B.Tech. *Computer Engineering* **SEMESTER V**

Course No.	Type	Course	L	T	P	Credits	Evaluation Scheme (Percentage weights)					Offering Dept.	AICTE COURSE TYPE	Pre-requisite	
							Theory			Practical					
							CA	MS	ES	CA	ES			Code	Title
COCSC14	CC	Principles of Compiler Construction	3	0	2	4	15	15	40	15	15	CSE	PROGRAM CORE	COCSC10	Theory of Automata & Formal languages
COCSC15	CC	Cloud Computing	3	0	2	4	15	15	40	15	15	CSE	PROGRAM CORE	COECC12	Data Communication
COCSC16	CC	Data Mining	3	0	2	4	15	15	40	15	15	CSE	PROGRAM CORE	COCSC05	Database Management Systems
COCSC17	CC	Machine Learning	3	0	2	4	15	15	40	15	15	CSE	PROGRAM CORE	COMTC13	Probability and Stochastic Processes
COCSExx	ED					4									
COCSExx	ED					4									
	EO	Elective Open	-	-	-	4	-	-	-	-	-	-	MANDATORY COURSE		
			28 2*			**28**									

2*: The actual weekly load depends upon the elective chosen by the student under FE. Maximum 28 credits.

Minor 1: Software Designing and Development			**Minor 2: Artificial Intelligence**			**Minor 3: Data Science**		
Course No.	Course Name	Prerequisite	Course No.	Course Name	Prerequisite	Course No.	Course Name	Prerequisite
COCSE01	Software Testing 3 0 2	COCSC11	COCSE04	Semantic Web 3 1 0	COCSC05	COCSE07	Data Handling and Visualization Tools 3 0 2	COCSC05
COCSE02	Software Quality 3 1 0	COCSC11	COCSE05	Object oriented analysis and design 3 0 2	COCSC02	COCSE08	Web Analytics 3 1 0	COCSC05
COCSE03	Software Security3 1 0	COCSC11	COCSE06	Cryptography techniques 3 1 0	COCSC01	COCSE09	Object oriented Databases 3 1 0	COCSC05

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
COCSC14	CC	Principles of Compiler Construction	3	0	2	4	15	15	40	15	15	Theory Automata and Formal Languages

COURSE OUTCOMES

1 Understand the internal organization and behavior of the compilers and other language processors.
2 Apply the formal constructs for designing a compiler.
3 Study and understand the functioning of a compiler.
4 Gain an ability to design simple domain-specific languages (DSLs) using compiler construction tools.
5 Go for the translation of languages or design the tools for online processing.

COURSE CONTENTS

Unit 1

Introduction: Language processors, structure of a compiler, compiler-construction tools, evolution of programming languages, applications of compiler technology, Transition diagrams, bootstrapping, just-in-time compilation.

Unit 2

Lexical analysis: Input buffering, specification and recognition of tokens, lexical analyzer generator.

Unit 3

Syntax analysis: Specification of syntax using grammar. Top-down parsing – recursive-descent, predictive. Bottom-up parsing – shift-reduce, SLR, CLR, LALR. Parser generator.

Unit 4

Intermediate-code generation: Syntax-directed translation. Three-address code. Translation of declarations, expressions, control flow. Backpatching. Runtime environment: Activation trees and records.

Unit 5

Code optimization: Sources of optimization, basic blocks, optimization of basic blocks, data-flow analysis, loop optimizations. Code generation: Issues, register allocation and assignment, peephole optimization

Practical:

1. Develop simple language processors like desk calculator and assembler.
2. Design a small high-level language.
3. Develop a lexical analyzer and a syntax analyzer for the same using the LEX and YACC tools. Also implement the bookkeeper module.
4. Design a small high-level language and implement a compiler for the same. If the target machine of the compiler is a hypothetical machine, then implement a simulator for it.
5. Develop a simple calculator using LEX and YACC tools.
6. Implement a program for symbol table using hashing
7. Implement a two-pass assembler

8. Implement a bottom-up parser using YACC tool.
9. Represent 'C' language using Context Free Grammar
10. Add assignment statement, If then else statement and while loop to the calculator and generate the three address code for the same.

SUGGESTED READINGS

1. Aho, A. V., Lam, M. S., Sethi, R. and Ullman J. D., "Compilers – Principles, Techniques and Tools (2nd ed.)", Pearson.
2. Chattopadhyay, S. 2005, "Compiler Design, PHI".
3. Appel, A. W. 200, "Modern Compiler Implementation in C", Cambridge University Press.
4. Kenneth C. Louden (1997), Compiler Construction– Principles and Practice, 1st edition, PWS Publishing.

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
COCSC15	**CC**	**Cloud Computing**	**3**	**1**	**0**	**4**	**25**	**25**	**50**			**Networking**

COURSE OUTCOMES

1. Understand the concept of cloud computing, its quality issues, services, applications, benefits and limitations.
2. Understand the underlying technologies that drive a cloud computing environment.
3. To keep abreast of the trends in cloud technology
4. To get acquainted with available cloud environments such as GoogleApps, Microsoft Azure and Amazon Web Services.
5. To be able to use the cloud services.

