

IV B.Tech I Semester

15ACS51-SOFTWARE TESTING

L	T	P	C
3	1	0	3

Course Objectives

1. To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
2. Analysis of various testing methodologies and procedure to design test cases.

UNIT I

Introduction: Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs.

Flow graphs and Path testing: Basics concepts of path testing, predicates, path predicates and Achievable paths, path sensitizing, path instrumentation, application of path testing.

UNIT II

Introduction to Testing Design Strategies – The Smarter Tester – Test Case Design Strategies

Test Case Design Black-Box Approach - Using Black Box Approach to Test Case Design
 Random Testing – Requirements based testing –Boundary Value Analysis – decision tables -
 Equivalence Class Partitioning- statebased testing– cause effect graphing – error guessing -
 compatibility testing – user documentation testing – domain testing.

UNIT III

Test Case Design White-Box Approach-Using White-Box Approach to Test design – Test Adequacy Criteria – static testing vs. structural testing – code functional testing - Coverage and Control Flow Graphs – Covering Code Logic – Paths – Their Role in White-box Based Test Design – code complexity testing – Evaluating Test Adequacy Criteria.

UNIT IV

Paths, Path products and Regular expressions: path products & path expression, reduction Procedure, applications, regular expressions & flow anomaly detection. **Logic Based Testing:** overview, decision tables, path expressions, kv charts, specifications.

UNIT V

State, State Graphs and Transition testing: state graphs, good & bad state graphs, state testing, Testability tips. **Graph Matrices and Application:** Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to tools like Win runner, QTP, Selenium, Jmeter).

Course Outcomes

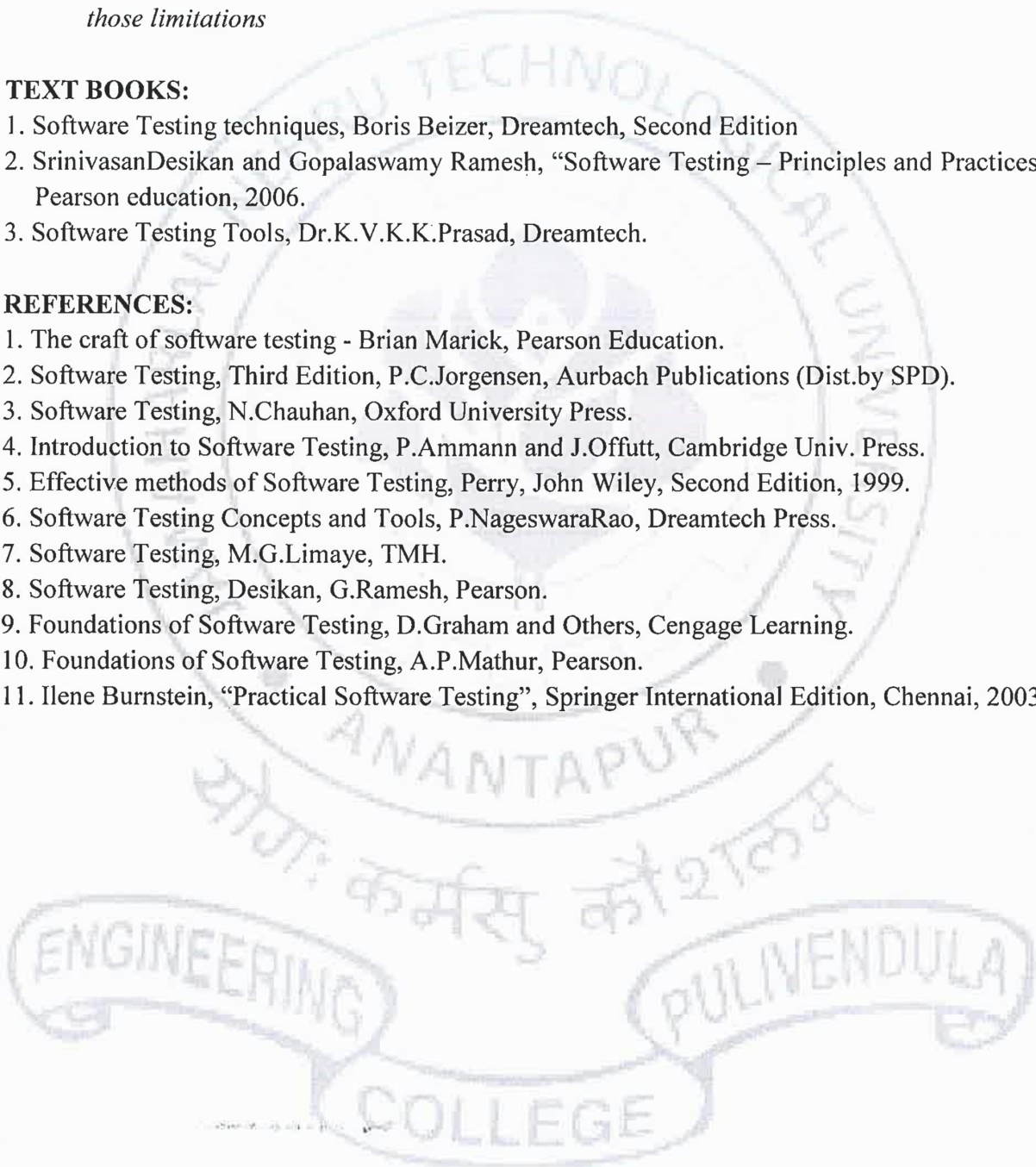
1. Analyze requirements to determine appropriate testing strategies.
2. Apply a wide variety of testing techniques in an effective and efficient manner
3. Compute test coverage and yield according to a variety of criteria
4. Evaluate the limitations of a given testing process and provide a succinct summary of those limitations

TEXT BOOKS:

1. Software Testing techniques, Boris Beizer, Dreamtech, Second Edition
2. SrinivasanDesikan and Gopaldaswamy Ramesh, "Software Testing – Principles and Practices", Pearson education, 2006.
3. Software Testing Tools, Dr.K.V.K.K.Prasad, Dreamtech.

