



# School of Computer Science & IT Devi Ahilya Vishwavidyalaya

## SYLLABUS

**M. Tech (Information Architecture & Software Engineering) 2 years**

### Program Educational Objectives (PEOs)

- PEO 1:** Develop software engineering skills for architecting quality products and carve out successful careers in IT industry and academia.
- PEO 2:** Demonstrate ability to manage software projects and participate effectively in group tasks and initiate entrepreneurship activities.
- PEO 3:** Deliver quality software products and perform research in the domain of software engineering.

### Program Specific Outcomes (PSOs)

- PSO 1:** Acquire expertise to build real-life software applications using agile approach.
- PSO 2:** Design, implement and integrate software with maintainability, reusability, refactoring and design pattern issues in mind.

## II - SEMESTER

## CS-5319: AGILE SOFTWARE ENGINEERING

---

### Course Outcomes (COs):

- CO1:** Interpret the concept of agile software engineering and its advantages in software development.
- CO2:** Analyze the core practices behind several specific agile methodologies.
- CO3:** Identify the roles and responsibilities in agile projects and their difference from projects following traditional methodologies.
- CO4:** Familiarizing the students with basic Agile Estimation & Planning.
- CO5:** Facilitate students in understanding the Tracking Agile Projects, differentiating tools and testing.
- 

### Course Contents

#### UNIT-I

**No. of Hours: 8**

Agile Software Development: The History and Value of Agile Software Development, The Agile Manifesto, Principles of Agile, Organizational Culture Considerations with Agile, What Is Organizational Culture, and Why Does It Matter? The Team Members' Viewpoint, A Manager's Viewpoint, An Executive's Viewpoint, Success and Failures.

#### UNIT-II

**No. of Hours: 8**

Understanding the Different Types of Agile: Introduction to Extreme Programming (XP), Pair Programming, Scrum, Feature-Driven Development, Dynamic Systems Development Method, Lean Software Development and Kanban. Describing the Different Roles: Deep Dive into Scrum Roles, Product Owner, Scrum Master, The Team, Extended Team Members, Roles in Other Methodologies: Project Sponsor, Requirements Gatherer, Agile Coach.

#### UNIT-III

**No. of Hours: 8**

The New Way to Collect and Document Requirements: Old Form of Requirements Gathering, Agile Requirements in Scrum, Enhancing Requirements, From User Stories to Deliverables, Case Study.

#### UNIT-IV

**No. of Hours: 8**

Grooming and Planning: Product Backlog (Scrum and XP), Prioritization of Stories, Estimating, Scrum: Sprint Planning, Extreme Programming: XP Planning Game, Triple Constraints.

#### UNIT-V

**No. of Hours: 8**

Testing, Quality, and Integration: Creating a Quality-Focused Culture, Test-Driven Development (Unit Testing), Refactoring. Tracking and Reporting: Kanban Board, Work in Progress (WIP) Limit, Burn Charts, Feature-Driven Development (FDD) Parking Lots. Meetings or Ceremonies: Daily Stand-Up Meeting, Sprint Review or Demo, Retrospectives.

**Text Books:**

1. Introduction to Agile Methods- Sondra Ashmore, Kristin Runyan, Addison Wesley, 2015.
2. Agile Software Engineering- Orit Hazzan, Yael Dubinsky, Springer, 2008

**Reference Books:**

1. Software Engineering: Concepts & Practices- Ugrasen Suman, Cengage Learning Publications, 2nd Ed., 2022.

**Online Resources:**

# CS-5319: AGILE SOFTWARE ENGINEERING - PRACTICAL

---

## Course Outcomes (COs):

- CO1:** Understand the Agile Manifesto and its principles, and create and manage an Agile Product Backlog.
- CO2:** Able to write User Stories and Acceptance Criteria.
- CO3:** Implement Sprint Planning, Daily Stand-ups, and Retrospectives.
- CO4:** Able to use Tools for Agile Management.
- CO5:** Able to apply Test-Driven Development.
- 

## Course Contents

### UNIT-I

**No. of Hours: 4**

Agile Project Inception & Requirements Analysis

Topic: Project Charter Creation & User Story Development

Focus Areas: Introduction to Agile & Scrum, Project Charter (Scope, Goals, Stakeholders), User Roles & Personas, User Story Writing, Story Point Estimation.

Deliverable: Product Backlog document with user stories & story points.

### UNIT-II

**No. of Hours: 4**

Sprint Planning & Agile Tool Configuration

Topic: Sprint Planning and Kanban Board Setup

Focus Areas: Scrum roles & ceremonies, Sprint Planning, Product Backlog vs Sprint Backlog, Agile tools (Trello / Jira), Kanban workflow

Deliverable: Screenshot of configured board, Sprint Backlog with selected stories.