COURSE CONTENT

UNIT I

Introduction: Concept of a cloud, Purpose, characteristics, challenges and developments in cloud computing, Virtualization, On-demand Cloud Computing, Current cloud Technologies and Environments, Benefits and limitations.

UNIT II

Virtualization: Characteristics of virtualization, Types of virtualization, Hypervisors and some case studies.

UNIT III

Cloud architectures: Software as a Service, Platform as a Service, Infrastructure as a Service, Storage as a Service, Applications as a Service, other services

UNIT IV

Types of cloud architectures: Public, Private, Hybrid, Design issues with cloud: scalability, fault tolerance, security, trust, privacy.

UNIT V

Data in the cloud: GFS,HDFS, Big Tables.

Concurrent Computing: Thread programming, MPI programming, Parallel Computing with Map Reduce and extensions.

Case studies and emerging trends: Related to issues in migration to cloud, Cloud computing economics etc.

SUGGESTED READINGS

1. K. Chandrasekaran, "Essentials of Cloud Computing,"
2. T. Velte, A. Velte and R. Estenpeter, "Cloud Computing – A practical approach,"
3. U.S. Pandey, Kavita Choudhary, "Cloud Computing,"
4. S. Chand,R. Buyya, C. Vecchiola, S.T. Selvi, "Mastering Cloud Computing,"

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
COCSC16	**CC**	**Data Mining**	**3**	**0**	**2**	**4**	**15**	**15**	**40**	**15**	**15**	**Databases**

COURSE OUTCOMES

1. To introduce students to the basic concepts and techniques of Data Mining.
2. To develop skills of using recent data mining software for solving practical problems.
3. To gain experience of doing independent study.
4. To encourage students for independent research.
5. To impart knowledge for broader view of mining operations.

COURSE CONTENTS

Unit I

Introduction to Data Mining : What is data mining, Related technologies - Machine Learning, DBMS, OLAP, Statistics , Data Mining Goals , Stages of the Data Mining Process , Data Mining Techniques ,Knowledge Representation Methods , Applications , Example: weather data

Unit II

Data Warehouse and OLAP , Data Warehouse and DBMS , Multidimensional data model , OLAP operations , Example: loan data set , Data preprocessing , Data cleaning , Data transformation , Data reduction , Discretization and generating concept hierarchies , Installing Weka 3 Data Mining System , Experiments with Weka - filters, discretization

Unit III

Data mining knowledge representation , Task relevant data , Background knowledge , Interestingness measures , Representing input data and output knowledge , Visualization techniques , Experiments with Weka - visualization ,

Attribute-oriented analysis , Attribute generalization , Attribute relevance , Class comparison , Statistical measures Experiments with Weka - using filters and statistics , Data mining algorithms: Association rules , Motivation and terminology , Example: mining weather data , Basic idea: item sets , Generating item sets and rules efficiently, Correlation analysis , Experiments with Weka - mining association rules

Unit IV

Data mining algorithms: Classification , Basic learning/mining tasks , Inferring rudimentary rules: 1R algorithm , Decision trees , Covering rules , Experiments with Weka - decision trees, rules , Data mining algorithms: Prediction , The prediction task , Statistical (Bayesian) classification , Bayesian networks , Instance-based methods , (nearest neighbor) , Linear models , Experiments with Weka - Prediction , Evaluating what's been learned , Basic issues , Training and testing , Estimating classifier accuracy (holdout, cross-validation, leave-one-out) , Combining multiple models (bagging, boosting, stacking) , Minimum Description Length Principle (MLD) Experiments with Weka - training and testing

Unit V

Mining real data , Preprocessing data from a real medical domain , Applying various data mining techniques to create a comprehensive and accurate model of the data. Clustering , Basic issues in clustering , First conceptual clustering system: Cluster- 2 , Partitioning methods: k-means, expectation maximization (EM) , Hierarchical methods: distance-based agglomerative and divisible clustering , Conceptual clustering: Cobweb Experiments with Weka - k-means, EM, Cobweb , Advanced techniques, Data Mining software and applications Text mining: extracting attributes (keywords), structural approaches (parsing, soft parsing). Bayesian approach to classifying text , Web mining: classifying web pages, extracting knowledge from the web , Data Mining software and applications

SUGGESTED READINGS

1. introduction to Data Mining- Tan, Steinbach & Kumar
2. An Introduction to Statistical Learning: with Applications in R- Gareth James & Daniela Witten
3. Data Science for Business: What you need to know about data mining and data-analytic thinking - Foster Provost & Tom Fawcett
4. Modeling With Data - Ben Klemens

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
COCSC17	**CC**	**Machine Learning**	**3**	**0**	**2**	**4**	**15**	**15**	**40**	**15**	**15**	**Algorithms**

COURSE OUTCOMES

1. To develop an understanding of the fundamentals of machine learning.
2. To develop an understanding of statistical pattern recognition.
3. To gain an insight into the various components of machine learning such as supervised learning, unsupervised learning, learning theory, reinforcement learning and adaptive control.
4. To acquire skills that can be applied to various components of machine learning to applications like robotic control, data mining, autonomous navigation, bioinformatics, speech recognition, and text and web data processing.
5. To apply the knowledge gained to the projects

COURSE CONTENTS

Unit I

Introduction: Definition of learning systems. Goals and applications of machine learning.
Inductive Classification: The concept learning task. Concept learning as search through a hypothesis space. General-to-specific ordering of hypotheses. Finding maximally specific hypotheses. Version spaces and the candidate elimination algorithm. Learning conjunctive concepts.