REFERENCES:

1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing, Third Edition, P.C.Jorgensen, Aurbach Publications (Dist.by SPD).
3. Software Testing, N.Chauhan, Oxford University Press.
4. Introduction to Software Testing, P.Ammann and J.Offutt, Cambridge Univ. Press.
5. Effective methods of Software Testing, Perry, John Wiley, Second Edition, 1999.
6. Software Testing Concepts and Tools, P.NageswaraRao, Dreamtech Press.
7. Software Testing, M.G.Limaye, TMH.
8. Software Testing, Desikan, G.Ramesh, Pearson.
9. Foundations of Software Testing, D.Graham and Others, Cengage Learning.
10. Foundations of Software Testing, A.P.Mathur, Pearson.
11. Ilene Burnstein, "Practical Software Testing", Springer International Edition, Chennai, 2003



[Handwritten signatures]

IV B.Tech I Semester

15ACS52-ADVANCED COMPUTER NETWORKS

L	T	P	C
3	1	0	3

Course Objectives:

1. To give the student ideas and insights on important design issues associated with computer networks.
2. This course aims to provide advanced background on relevant computer networking topics to have a comprehensive and deep knowledge in computer networks.

UNIT-I**Review of Computer Networks and the Internet:**

What is the Internet, The Network core, Access Networks and Physical media, ISPs and Internet Backbones, Delay and Loss in Packet-Switched Networks, History of Computer Networking and the Internet.

Foundation of Networking Protocols: TCP/IP Model, Internet Protocols and Addressing, Equal-Sized Packets Model: ATM

UNIT-II

Networking Devices: Multiplexers, Modems and Internet Access Devices, Switching and Routing Devices, Devices in different layers, Router Structure.

The Link Layer and Local Area Networks: Link Layer: Introduction and Services, Error-Detection and Error-Correction techniques, Multiple Access Protocols, PPP: The Point-to-Point Protocol.

UNIT-III

Routing and Internetworking: Network-Layer Routing, Least-Cost-Path algorithms, Non-Least-Cost-Path algorithms, Intradomain Routing Protocols, Interdomain Routing Protocols, Congestion Control at Network Layer.

Internet Protocol: Internet working, IPv4, IPv6, Transition from IPv4 to IPv6.

Multicasting Techniques and Protocols: Basic Definition and Techniques, Intradomain Multicast Protocols, Interdomain Multicast Protocols, Node-Level Multicast algorithms.

UNIT-IV

Transport and End-to-End Protocols: Transport Layer, Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Mobile Transport Protocols, TCP Congestion Control.

Application Layer: Principles of Network Applications, The Web and HTTP, File Transfer: FTP, Electronic Mail in the Internet, Domain Name System (DNS), P2P File Sharing.

UNIT-V

Wireless Networks and Mobile IP: Infrastructure of Wireless Networks, Wireless LAN Technologies, IEEE 802.11 Wireless Standard, Cellular Networks, Mobile IP, Types of Wireless networks.

Mobile Adhoc Networks: Overview of Wireless Ad-Hoc Networks, Routing in Ad- Hoc Networks, Routing Protocols for Ad hoc Networks.

Course Outcomes:

1. Understanding of holistic approach to computer networking.
2. Ability to understand the computer networks and their applications.
3. Ability to design simulation concepts related to packet forwarding in networks.

TEXT BOOKS:

1. Computer Networking: A Top-Down Approach Featuring the Internet, James F. Kurose, Keith W. Ross, Third Edition, Pearson Education, 2007.
2. Computer and Communication Networks, Nader F. Mir, Pearson Education, 2007.

REFERENCES:

1. Data Communications and Networking, Behrouz A. Forouzan, Fourth Edition, Tata McGraw Hill, 2007.
2. An Engineering Approach to Computer Networking, S. Keshav, Pearson Education.
3. Computer Networks, Andrew S. Tanenbaum, Fourth Edition, Prentice Hall.
4. C. Sivaram Murthy, B.S. Manoj, "Ad hoc Wireless Networks – Architecture and Protocols", Second Edition, Pearson Education.



[Handwritten signature] *[Handwritten signature]*

IV B.Tech I Semester

15ACS53-CLOUD COMPUTING

L	T	P	C
3	1	0	3

Course Objectives:

1. To provide detailed and ample knowledge of the Cloud Computing fundamental issues, technologies, applications and implementations.
2. To expose the students to the cutting edge areas of Cloud Computing
3. To provide practical in-sight to the students in developing cloud applications
4. To shed light on the Security issues in Cloud Computing.

UNIT I – Fundamentals of Distributed Computing and Cloud Evolution

History of Centralized and Distributed Computing - Overview of Distributed Computing, Cluster computing, Grid computing. Technologies for Network based systems- System models for Distributed and cloud computing- Software environments for distributed systems and clouds.

UNIT II – Introduction to Cloud Computing and Virtualization

Introduction to Cloud Computing- Cloud issues and challenges - Properties - Characteristics - Service models, Deployment models. Cloud resources: Network and API - Virtual and Physical computational resources - Data-storage. Virtualization concepts - Types of Virtualization- Introduction to Various Hypervisors - High Availability (HA)/Disaster Recovery (DR) using Virtualization, load balancing and migration of VMs .

UNIT III – Cloud Application Programming

Cloud Programming and Software Environments- Parallel and Distributed Programming paradigms – Programming on Amazon AWS and Microsoft Azure – Programming support of Google App Engine- Aneka platform- Anatomy of Aneka Container- Building Aneka Clouds- Aneka SDK .

UNIT IV - Concurrent Computing and Data-Intensive Computing

Concurrent Computing:Introducing parallelism for single-machine computation, Programming applications with threads, Multithreading with aneka, Programming applications with aneka threads.

Data-Intensive Computing: What is data-intensive computing?,Technologies for data-intensive computingAnekaMapReduce programming.

UNIT V- Cloud Security

Cloud Access: authentication, authorization and accounting - Cloud Provenance and meta-data - Cloud Reliability and fault-tolerance - Cloud Security, privacy, policy and compliance- Cloud federation, interoperability and standards.