### UNIT-III

**No. of Hours: 4**

Iterative Development & Agile Communication

Topic: Daily Stand-ups & UI Prototyping

Focus Areas: Daily Stand-up meetings, Incremental development, Agile communication, UI/UX basics & wireframing

Deliverable: Stand-up log (3–5 days), UI mock-up screenshots, Updated Scrum board showing progress.

### UNIT-IV

**No. of Hours: 4**

Agile Testing & Test-Driven Development (TDD)

Topic: Implementing TDD for Checkout Functionality

Focus Areas: Agile testing principles, Test-Driven Development (Red–Green–Refactor), Unit testing, Code refactoring

Deliverable: TDD report including: Test cases, Code snippets, Test results (pass/fail).

**UNIT-V****No. of Hours: 4**

Sprint Review, Retrospective & Agile Metrics

Topic: Sprint Evaluation and Continuous Improvement

Focus Areas: Sprint Review, Sprint Retrospective, Agile metrics (Velocity, Burndown), Continuous improvement

Deliverable: Final Sprint Report including: Completed user stories, Velocity chart, Retrospective summary, Improvements planned for next sprint.

**Text Books:**

1. Introduction to Agile Methods- Sondra Ashmore, Kristin Runyan, Addison Wesley, 2015.
2. Agile Software Engineering- Orit Hazzan, Yael Dubinsky, Springer, 2008.

**Reference Books:**

1. Software Engineering: Concepts & Practices- Ugrasen Suman, Cengage Learning Publications, 2nd Ed., 2022.

**Online Resources:**

## CS-5320: SOFTWARE ARCHITECTURE

---

### Course Outcomes (COs):

**CO1:** Understand the fundamentals of software architecture

**CO2:** Study the various architecture evaluation techniques.

**CO3:** Apply architectural styles in different applications.

**CO4:** Analysis of design patterns.

**CO5:** Apply design patterns in producing quality software.

---

### Course Contents

#### UNIT-I

**No. of Hours: 8**

Basic Concepts, Designing Architectures

Understanding Software Architecture- Concepts, Models, Processes, Stakeholders. The Design Process, Architectural Conception, Refined Experience in Action: Styles and Architectural Patterns, Architectural Conception in Absence of Experience.

#### UNIT-II

**No. of Hours: 8**

Connectors, Modeling

Connectors in Action: A Motivating Example, Connector Foundations, Connector Roles, Connector Types and Their Variation Dimensions, Example Connectors, Modeling Concepts, Ambiguity, Accuracy, and Precision, Complex Modeling: Mixed Content and Multiple Views, Evaluating Modeling Techniques (ATAM, SAAM, CBAM etc.), Specific Modeling Techniques.

#### UNIT-III

**No. of Hours: 8**

Analysis, Implementation and Deployment

Concepts, Existing Frameworks, Software Architecture and Deployment, Software Architecture and Mobility, Pipes and Filters, Event- based, Implicit Invocation, Layered systems, Repositories Interpreters, Process control. Distributed and Networked Architectures, Architectures for Network-Based Applications, Decentralized Architectures. Service-Oriented Architectures and Web Services.

#### UNIT-IV

**No. of Hours: 8**

Introduction to Design Patterns

What Is a Design Pattern? Describing Design Patterns, The Catalog of Design Patterns, Organizing the Catalog, How Design Patterns Solve Design Problems, How to Select a Design Pattern, How to Use a Design Pattern. Design Anti-Patterns.

#### UNIT-V

**No. of Hours: 8**

Applications of Design Patterns

Categories of Design Patterns. Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype, Singleton. Structural Patterns: Adapter, Bridge, Composite, Decorator, Façade,

Flyweight, Proxy. Behavioral Patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, Strategy, Template Method, Visitor.

**Text Books:**

1. Software Architecture in Practice- Len Bass, Paul Clements, Rick Kazman, Pearson, 4th Edition, 2021.
2. Software Architecture: Foundations, Theory, and Practice- Richard N. Taylor, Nenad Medvidovic, Eric Dashofy, 2009.
3. Design Patterns: Elements of Reusable Object Oriented- Erich Gamma et al, Pearson Education, 2015.

**Reference Books:**

1. Software Engineering: Concepts & Practices- Ugrasen Suman, Cengage Learning Publications, 2nd Ed., 2022.
2. Fundamentals of Software Architecture, An Engineering Approach- Mark Richards, Neal Ford, O'Reilly Media 2020.
3. Design Patterns Explained -Alan Shalloway, Pearson Education.