Unit II

Ensemble Learning: Using committees of multiple hypotheses. Bagging, boosting, and DECORATE. Active learning with ensembles.
Experimental Evaluation of Learning Algorithms: Measuring the accuracy of learned hypotheses. Comparing learning algorithms: cross-validation, learning curves, and statistical hypothesis testing.

Unit III

Computational Learning Theory: Models of learnability: learning in the limit; probably approximately correct (PAC) learning. Sample complexity: quantifying the number of examples needed to PAC learn. Computational complexity of training.
Sample complexity for finite hypothesis spaces. PAC results for learning conjunctions, kDNF, and kCNF. Sample complexity for infinite hypothesis spaces, Vapnik- Chervonenkis dimension.

Unit IV

Rule Learning: Propositional and First-Order, Translating decision trees into rules. Heuristic rule induction using separate and conquer and information gain. First-order Horn-clause induction (Inductive Logic Programming) and Foil. Learning recursive rules. Inverse resolution, Golem, and Progol.
Artificial Neural Networks: Neurons and biological motivation. Linear threshold units. Perceptrons: representational limitation and gradient descent training.
Multilayer networks and backpropagation. Hidden layers and constructing intermediate, distributed representations. Overfitting, learning network structure, recurrent networks.

Unit V

Support Vector Machines: Maximum margin linear separators. Quadractic programming solution to finding maximum margin separators. Kernels for learning non-linear functions.
Bayesian Learning: Probability theory and Bayes rule. Naive Bayes learning algorithm. Parameter smoothing. Generative vs. discriminative training. Logisitic regression. Bayes nets and Markov nets for representing dependencies.

Guidelines for project based work: Semester long projects, presentations, research work, term papers based on the above topics.

SUGGESTED READINGS

1. Richard Duda, Peter Hart and David Stork, Pattern Classification, 2nd ed. John Wiley & Sons, 2001.
2. Tom Mitchell, Machine Learning. McGraw-Hill, 1997.
3. Richard Sutton and Andrew Barto, Reinforcement Learning: An introduction. MIT Press, 1998
4. Trevor Hastie, Robert Tibshirani and Jerome Friedman, The Elements of Statistical Learning. Springer, 2009

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
COCSE01	ED	Software Testing	3	0	2	4	15	15	40	15	15	Software Engineering

COURSE OUTCOMES

1. Learn the various concepts, terminology and methods that can be used to test software before it is delivered to the end user.
2. To Learn different Verification and Validation Testing Techniques in detail
3. To learn different System testing, Adhoc testing and Combinatorial testing techniques
4. To learn different Testing models, Testing Metrics , Prioritization techniques for test cases, Test plan etc.
5. Learn about various challenges and difficulties faced during the process of software testing and approach for tackling them.

COURSE CONTENTS

Unit I

Introduction to Software Testing: Definition, Goals, Role of testing in SDLC, Discussion of testing terminology such as error, bug and failure, test case ,Test plan, Oracle etc., Testing vs Debugging, Testing Life Cycle Models, Role of a tester
Software Verification and validation, Role of verification and Validation in Testing Strategy.
Verification methods like Inspections, Walkthroughs and reviews in detail, Reading techniques, Check lists, SRS document verification, SDD document verification.

Unit II

Validation Testing Techniques –
White Box Testing Techniques : Logic Coverage Criteria, Basis Path Testing, Loop testing, Data Flow Testing, slice based testing
Black Box Testing techniques : Boundary Value Analysis(BVA), Equivalence Class Testing, State-Table Based Testing, decision Table Based Testing, Cause-Effect Graphing Based Testing.

Unit III

Validation Testing activities - Unit testing, Integration Testing, Usability Testing
Mutation Testing: Mutation and Mutants, mutation operators, Test assessment using mutation
Regression Testing : Test selection, minimization and prioritization of test cases for regression testing. Prioritization techniques in detail, Debugging
Introduction to Object Oriented Testing: Conventional testing vs OOT testing, Issues in OO testing, Testing of OO classes, Inheritance testing, Use case diagram based testing

Unit IV
Test Generation from Combinatorial Designs: Test Configuration and test set, A combinatorial test design process, Fault model, Latin squares, Mutually orthogonal Latin squares, Pairwise Design: Binary Factors/Multi valued factors, Orthogonal arrays, Covering and Mixed Level Covering arrays
Testing Metrics for Monitoring and Controlling the Testing Process
Unit V
System Testing techniques like- Performance testing, Load testing, Volume testing, Scalability testing, Recovery testing, Interoperability testing, Security testing etc.
Different Adhoc testing techniques, Test Management