Course Outcomes: After successful completion of the course student will be able to:

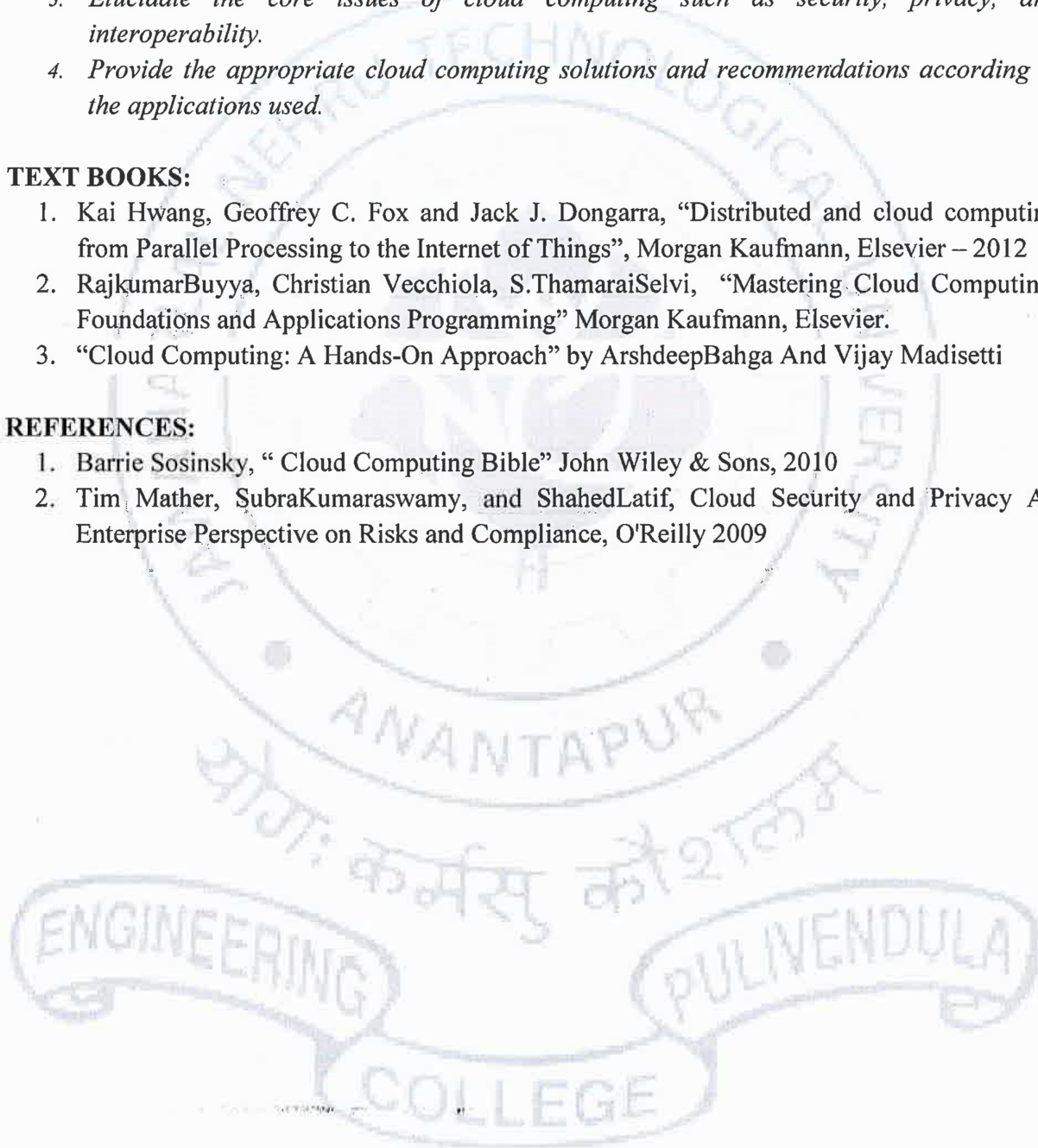
1. Communicate the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing
2. Categorise the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc.
3. Elucidate the core issues of cloud computing such as security, privacy, and interoperability.
4. Provide the appropriate cloud computing solutions and recommendations according to the applications used.



TEXT BOOKS:

1. Kai Hwang, Geoffrey C. Fox and Jack J. Dongarra, "Distributed and cloud computing from Parallel Processing to the Internet of Things", Morgan Kaufmann, Elsevier – 2012
2. RajkumarBuyya, Christian Vecchiola, S.ThamaraiSelvi, "Mastering Cloud Computing: Foundations and Applications Programming" Morgan Kaufmann, Elsevier.
3. "Cloud Computing: A Hands-On Approach" by ArshdeepBahga And Vijay Madiseti

REFERENCES:

1. Barrie Sosinsky, "Cloud Computing Bible" John Wiley & Sons, 2010
2. Tim Mather, SubraKumaraswamy, and ShahedLatif, Cloud Security and Privacy An Enterprise Perspective on Risks and Compliance, O'Reilly 2009



IV B.Tech I Semester

**15ACS54-INTERNET OF THINGS
(CBCC (DEPARTMENT SPECIFIC))**

**L T P C
3 1 0 3**

Course Objectives:

- To understand the fundamentals of Internet of Things.
- To build a small low cost embedded system using Arduino / Raspberry Pi or equivalent boards.
- To apply the concept of Internet of Things in the real world scenario.

UNIT I :Fundamentals of IOT

Introduction – Characteristics-Physical Design - Protocols – Logical Design – Enabling technologies – IoT Levels – Six Levels of IoT - Domain Specific IoTs.

UNIT II: IOT and M2M

M2M, IoTvs M2M, SDN and NFV for IoT, IOT system Management with NETCONF-YANG.

UNIT III: IoTDesign Methodology

IoT Systems Management – IoT Design Methodology – Specifications Integration and Application Development.

UNIT IV: Data Analytics for IoT

Apache Hadoop, Using HadoopMapReduce for Batch Data Analysis, Apache Oozie, Apache Spark, Apache Storm, Using Apache Storm for Real-time Data Analysis.

UNIT V :Tools for IoT

Chef, Puppet, IOT code generator Case studies: Chef. Puppet – Multi-tier Deployment, NETCONF-YANG, Raspberry Pi.

Course Outcomes:

- Design a portable IoT using Arduino/ equivalent boards and relevant protocols.
- Develop web services to access/control IoT devices.
- Deploy an IoT application and connect to the cloud.
- Analyze applications of IoT in real time scenario.