**Online Resources:**

## CS-5320: SOFTWARE ARCHITECTURE-PRACTICAL

---

### Course Outcomes (COs):

**CO1:** Apply architectural styles in various applications.

**CO2:** Implementation of creational patterns.

**CO3:** Implementation of structural patterns.

**CO4:** Implementation of behavioral patterns.

**CO5:** Identify and implement design anti-patterns in various design patterns.

---

### Course Contents

#### UNIT-I

**No. of Hours: 8**

Architectural Styles

Implementation of various monolithic architectural styles: Layered, Pipes and Filters, Event-based, Microkernel, Interpreters, Data-flow, Interpreter, Rule-based etc.

#### UNIT-II

**No. of Hours: 8**

Architectural Styles

Implementation of various distributed architectural styles: Client/ Server, Shared data, Event-based, Service-oriented, Space-based, Microservices

#### UNIT-III

**No. of Hours: 8**

Design Patterns

Implementation of Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype, Singleton.

Implementation of design Anti-Patterns for creational patterns.

#### UNIT-IV

**No. of Hours: 8**

Introduction to Design Patterns

Implementation of Structural Patterns: Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy.

Implementation of design Anti-Patterns for structural patterns.

#### UNIT-V

**No. of Hours: 8**

Applications of Design Patterns

Implementation of Behavioral Patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, Strategy, Template Method, Visitor.

Implementation of design Anti-Patterns for behavioral patterns.

#### Text Books:

1. Software Architecture in Practice- Len Bass, Paul Clements, Rick Kazman, Pearson, 4th Edition, 2021.

2. Software Architecture: Foundations, Theory, and Practice- Richard N. Taylor, Nenad Medvidovic, Eric Dashofy, 2009.
3. Design Patterns: Elements of Reusable Object Oriented- Erich Gamma et al, Pearson Education, 2015.

**Reference Books:**

1. Software Engineering: Concepts & Practices- Ugrasen Suman, Cengage Learning Publications, 2nd Ed., 2022.
2. Fundamentals of Software Architecture, An Engineering Approach- Mark Richards, Neal Ford, O'Reilly Media 2020.
3. Design Patterns Explained -Alan Shalloway, Pearson Education.

**Online Resources:**

## CS-6226: ADVANCED ALGORITHM DESIGN

---

### Course Outcomes (COs):

**CO1:** Understand complexity representation in terms of asymptotic notations.

**CO2:** Apply different algorithm design paradigms such as divide-and-conquer, greedy and dynamic programming for effective problem-solving

**CO3:** Understand and analyse major graph algorithms.

**CO4:** Apply string matching algorithms.

**CO5:** Understand design paradigms such as backtracking, branch & bound and concept of complexity class.

---

### Course Contents

#### UNIT-I

**No. of Hours: 8**

Introduction to Algorithms, Time & Space Complexity, Sorting: Merge Sort, Quick Sort, Heap Sort; Searching: Linear Search, Binary Search, Hashing.

#### UNIT-II

**No. of Hours: 8**

Greedy Techniques & its Applications: Fractional Knapsack, Activity Selection Problem  
Dynamic Programming & its Applications: Fibonacci Computation, 0/1 Knapsack Problem, Longest Common Subsequence Problem, Edit Distance Problem; Matrix Chain Multiplication. Edit distance, All pair shortest paths – Warshal's and Floyd's algorithms.

#### UNIT-III

**No. of Hours: 8**

Graph Algorithms, Traversal: BFS, DFS; Shortest Path: Dijkstra's Shortest Path Algorithm; Computation of Minimum Spanning Trees: Prim Algorithm, Kruskal Algorithm.

#### UNIT-IV

**No. of Hours: 8**

String Matching Algorithms: Naïve Algorithm, KMP Algorithm, Rabin-Karp Algorithm.

#### UNIT-V

**No. of Hours: 8**

Backtracking: n-Queen Problem, Sum of Subset, Branch and Bound: Travelling Salesman Problem, Introduction to P, NP, NP-complete, NP-Hard.

#### Text Books:

1. Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest (2022), Introduction to Algorithms, Fourth edition, Prentice Hall of India.
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran (2008), "Fundamentals of Computer Algorithms", Orient Longman Pvt. Ltd.

#### Reference Books:

1. Gilles Brassard Paul Bratley (1996), "Fundamentals of Algorithms", Prentice Hall.
2. Anany Levitin (2009), "An Introduction to Design and Analysis of Algorithms", Pearson.