Practical work for Software Testing:

1. Perform verification testing on a program using different checklists
2. Apply different white box testing techniques like condition coverage, basis path testing, data flow testing on a program
3. Writing and executing the Unit Test cases using Test Automation frameworks like NUnit, JUnit, Visual studio etc. for a program using
   -Boundary Value Analysis
   -Equivalence Partitioning
4. Writing and executing the Unit Test cases using Test Automation frameworks like NUnit, JUnit etc. for an Object Oriented program
5. Learn how to raise and report Bugs using Bug tracking tool (Bugzilla, Jira using QA Complete etc)
6. Study and use of Open Source tools for Combinatorial Testing to generate test cases using pairwise design
7. Study and Use of Mutation Testing Tools
8. Study and use of any performance testing tool like Jmeter

SUGGESTED READINGS:

1. Aditya P. Mathur, "Foundations of Software Testing, Fundamental Algorithms and Techniques", Pearson Education.
2. Naresh Chauhan, "Software Testing Principles and Practices", 2nd Edition, Oxford University Press.
3. Ramesh Desikan," Software Testing Principles and Practices", Pearson Education.
4. Yogesh Singh, "Software Testing", Cambridge University Press.

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
COCSE02	ED	Software Quality	3	1	0	4	25	25	50	-	-	Software Engineering

COURSE OUTCOMES

1. Understand the quantitative aspect of software quality.
2. Learn and apply prevalent software quality tools and techniques for measuring quality in traditional manufacturing set up.
3. Apply these tools and techniques in the software scenario.
4. Understand and learn the various quality management tools in the different stages of Software Development life cycle

COURSE CONTENTS

Unit I

Introduction to Quality: The Quality Tradition: Origins of Quality Movement: Deming and Crosby's view of quality, Different Views of Quality: Transcendental, User, Manufacturing, Product, Value

based, Total Quality Movement (TQM), Application of TQM to Software Engineering. Why does software fail, Software quality: definition, how is software quality different? Static quality attributes, Dynamic quality attributes.

Unit II

Software Quality Models: McCalls, Boehms, ISO9126, GQM, Gilb's template Quality Management, Quality assurance Standards, ISO standards, CMM, CMMI, 3 Sigma, 6 Sigma Statistical Process Control (SPC). Seven tools of quality control: Pareto Charts, Graphs, Check sheets, histograms, Scatter Plots, Cause and Effect Diagrams.

Unit III

Business Process Redesign (BPR): Benefits of BPR in software development, TQM and BPR poised opposite to each other, Quality Function Deployment (QFD).

Unit IV

Application of Seven Management and Planning tools for Software Requirements Capturing: Affinity diagrams, Interrelationship diagraphs, hierarchy diagrams, Matrix diagram, Matrix data analysis, process decision program chart, arrow Diagram/Precedence Diagram, Computer Aided quality engineering (CAQE) and tools for quality management.

Unit V

Metrics Software: Definition, Types of Software Metrics, Organisation, Project, Process, Product, Product Complexity metrics, Halsteads' Software Science Metrics, OO Metrics: Chidamber and Kemrer, OO metrics suite. Study and use of various available quality assurance tools in manufacturing and software scenarios

SUGGESTED READINGS:

1. Akao, Y., "Quality function deployment: Integrating Customer requirements into Product design," Taylor & Francis, 2004.
2. Crosby, P.,"Quality is free: The Art of Making Quality Certain,"
3. Fenton N., "Software Metrics a Rigrous Approach," Wiley.

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
COCSE03	**ED**	**Software Security**	**3**	**1**	**0**	**4**	**25**	**25**	**50**	**-**	**-**	

COURSE OUTCOMES

1. To get acquainted with security issues in software.
2. Different methods to find the main issue(s)
3. Getting the exposure of different crucial points in software
4. Learning the issues of different touchpoint in software security.
5. Getting exposure about the knowledge management and skills for software security

COURSE CONTENTS

Unit I

Software Security Fundamentals - Defining a Discipline, The Security Problem, The Trinity of Trouble: Why

the Problem Is Growing , Connectivity , Extensibility, Complexity , Security Problems in Software - Bugs and Flaws and Defects, The Range of Defects, The Problem with Application Security, Software Security and Operations. Solving the Problem: The Three Pillars of Software Security Applied Risk Management, Software Security Touchpoints, Knowledge A Risk Management Framework - Putting Risk Management into Practice, How to Use This Chapter, Five Stages of Activity, Understand the Business Context, Identify the Business and Technical Risks, synthesize and Rank the Risks, Define the Risk Mitigation Strategy, Carry Out Fixes and Validate, Measuring and Reporting on Risk.