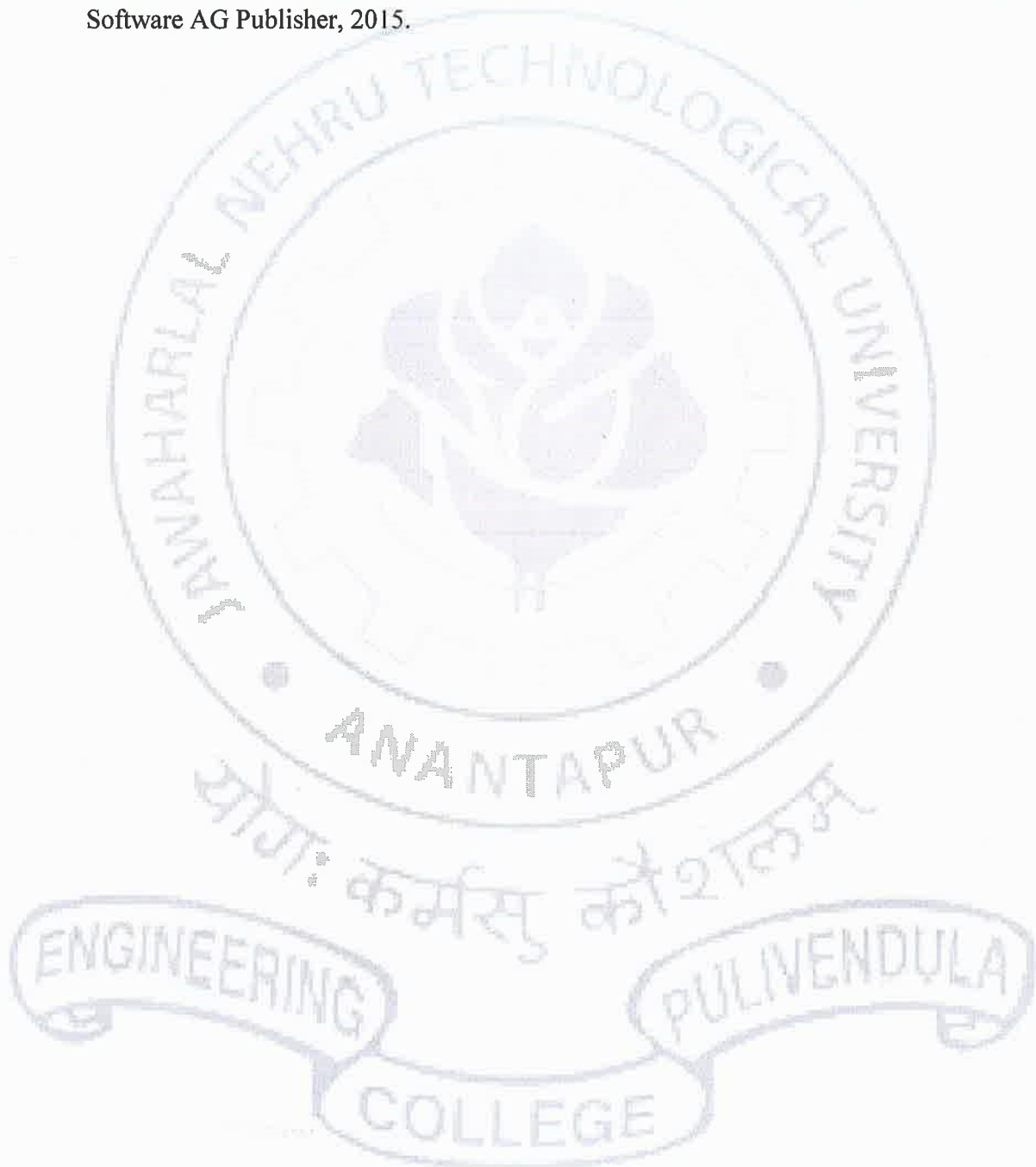
TEXT BOOKS:

1. ArshdeepBahga, Vijay Madiseti, “Internet of Things – A Hands-on Approach”, Universities Press, 2015.

REFERENCES:

1. Manoel Carlos Ramon, “Intel® Galileo and Intel® Galileo Gen 2: API Features and Arduino Projects for Linux Programmers”, Apress, 2014.
2. Marco Schwartz, “Internet of Things with the Arduino Yun”, Pack Publishing, 2014.

3. Simon Monk, "Programming the Raspberry Pi: Getting Started with Python", McGraw-Hill, 2013.
4. Charalampos Doukas, "Building Internet of Things With the Arduino", Second Edition, 2012.
5. Dr. John Bates, "Thingalytics: Smart Big Data Analytics for the Internet of Things", Software AG Publisher, 2015.



AM Grell

IV B.Tech I Semester

**15ACS55-NATURAL LANGUAGE PROCESSING
(CBCC (DEPARTMENT SPECIFIC))**

**L T P C
3 1 0 3**

Course Objectives: Upon completion, students will be able to explain and apply fundamental algorithms and techniques in the area of natural language processing (NLP). In particular, students will:

- Understand approaches to syntax and semantics in NLP.
- Understand current methods for statistical approaches to machine translation.
- Understand language modeling.
- Understand machine learning techniques used in NLP.

UNIT – I

Introduction to Natural Language Understanding, Syntactic Processing: Grammars and Parsing

UNIT-II:

Features and Augmented Grammars, Toward Efficient Parsing, Ambiguity Resolution

UNIT –III

Statistical Methods: Probabilistic Context-Free Grammars, Best-First Parsing.

UNIT-IV

Semantic Interpretation: Linking Syntax and Semantics, Ambiguity Resolution, other Strategies for Semantic Interpretation.

UNIT-V

Context and World Knowledge: Using World Knowledge, Discourse Structure, Defining a Conversational Agent.

TEXT BOOK:

1. Natural Language Understanding – James Allen, Second Edition, Pearson Education.

REFERENCE BOOKS:

1. Speech and Language Processing – Daniel Jurafsky, James H.Martin.
2. Foundations of Statistical Natural Language Processing – Christopher Manning, HinrichSchutze, MITPress.
3. Charniack, Eugene, Statistical Language Learning, MIT Press, 1993.
4. Jurafsky, Dan and Martin, James, Speech and Language Processing, 2nd Edition, Prentice Hall,2013-2014,2008.
5. Manning, Christopher and Henrich, Schutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999

IV B.Tech I Semester

**15ACS56-SOFTWARE PROJECT MANAGEMENT
(CBCC (DEPARTMENT SPECIFIC))**

**L T P C
3 1 0 3**

Course objectives:

1. To edify the core concepts of Software project management such as scheduling a software project, effort estimation, cost estimation.
2. To enhance the skills of analysing, Planning, developing a cost effective software project
3. To implant strong professional qualities and team dynamics that enables an individual to work successfully in teams with a single point objective

UNIT I- Project Evaluation and Project Planning :

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

UNIT II- Software Economics - Effort & Cost Estimation :

Evolution of Software Economics: Software Economics, pragmatic software cost estimation

Improving Software Economics: Reducing Software product size, improving software processes, improving team effectiveness, improving automation, Achieving required quality, peer inspections.

