan Kiselev, Sams, 2003

Learning Outcomes:

The students after completion of the course shall be having the knowledge of aspect-oriented software development, which enables a higher degree of the separation of concerns through crosscutting concern modularization. Students will be able to build solutions with AspectJ on completion of the course.

Devi Ahilya University, Indore, India Institute of Engineering & Technology				I Year M.E. (Computer Engineering Sp. in Software Engineering) (Full Time)			
Subject Code & Name	**Instructions Hours per Week**			**Credits**			
SER2G4	**L**	**T**	**P**	**L**	**T**	**P**	**Total**
Data Mining & Warehousing	**3**	**1**	**2**	**3**	**1**	**1**	**5**
Duration of Theory Paper: 3 Hours							

Objectives:

· To develop the abilities of critical analysis to data mining systems and applications.
· To implement practical and theoretical understanding of the technologies for data mining
· To understand the strengths and limitations of various data mining models.

Unit-I

Data Mining -Introduction: Data Mining Primitives, Languages, and System Architectures: Data mining primitives, Query language, Designing GUI based on a data mining query language, Knowledge Discovery in Databases (KDD), KDD Process, Data Preprocessing, Data Cleaning, Data Transformation, Data Compression and Dimension Reduction, Principal Component Analysis, Binning Methods.

Unit-II

Data Warehousing –Introduction and Design: Overview and Concepts: Need for data warehousing, basic elements of data warehousing, Architecture and Infrastructure: Architectural components, Infrastructure and metadata. Data Design And Data Representation: Principles of dimensional modeling, Dimensional modeling, data extraction, transformation and loading, data quality. OLAP in data warehouse –ROLAP, MOLAP, HOLAP. OLTP Vs OLAP, Various Data Warehouse Schemas.

UNIT-III

Association & Classification Techniques: Introduction, Frequent itemset mining methods – Apriori, FP-Growth, Pattern evaluation methods, Basic concepts of classification, Decision tree induction, Bayes classification, Rule-based classification.

UNIT-IV

Clustering Techniques: Introduction, Clustering paradigms; Partitioning algorithms – K-Means, K-Medioid, CLARA; Partition based clustering – BIRCH; Density based clustering - DBSCAN; Categorical clustering algorithms, Evaluation of Clustering.

UNIT-V

Other DM techniques & Web Mining: Spatial Mining, Spatial Mining tasks, Spatial clustering, Spatial Trends. Web Mining : Introduction Web content mining, Web structure Mining, Web Usage Mining.

Temporal and spatial DM: Temporal association rules, Sequence Mining, GSP, SPADE, SPIRIT, and WUM algorithms, Episode Discovery, Event prediction, Time series analysis.

Reference Books:

1. Data Mining Techniques; ArunK.Pujari ; University Press.
2. Data Mining Concepts and Techniques, Jiawei Hang Micheline Kamber,Jianpei, Morgan Kaufmannn.
3. Data Mining; Adriaans&Zantinge; Pearson education.
4. Mastering Data Mining; Berry Linoff; Wiley.
5. PaulrajPonniah, “Data Warehousing Fundamentals”, John Wiley.
6. Text Mining Applications, Konchandy, Cengage