Unit II

The RMF , Applying the RMF: KillerAppCo's iWare 1.0 Server, Understanding the Business Context Gathering the Artifacts , Conducting Project Research, Identifying the Business and Technical Risks, Developing Risk Questionnaires, Interviewing the Target Project Team, Analyzing the Research and Interview Data, Uncovering Technical Risks, Analyzing Software Artifacts, Synthesizing and Ranking the Risks, Reviewing the Risk Data, Conducting the Business and Technical Peer Review, Defining the Risk Mitigation Strategy, Brainstorming on Risk Mitigation, Authoring the Risk Analysis Report, Producing Final Deliverables, Carrying Out Fixes and Validating ,The Importance of Measurement, Measuring Return, Measurement and Metrics in the RMF , The Cigital Workbench

Unit III

Seven Touchpoints for Software Security- Introduction to Software Security Touchpoints - Seven Touchpoints, Code Review (Tools), Architectural Risk Analysis, Penetration Testing, Risk-Based Security Testing, Abuse Cases , Security Requirements, Security Operations, External Analysis, Touchpoints as Best Practices , Building a Software Security Group , Touchpoints to Success.
Architectural Risk Analysis - Common Themes among Security Risk Analysis Approaches, Traditional Risk Analysis Terminology, Knowledge Requirement, The Necessity of a Forest-Level View, A Traditional Example of a Risk Calculation, Limitations of Traditional Approaches, Modern Risk Analysis, Security Requirements, A Basic Risk Analysis Approach, Touchpoint Process: Architectural Risk Analysis, Attack Resistance Analysis, Ambiguity Analysis, Weakness Analysis Getting Started with Risk Analysis, Architectural Risk Analysis Is a Necessity

Unit IV

Software Penetration Testing-Penetration Testing Today , Software Penetration Testing—a Better Approach , Make Use of Tools , Test More Than Once, Incorporating Findings Back into Development , Using Penetration Tests to Assess the Application Landscape , Proper Penetration Testing Is Good. Risk-Based Security Testing-What's So Different about Security? , Risk Management and Security Testing, How to Approach Security Testing , Who , How , Thinking about (Malicious) Input , Getting Over Input , Leapfrogging the Penetration . Abuse Cases - Creating Useful Abuse Cases , Touchpoint Process: Abuse Case Development, Creating Anti-Requirements , Creating an Attack Model , Example, Abuse Cases Are Useful, Software Security Meets Security Operations - Enterprise Software Security Program

Unit V

The Business Climate , Building Blocks of Change, Building an Improvement Program, Establishing a Metrics Program, A Three-Step Enterprise Rollout, Continuous Improvement, Enterprise Information Architecture, Adopting a Secure Development Lifecycle
Knowledge for Software Security- Experience, Expertise, and Security, Security Knowledge: A Unified View ,Security Knowledge and the Touchpoints ,The Department of Homeland Security Build Security In Portal ,Knowledge Management Is Ongoing , Software Security Now.

SUGGESTED READINGS

Software Security: Building Security In – Garry Mcgraw, 2006 , Addison-Wesley Professional
ISBN: 9780321356703

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
COCSE04	ED	Semantic Web	3	1	0	4	25	25	50	-	-	Algorithms

COURSE OUTCOMES

1. Understand the rationale behind Semantic Web.
2. Model ontologies using Resource Description Framework (RDF).
3. Design RDF Schemas for ontologies, model and design ontologies using Web Ontology Language (OWL).
4. Query ontologies using SPARQL.
5. Understand and reflect on the principles of Ontology Engineering, make an association between Semantic web and Web 2.0. And apply Semantic web technologies to real world applications.

COURSE CONTENTS

Unit I

Introduction to the Semantic Web , ontologies and description logic

Unit II

Overview and Introduction: Knowledge Representation, Semantic Web in Depth: RDF and RDF Schema, Semantic Web in Depth: OWL. Resource description framework, lightweight ontologies, a query language for Resource description framework (RDF) - SPARQL

Unit III

Writing OWL ontologies: Protégé, Semantic Web Methodologies and Design Patterns, Semantic Web in Depth: SPARQL, Semantic Web in Depth: Rules.

Unit IV

Publishing on the Semantic Web: Linked Data, Semantic Web Vocabularies and Applications, Semantic Web vs Web2.0, Trust and Community.

Unit V

Applications: Information Integration, Ontology Alignment, Scalable Reasoning and Knowledge Acquisition.

SUGGESTED READINGS

1. A Semantic Web Primer, third edition, MIT Press, 2012, Grigoris Antoniou, Paul Groth, Frank van Harmelen and Rinke Hoekstra
2. Grigoris Antoniou, Frank Van Harmelen, A Semantic Web Primer, MIT Press, 2008.
3. Pascal Hitzler, Markus Krotzsch, Sebastian Rudolph, Foundations of Semantic Web Technologies, CRC Press, 2009.
4. Dean Allemang, James Hendler, Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL, Morgan Kauffmann, ISBN-10: 0-12-373556-4.
5. Geroimenko, Vladimir; Chen, Chaomei (Eds.) 2nd ed., 2006, XIV, 248 p. 108 illus., Hardcover ISBN: 978- 1-85233-976-0, Visualizing the Semantic Web XML-based Internet and Information Visualization, SpringerVerlag London Ltd; 2Rev Ed edition (Oct 2005).
6. Michael C. Daconta, Leo J. Obrst, Kevin T. Smith, The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management: A Guide to the Future of XML, Web Services and Knowledge Management, John Wiley & Sons (20 Jun 2003).
7. S Powers, Practical RDF (Paperback) , OReilly (1 Aug 2003).
8. Thomas B. Passin, Explorer's Guide to the Semantic Web (Paperback), Manning Publications (8