Estimation Techniques: COSMIC Full function points, COCOMO II A Parametric Productivity Model Staffing Pattern.

UNIT III - Activity Planning and Risk Management :

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

UNIT IV- Tracking Workflow of The Software Project :

Work Flows of the process: Software process workflows, Inter trans workflows.

Checkpoints of the Process: Major Mile Stones, Minor Milestones, Periodic status assessments.

Iterative Process Planning: Work breakdown structures, planning guidelines, cost and schedule estimating, Interaction planning process, Pragmatic planning.

Project Organizations and Responsibilities: Line-of-Business Organizations, Project Organizations, evolution of Organizations.

Process Automation: Automation Building Blocks, the Project Environment

UNIT V- Staffing and Project Control :

Staffing- Managing people – Organizational behavior – Best methods of staff selection – The Oldham Hackman job characteristic model – Ethical and Programmed concerns – Working in teams – Decision making – Team structures – Virtual teams – Communications genres – Communication plans

Project Control and Process instrumentation: The server care Metrics, Management indicators, quality indicators, life cycle expectations pragmatic Software Metrics, Metrics automation.

Course outcomes:

After completion of this course, a student will be able to:

1. *Gain knowledge on project planning and management, client management and project Scheduling and monitoring.*
2. *Analyze the testing based approach to development, team management and ongoing Project schedule tracking.*
3. *Apply Software Metrics for a given Project to calculate Cost estimation models.*
4. *Communicate effectively with IT-industries or organizations.*
5. *Engage in lifelong learning for effective project management and finance monitoring.*

TEXTBOOKS:

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

REFERENCES:

1. Applied Software Project Management, Andrew Stellman & Jennifer Greene, O'Reilly, 2006
2. Head First PMP, Jennifer Greene & Andrew Stellman, O'Reilly, 2007
3. Software Engineering Project Management, Richard H. Thayer & Edward Yourdon, second edition, Wiley India, 2004.
4. Agile Project Management, Jim Highsmith, Pearson education, 2004.



IV B.Tech I Semester

**15ACS57-DISTRIBUTED SYSTEMS
(CBCC (DEPARTMENT SPECIFIC))**

**L T P C
3 1 0 3**

Course Objectives:

1. To have a broad and up-to-date coverage of the principles and practice in the area of Distributed Systems.
2. To understand the heterogeneous systems such as computers, mobile phones, other devices and Internet) and their functionalities.

UNIT I Basic Concepts

Definition of a distributed systems, Examples, Resource sharing and the Web, Challenges, System models, Architectural and fundamental models, Networking Interprocess communication, External data representation and marshalling, Client-server and Group communication.

UNIT II Communication in Distributed System

System Model – Inter process Communication – the API for internet protocols – External data representation and Multicast communication. Network virtualization: Overlay networks. Case study: MPI Remote Method Invocation And Objects: Remote Invocation – Introduction – Request-reply protocols – Remote procedure call – Remote method invocation. Case study: Java RMI – Group communication – Publish-subscribe systems – Message queues – Shared memory approaches -Distributed objects – Case study: Enterprise Java Beans -from objects to components

UNIT III Peer to Peer Services and File System

Peer-to-peer Systems – Introduction – Napster and its legacy – Peer-to-peer – Middleware – Routing overlays. Overlay case studies: Pastry, Tapestry- Distributed File Systems –Introduction – File service architecture – Andrew File system. File System: Features-File model -File accessing models – File sharing semantics Naming: Identifiers, Addresses, Name Resolution – Name Space Implementation – Name Caches – LDAP.

UNIT IV Synchronization and Replication

Introduction – Clocks, events and process states – Synchronizing physical clocks- Logical time and logical clocks – Global states – Coordination and Agreement – Introduction – Distributed mutual exclusion – Elections – Transactions and Concurrency Control– Transactions -Nested transactions – Locks – Optimistic concurrency control – Timestamp ordering – Atomic Commit protocols -Distributed deadlocks – Replication – Case study – Coda.




UNIT V Process & Resource Management

Process Management: Process Migration: Features, Mechanism – Threads: Models, Issues, Implementation. Resource Management: Introduction- Features of Scheduling Algorithms –Task Assignment Approach.

Course Outcomes:

The primary learning outcome of the course is two-fold:

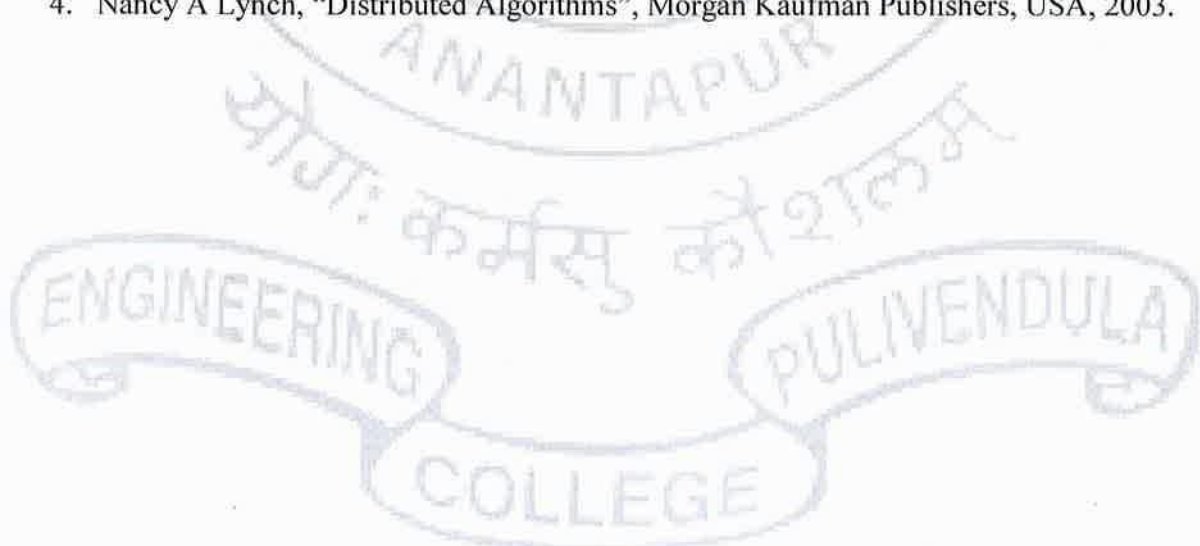
1. Students will identify the core concepts of distributed systems: the way in which several machines orchestrate to correctly solve problems in an efficient, reliable and scalable way.
2. Students will examine how existing systems have applied the concepts of distributed systems in designing