Jul 2004)

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
COCSE05	**ED**	**Object Oriented Analysis and Design**	**3**	**0**	**2**	**4**	**15**	**15**	**40**	**15**	**15**	DBMS, data structures

COURSE OUTCOMES

1. To appreciate the fact that software development cannot be done in an adhoc fashion and has to follow a disciplined systematic approach for timely development of software within budget using suitable Process model and techniques
2. To learn various techniques for Requirements Elicitation and Specification in order to develop SRS for a problem domain
3. To model a problem domain using Object oriented analysis and design using UML
4. To learn Different techniques for software project management like Feasibility Analysis, Cost and Effort Estimation, Scheduling a project
5. To learn about different Software Quality frameworks, OO metrics, Configuration Management etc.

COURSE CONTENTS

Unit 1

Introduction: Introduction to software engineering, Importance of software, The Software evolution, Software characteristics, Software components, Software applications, Crisis-Problem and causes. Difference between software engineering and system engineering

Software Process Models: Waterfall model, Evolutionary Models, prototyping, V Model, Spiral model Incremental Model, RAD Model etc. Introduction to Agile models like Scrum, Extreme Programming, Feature Driven Development, Crystal etc., Comparison between Traditional and Agile models

Unit 2

Requirement Engineering: Different Types of Requirements: Functional, Non Functional and Domain Requirements in detail, Requirem**ent** elicitation Techniques like interviews, questionnaire, brainstorming, JAD, Scenario, Mind mapping, Requirement workshop, Prototyping, CRC Cards etc. Requirements Management, Writing SRS as per IEEE standard, Quality characteristics of SRS

Unit 3

Requirements Specification: Difference between structured and Object Oriented Analysis, Different views of modeling, Quick review of ER diagram, Data flow diagrams, State Transition Diagrams, data Dictionary,
Introduction to Unified Modeling language Conceptual Model of Unified Modeling Language (UML), Use case Diagram, Activity Diagram, Class Diagram, Object Diagram,

Unit 4

Sequence Diagram, Communication Diagram, State Machine Diagram, Timing Diagram, Composite Structure diagram, Package Diagram, Component Diagram, Deployment Diagram, Introduction to Rational Unified Process(RUP)

Unit 5

Software project Management: Project Management Process, System Request, Feasibility Analysis in detail Project scheduling, Finding Critical Path, Effort Estimation for OO systems using Use case diagram, Introduction to Object Oriented Metrics, Configuration Management,

Software Quality : Software Quality Models like McCall's Quality model, Quality frameworks like Capability Maturity Model, ISO9001

Practical:

1. Choose a problem domain and textually write the Functional and Non Functional requirements of the domain. While writing make use of elicitation techniques discussed in the class
2. For a Problem domain develop a Mind-Map
3. Draw the static View and functional view of the Problem domain of exercise 1 using ER diagram and DFD
4. Draw the Dynamic view of working of an ATM Machine or Microwave Oven
5. Develop a Use Case Model(use Case diagram and Use case Narratives) for a problem domain
6. For the problem of Q5, draw activity diagram, sequence diagrams and Class diagram
7. Draw the state machine diagram for a CD player
8. Perform Effort estimation activity and Implement Critical path method using an open source Tool for a case study
9. Write SRS as per IEEE std-830 for the problem domain of Q5

SUGGESTED READINGS

1. Sommerville, "Software Engineering", 10th Edition Published by Pearson
2. Sangeeta Sabharwal, "Software Engineering: Principles and Techniques", Second Edition, Published by New Age International Publishers, 2020
3. R . S. Pressman, "Software Engineering – A practitioner's approach", McGraw Hill Int. Ed.
4. Object Oriented Modeling and Design with UML by Michael R Blaha and James R Rumbaugh, 2nd Edition, Pearson
5. UML distilled, Third Edition by Martin Fowler, Addison Wesley .
6. The Unified Software Development Process by I*var* Jacobson, Booch and Rumbaugh,Addison Wesley, 2007

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
COCSE06	**ED**	**Cryptography techniques**	**3**	**1**	**0**	**4**	**25**	**25**	**50**	**-**	**-**	Networking

COURSE OUTCOMES

1. Explain common attacks against network assets, the associated threats and vulnerabilities, and what network security personnel do to secure assets,
2. Explain how to use cryptography to help protect information and how to choose an appropriate encryption method for an organization
3. Help protect information in an organization by using authentication and access control and deploy and manage certificates.
4. Help protect transmission of data by identifying threats to network devices and implementing security for common data transmission, remote access, and wireless network traffic

5. Identify common security threats and vulnerabilities to directory services and DNS, and then apply security methods to help protect them

COURSE CONTENTS

Unit I

Foundation of Security & Cryptography: OSI security architecture, Security attack, security services and mechanisms, model of security, Classical encryption techniques: Substitution Techniques, Transposition Techniques and Steganography.

Unit II

Block Ciphers and Public key cryptography: Design Principle of Block Ciphers: DES, AES, Multiple Encryption, Block Cipher modes of operation, stream ciphers, RC4, Public Key Cryptography: RSA, Key management, Diffie-Hellman Key exchange, Elliptic Curve Cryptography

Unit III

Hashes & Digital Signatures: Authentication functions, Message authentication codes, Hash functions and their security, HMAC, CMAC, Secure hash algorithms, Digital Signature: Certificates & standards, authentication protocols

Unit IV

Authentication Applications: Kerberos, X.509 Authentication service, public key infrastructure, electronic Mail Security: pretty good privacy, S/MIME

Unit V

IP and Web Security Protocols: IPsec, Secure socket layer and transport layer security, secure e-transaction, System Security: Computer Virus, Firewall & Intrusion Detection, Trusted systems

SUGGESTED READINGS

1. Cryptography & Network Security by Stallings, William (Fourth Edition or later)
2. Foundations of Cryptography (Basic Tools), Oded Goldreich Cambridge 2001.
3. Cryptography: Theory and Practice, by Douglas R. Stinson, First Edition, second edition: first volume
4. An Introduction to Cryptology, Henk C.A. van Tilborg, Kluwer Academic Publishers, 1987

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
COCSE07	**ED**	**Data Handling and Visualization Tools**	**3**	**1**	**0**	**4**	**25**	**25**	**50**	**-**	**-**	

COURSE OUTCOMES

1. Getting introduction about the data science and its operations
2. Finding statistical analysis of the data
3. Getting introduction to the representation of data with various graphs and presentation tools
4. Using standard library for visualization
5. Getting exposure to various metrics , usage and presentation.

COURSE CONTENTS

Unit I

Introduction to Data Science – Evolution–Roles – Stages in a Data Science Project – Applications of Data Science in various fields – Data Security Issues.
Data Collection and Data Pre-Processing Data Collection Strategies – Data Pre-Processing Overview – Data Cleaning – Data Integration and Transformation – Data Reduction – Data Discretization.

Unit II

Exploratory Data Analytics Descriptive Statistics – Mean, Standard Deviation, Skewness and Kurtosis – Box Plots – Pivot Table – Heat Map – Correlation Statistics – ANOVA.
Model Development Simple and Multiple Regression – Model Evaluation using Visualization – Residual Plot – Distribution Plot – Polynomial Regression and Pipelines – Measures for In-sample Evaluation – Prediction and Decision Making.

Unit III

The Value of Visualization, Effective Use of Form and Space – Fundamentals of Graphs Readings: Graph Selection Matrix , Seven Common quantitative relationships in Graphs and how to display them, Constructing Correlation Bar And Paired Bar Graphs With Microsoft Excel, Integrity in Visualization, Visual Perception and Quantitative Communication, Effective Use of Form and Space – Detailed Design of Tables and Graphs.

Unit IV

Python language will be used for following parts:
Data Wrangling - Combining and Merging Data Sets – Reshaping and Pivoting – Data Transformation – String manipulations – Regular Expressions.
Data Aggregation and Group Operations GroupBy Mechanics – Data Aggregation – GroupWise Operations – Transformations – Pivot Tables – Cross Tabulations – Date and Time data types.
Visualization in Python Matplotlib Package – Plotting Graph - Controlling Graphs – Adding Text – More Graph Types – Getting and Setting Values – Patches.

Unit V

Model Evaluation Generalization Error – Out-of-Sample Evaluation Metrics – Cross Validation – Overfitting – Under Fitting and Model Selection – Prediction by using Ridge Regression – Testing Multiple Parameters by using Grid Search. Tools and library in python for related operations.

SUGGESTED READINGS

1. Jojo Moolayil, "Smarter Decisions : The Intersection of IoT and Data Science", PACKT, 2016.
2. Cathy O'Neil and Rachel Schutt , "Doing Data Science", O'Reilly, 2015.
3. David Dietrich, Barry Heller, Beibei Yang, "Data Science and Big data Analytics", EMC 2013
4. Raj, Pethuru, "Handbook of Research on Cloud Infrastructures for Big Data Analytics", IGI Global.
5. Joel Grus, "Data Science from scratch", O'Reilly, 2015.
6. Foundations of Data Science – Avrim Blum, John Hopcroft, Ravi Kannan – Hindustan Publishing

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
COCSE08	ED	Web Analytics	3	1	0	4	25	25	50	-	-	Web Technol

												ogy

COURSE OUTCOMES

1. Understanding the basic terminology and concepts.
2. Understanding surveys, impacts, linking and required tools.
3. Understanding the various technologies and tools for online offline web analytics
4. Apply web metric and web 2.0. concepts for web analytics and google analytics
5. Applying tools and technologies to projects.