TEXT BOOK:

1. George Coulouris, Jean Dollimore and Tim Kindberg, "Distributed Systems Concepts and Design", Fifth Edition, Pearson Education, 2012.

REFERENCES:

1. Pradeep K Sinha, "Distributed Operating Systems: Concepts and Design", Prentice Hall of India, 2007.
2. Tanenbaum A.S., Van Steen M., "Distributed Systems: Principles and Paradigms", Pearson Education, 2007.
3. Liu M.L., "Distributed Computing, Principles and Applications", Pearson Education, 2004.
4. Nancy A Lynch, "Distributed Algorithms", Morgan Kaufman Publishers, USA, 2003.



IV B.Tech I Semester

**15ACS58-ENTERPRISE APPLICATION SYSTEM
(CBCC (DEPARTMENT SPECIFIC))**

**L T P C
3 1 0 3**

Course Objectives:

1. Describe approaches to enterprise application integration
2. Understand the integration middleware
3. Evaluate the integration approaches suitable for a given problem

UNIT I: Introduction

Requirements for EAI - Challenges in EAI – Integration with legacy systems – Integration with partners - Heterogeneous environment – Implementation approaches – Web services, messaging, ETL, direct data integration – Middleware requirements – Approaches to integration – services oriented and messaging.

UNIT II: Integration Patterns

Introduction to integration patterns – Architecture for application integration – Integration patterns – Point to point, broker, message bus, publish/subscribe, Challenges in performance, security, reliability - Case studies

UNIT III: Service Oriented Integration

Business process integration - Composite applications-services – Web services – Service choreography and orchestration - Business process modeling - BPMN, Business process execution - BPEL – Middleware infrastructure - Case studies

UNIT IV: Messaging Based Integration

Messaging – Synchronous and asynchronous – Message structure – Message oriented middleware – Reliability mechanisms – Challenges – Messaging infrastructure – Java Messaging Services – Case studies

UNIT V: Enterprise Service Bus

Enterprise Service Bus – routing, scalable connectivity, protocol and message transformations, data enrichment, distribution, correlation, monitoring – Deployment configurations – Global ESB, Directly connected, Federated, brokered ESBs – Application server based – Messaging system based – Hardware based ESBs – Support to SOA, message based and event based integrations - Case studies.

Course Outcomes:

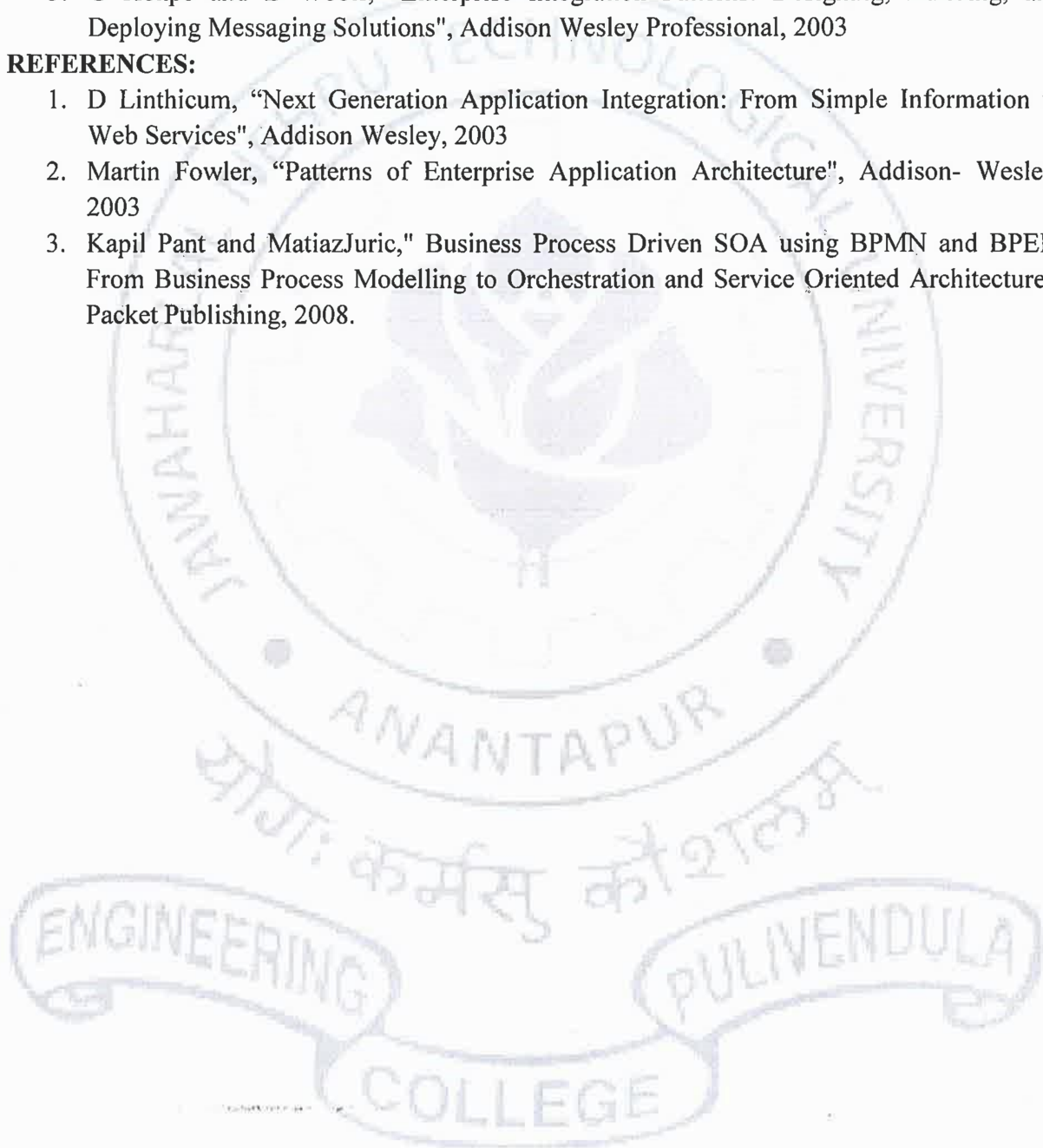
1. Describe different approaches to integration enterprise applications
2. Analyse specifications and identify appropriate integration approaches
3. Develop a suitable integration design for a given problem
4. Identify appropriate integration middleware for a given problem
5. Evaluate the integration approaches against specified requirements

TEXT BOOKS:

1. George Mentzas and Andreas Frezen (Eds), "Semantic Enterprise Application Integration for Business Processes: Service-oriented Frameworks", Business Science Reference, 2009
2. WaseemRoshen,"SOA Based Enterprise Integration", Tata McGraw Hill, 2009.
3. G Hohpe and B Woolf, "Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions", Addison Wesley Professional, 2003

REFERENCES:

1. D Linthicum, "Next Generation Application Integration: From Simple Information to Web Services", Addison Wesley, 2003
2. Martin Fowler, "Patterns of Enterprise Application Architecture", Addison- Wesley, 2003
3. Kapil Pant and MatiazJuric," Business Process Driven SOA using BPMN and BPEL: From Business Process Modelling to Orchestration and Service Oriented Architecture", Packet Publishing, 2008.



IV B.Tech I Semester

**15ACS59-SEMANTIC WEB
(CBCC (DEPARTMENT SPECIFIC))**

**L T P C
3 1 0 3**

Course Objectives

- To learn Web Intelligence
- To learn Knowledge Representation for the Semantic Web
- To learn Ontology Engineering
- To learn Semantic Web Applications, Services and Technology.

UNIT I: Introduction

Introduction to the Syntactic web and Semantic Web – Evolution of the Web – The visual and syntactic web – Levels of Semantics – Metadata for web information - The semantic web architecture and technologies –Contrasting Semantic with Conventional Technologies–Semantic Modeling -Potential of semantic web solutions and challenges of adoption.

UNIT II: Ontological Engineering

Ontologies – Taxonomies –Topic Maps – Classifying Ontologies - Terminological aspects: concepts, terms, relations between them – Complex Objects -Subclasses and Sub-properties definitions –Upper Ontologies – Quality – Uses - Types of terminological resources for – Methods and methodologies for building ontologies – Multilingual Ontologies-Ontology Development process and Life cycle – Methods for Ontology Learning – Ontology Evolution – Versioning.

UNIT III :Structuring and Describing Web Resources

Structured Web Documents - XML – Structuring – Namespaces – Addressing – Querying – Processing - RDF – RDF Data Model – Serialization Formats- RDF Vocabulary –Inferencing- RDFS – basic Idea – Classes – Properties- Utility Properties – RDFS ModellingforCombinations and Patterns- Transitivity.

UNIT IV: Web Ontology Language

OWL – Sub-Languages – Basic Notions -Classes- Defining and Using Properties – Domainand Range – Describing Properties - Data Types – Counting and Sets- Negative PropertyAssertions – Advanced Class Description – Equivalence – Owl Logic.

UNIT V: Semantic Web Tools and Applications

Development Tools for Semantic Web – Jena Framework – SPARL –Querying semantic web - Semantic Desktop – Semantic Wikis -Semantic Web Services – Application in Science – Business

Course Outcomes:

- Understand the concept structure of the semantic web technology and how this technology revolutionizes the World Wide Web and its uses.
- Understand the concepts of metadata, semantics of knowledge and resource, ontology, and their descriptions in XML-based syntax and web ontology language (OWL).
- Describe logic semantics and inference with OWL.
- Use ontology engineering approaches in semantic applications
- Program semantic applications with Java API.

TEXT BOOKS:

1. Liyang Yu, A Developer's Guide to the Semantic Web, Springer; 1st Edition, 2011.
2. John Hebel, Matthew Fisher, Ryan Blace and Andrew Perez-Lopez, Semantic Web Programming, Wiley; 1 edition, 2009.
3. Grigoris Antoniou, Frank van Harmelen, A Semantic Web Primer, Second Edition (Cooperative Information Systems) (Hardcover), MIT Press, 2008.
4. Robert M. Colomb, Ontology and the Semantic Web: Volume 156 Frontiers in Artificial Intelligence and Applications (Frontier in Artificial Intelligence and Applications), IOS Press, 2007.
5. Dean Allemang and James Hendler, Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL, Morgan Kaufmann; 2 edition, 2011.

REFERENCES:

1. Michael C. Daconta, Leo J. Obrst and Kevin T. Smith, The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management, Wiley; 1 edition 2003.
2. Karin Breitman, Marco Antonio Casanova and Walt Truszkowski, Semantic Web: Concepts, Technologies and Applications (NASA Monographs in Systems and Software Engineering), Springer; Softcover, 2010.
3. Vipul Kashyap, Christoph Bussler and Matthew Moran, The Semantic Web: Semantics for Data and Services on the Web (Data-Centric Systems and Applications), Springer, 2008.



[Handwritten signature]

[Handwritten signature]

IV B.Tech I Semester

15ACS60-SOFTWARE TESTING LAB

L T P C

0 03 2

PRE-REQUISITES:

- A Course on “Software Engineering” at UG Level.

Course objectives:

1. To gain knowledge on practical concepts of Testing techniques, Virtualization & Cloud Technologies.
2. Can able to apply the testing techniques in real time.
3. To acquire skills on QTP, Load Runner, Windows Azure, SalesForce, VMware.