COURSE CONTENTS

Unit I

Introduction: Definition, Process, Key terms: Site references, Keywords and Key phrases; building block terms: Visit characterization terms, Content characterization terms, Conversion metrics; Categories: Offsite web, On site web; Web analytics platform, Web analytics evolution, Need for web analytics, Advantages, Limitations. Data Collection: Click stream Data: Web logs, Web Beacons, JavaScript tags, Packet Sniffing; Outcomes Data: E-commerce, Lead generation, Brand/Advocacy and Support; Research data: Mindset, Organizational structure, Timing; Competitive Data: Panel-Based measurement, ISP-based measurement, Search Engine data.

Unit II

Qualitative Analysis: Heuristic evaluations: Conducting a heuristic evaluation, Benefits of heuristic evaluations; Site Visits: Conducting a site visit, Benefits of site visits; Surveys: Website surveys, Post-visit surveys, creating and running a survey, Benefits of surveys. Web Analytic fundamentals: Capturing data: Web logs or JavaScript's tags, Separate data serving and data capture, Type and size of data, Innovation, Integration, Selecting optimal web analytic tool, Understanding click stream data quality, Identifying unique page definition, Using cookies, Link coding issues.

Unit III

Web Metrics: Common metrics: Hits, Page views, Visits, Unique visitors, Unique page views, Bounce, Bounce rate, Page/visit, Average time on site, New visits; Optimization (e-commerce, non e-commerce sites): Improving bounce rates, Optimizing adwords campaigns; Real time report, Audience report, Traffic source report, Custom campaigns, Content report, Google analytics, Introduction to KPI, characteristics, Need for KPI, Perspective of KPI, Uses of KPI. Relevant Technologies: Internet & TCP/IP, Client / Server Computing, HTTP (Hypertext Transfer Protocol), Server Log Files & Cookies, Web Bugs.

Unit IV

Web Analytics 2.0:Web analytics 1.0, Limitations of web analytics 1.0, Introduction to analytic 2.0, Competitive intelligence analysis : CI data sources, Toolbar data, Panel data ,ISP data, Search engine data, Hybrid data, Website traffic analysis: Comparing long term traffic trends, Analyzing competitive site overlap and opportunities.

Unit V

Google Analytics: Brief introduction and working, Adwords, Benchmarking, Categories of traffic: Organic traffic, Paid traffic; Google website optimizer, Implementation technology, Limitations, Performance concerns, Privacy issues.

SUGGESTED READINGS

1. Clifton B., Advanced Web Metrics with Google Analytics, Wiley Publishing, Inc.2nd ed.

2. Kaushik A., Web Analytics 2.0, The Art of Online Accountability and Science of Customer Centricity, Wiley Publishing, Inc. 1st ed.
3. Sterne J., Web Metrics: Proven methods for measuring web site success, John Wiley and Sons

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
COCSE09	**ED**	**Object oriented Databases**	**3**	**1**	**0**	**4**	**25**	**25**	**50**	**-**	**-**	**COCSC05**

COURSE OUTCOMES

After the successful completion of the course the students will be able

1. to understand the concepts of object orientation
2. to analyze the issues in traditional database systems
3. to develop data models for object oriented DBMS
4. to apply the concepts of object orientation for development of database management systems
5. Study specific topics in database with current trends

COURSE CONTENTS

Unit I

Object Oriented Database Concept: Data types and Object, Evolution of Object-Oriented Concepts, Characteristics of Object Oriented Data Model. Object Hierarchies - Generalization, Specialization, Aggregation. Object Schema. Inter-object Relationships, Similarities and difference between Object Oriented Database model and Other Data models.

Unit II

Traditional Databases Vs Object oriented databases
Traditional Database Models & SQL: Limitations of Relational Model- Need for Object Orientation; Conceptual Database Design: ANSI/SPARC 3 Level Architecture; Logical Database Design and Physical Database Design - Overview.

Unit III

OODBMS Architecture: Semantic Database Approach, Object Oriented Programming Language for database– Persistent programming language, persistent objects, storage and access of persistent objects, Object Database Management Group (ODMG)– Object Query language: Object Definition Language and manipulation Language, Object oriented DBMS, ODMG Compliance, CORBA etc.

Unit IV
Object Relational Database: The Extended Relational Model, Issues in Relational databases, the Database Design for an Object Relational DBMS, The Structured Typed and ADTs, Extending the ER Model, Storage and Access Methods, Query languages for object relational databases, Query Processing, Data Access API

Unit V

Special Topics in Databases:

Non-SQL databases, Temporal databases, Deductive databases, database technology for decision support systems, Distributed and Web databases, Advanced database concepts, emerging technologies and applications.

SUGGESTED READINGS

1. Nabil R. Adam, Bharat K. Bhargava, Advanced database systems, Lecture Notes in Computer Science.
2. Object Oriented databases clearly explained, jan L. Harrington, Morgan Kaufmann, 2000.
3. Carlo Zanialo, Advanced database systems, Morgan Kauffmann, 1997.