LABORATORY EXERCISES:**Testing:**

1. Take any system (e.g. ATM system) and study its system specifications and report the various bugs.
2. Write the test cases for any known application (e.g. Banking application)
3. Create a test plan document for any application (e.g. Library Management System)
4. Test the supplied project/Application through testing tool: Quick Test Professional by generating appropriate test cases.
5. Test the supplied project/Application through testing tool: LoadRunner by generating appropriate test cases.
6. Take any project (e.g. University admission, Placement Portal) and execute it. During the life cycle of the project create the various testing documents* and final test report document.
7. A program written in c language for matrix multiplication fails “Introspect the causes for its failure and write down the possible reasons for its failure”.
8. Study the implementation of Win Runner and generate test report
9. Identify the usage of stubs or drivers in the context of an integration testing scenario
10. Generate test cases using Black box testing technique to Calculate Standard Deduction on Taxable Income. The standard deduction is higher for tax payers who are 65 or older or blind.
Use the method given below to calculate tax.
 - a) The first factor that determines the standard deduction is the filing status. The basic standard deduction for the various filing status are:
 - Single \$4,750
 - Married, filing a joint return \$9,500
 - Married, filing a separate return \$7,000
 - b) If a married couple is filing separate returns and one spouse is not taking standard Deduction, the other spouse also is not eligible for standard deduction.

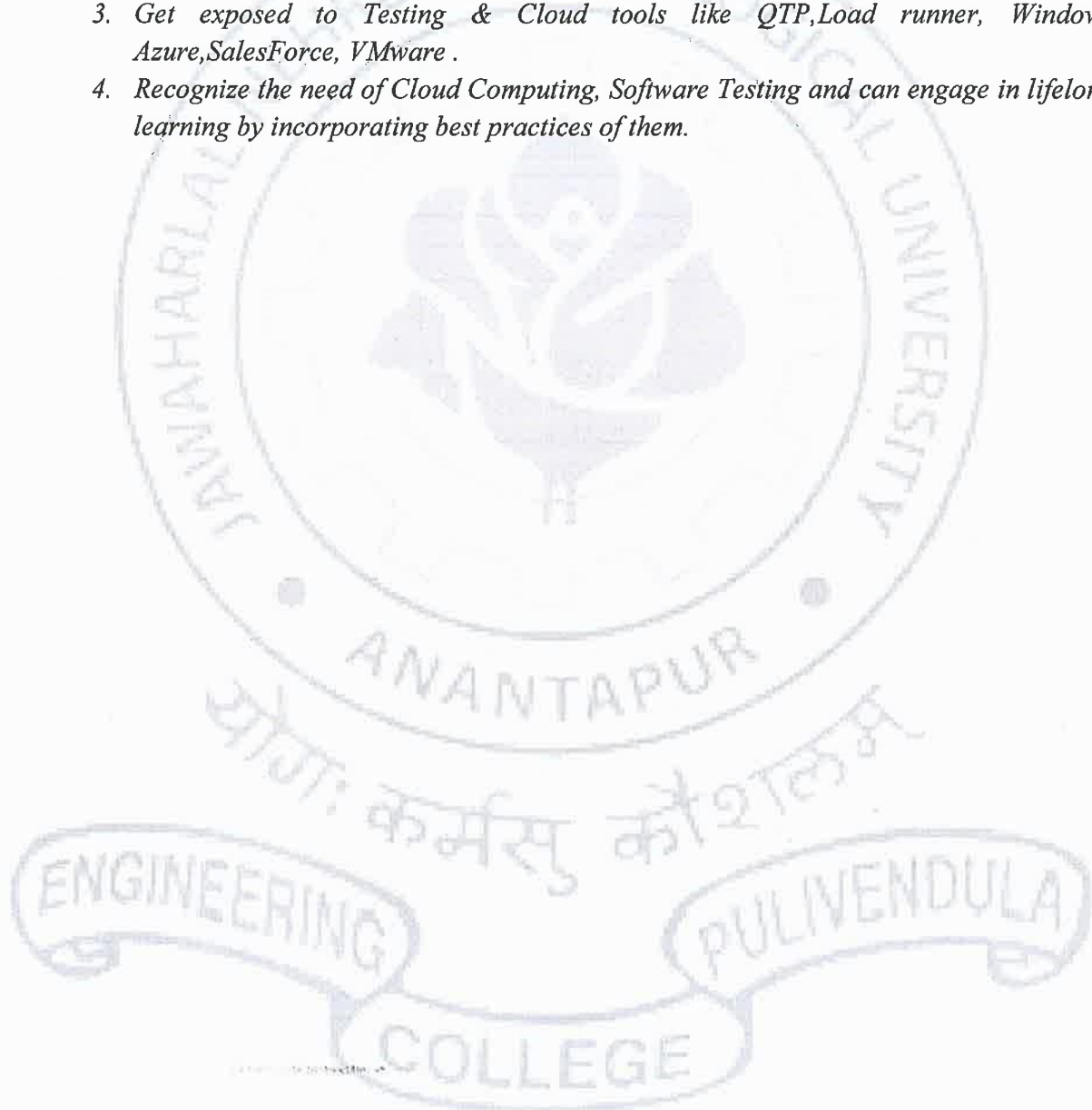


- c) An additional \$1,000 is allowed as standard deduction, if either the filer is 65 yrs or the spouse is 65 yrs or older (the latter case applicable when the filing status is "Married" and filing "joint").

Course outcomes:

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

1. Gain Knowledge on the real time applications of Software Testing & Cloud (PO 1).
2. Analyse the software from the perspective of software tester (PO 2).
3. Get exposed to Testing & Cloud tools like QTP, Load runner, Windows Azure, SalesForce, VMware.
4. Recognize the need of Cloud Computing, Software Testing and can engage in lifelong learning by incorporating best practices of them.



[Handwritten signatures]

B.Tech IV-ISem(R15)

L T P C

0 0 3 2

15ACS61 – CLOUD COMPUTING LAB

1. Illustrate the installation and configuration of aneka cloud.
2. Write a program to print "Hello World" using Aneka Thread Programming model use Single Thread?
3. Write a program to print "Hello World" based on Thread model and use threads also print the executor node information along with the Submission Time and Completion Time?
4. Find procedure to run the virtual machine of different configuration. Check how many virtual machines can be utilized at particular time.
5. Find procedure to attach virtual block to the virtual machine and check whether it holds the data even after the release of the virtual machine.
6. Install a C compiler in the virtual machine and execute a sample program.
7. Show the virtual machine migration based on the certain condition from one node to the other.
8. Find procedure to set up the one node Hadoop cluster.
9. Mount the one node Hadoop cluster using FUSE.
10. Write a program to use the API's of Hadoop to interact with it.