**Online Resources:**

1. NPTEL Course Video Lectures on "Design and Analysis of Algorithms" - By Prof. Abhiram G Ranade, Prof. Ajit A Diwan, Prof. Sundar Viswanathan IIT Bombay
2. NPTEL Course Video Lectures on "Design and Analysis of Algorithms" - Prof. Madhavan Mukund, Chennai Mathematical Institute.

## CS-6226: ADVANCED ALGORITHM DESIGN - PRACTICAL

---

### Course Outcomes (COs):

**CO1:** Compare and analyse different data structures and sorting algorithms.

**CO2:** Analyse major graph algorithms.

**CO3:** Implement different algorithm design paradigms such as greedy, dynamic programming, backtracking, branch and bound for effective problem-solving.

**CO4:** Apply string matching algorithms.

**CO5:** Understand design paradigms such as backtracking, branch & bound and concept of complexity class.

---

### Course Contents

#### UNIT-I

**No. of Hours: 8**

Implement the following sorting algorithm for a given set of elements and determine the time required to sort the elements. The elements can be read from a file or can be generated using the random number generator.

1. Quick sort
2. Merge sort
3. Heap Sort
4. Counting Sort

Implement the following Searching algorithm for a given set of elements and determine the time required to sort the elements. The elements can be read from a file or can be generated using the random number generator.

1. Linear Search,
2. Binary Search,
3. Hashing

#### UNIT-II

**No. of Hours: 8**

Write programs to find the optimal solution for the following problems using the Greedy Method

1. Fractional Knapsack
2. Activity Selection Problem.
3. Minimum Spanning trees – Prim's and Kruskal's algorithms.
4. Single-source shortest paths - Dijkstra's algorithms

#### UNIT-III

**No. of Hours: 8**

Write programs to find the optimal solution for the following problems using the Dynamic Programming approach

1. 0/1 Knapsack
2. Longest Common Subsequence (LCS)

3. Matrix Chain Multiplication
4. Edit distance
5. All pair shortest paths – Warshal’s and Floyd’s algorithms

#### **UNIT-IV**

**No. of Hours: 8**

Write programs to print all the nodes reachable from a given starting node in a graph using the following traversal methods

1. Breadth First Search (BFS)
2. Depth First Search (DFS)

Write programs to implement the following string matching algorithms:

1. Naïve Algorithm
2. KMP Algorithm,
3. Rabin-Karp Algorithm

#### **UNIT-V**

**No. of Hours: 6**

Write programs to find the optimal solution for the following problems using the Backtracking approach

1. Graph Coloring
2. Sum of Subset

Write a program to find the optimal solution for the following problems using the Branch and Bound method

1. Travelling Salesman Problem

#### **Text Books:**

1. Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest (2022), Introduction to Algorithms, Fourth edition, Prentice Hall of India.
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran (2008), “Fundamentals of Computer Algorithms”, Orient Longman Pvt. Ltd.

#### **Reference Books:**

#### **Online Resources:**

1. NPTEL Course Video Lectures on “Design and Analysis of Algorithms” - By Prof. Abhiram G Ranade, Prof. Ajit A Diwan, Prof. Sundar Viswanathan IIT Bombay
2. NPTEL Course Video Lectures on “Design and Analysis of Algorithms” - Prof. Madhavan Mukund, Chennai Mathematical Institute.

## CS-6313: SOFTWARE TESTING & QUALITY ASSURANCE

---

### Course Outcomes (COs):

**CO1:** Develop the tester's perspective resulting in software quality improvement

**CO2:** Learn modern techniques of manual and automated testing

**CO3:** Acquire skills of creating, managing and tracking the test cases.

**CO4:** Define and comply with custom specific quality process in the company.

**CO5:** Acquire proficiency in tools for configuration management, test case management and automated testing.

---

### Course Contents

#### UNIT-I

**No. of Hours: 10**

Software Testing: Introduction and background, Big picture of software development process and testing component in every phase of the process, Software testing terms and definitions: Black box & white box testing, static and dynamic testing, unit, integration, system, validation, acceptance, regression testing. Unit test Automation with JUnit, Techniques of black box testing: Preparing tests-to-pass and tests-to-fail, equivalence partitioning, data and state testing, White box testing: formal reviews of the code, programming standards and guidelines, preparing code review checklists, Data & Code coverage techniques

#### UNIT-II

**No. of Hours: 10**

Configuration and Compatibility testing: Isolating configuration bugs, identifying requirements of hardware, software and network, identifying platform and application versions, backward and forward compatibility, Data sharing compatibility, Foreign Language testing: Translation issues, ASCII, EBCDIC, Hotkeys and shortcuts, extended characters, computation on characters, localisation, compatibility and configurability issues, Documentation testing, Types of documentation testing, preparing checklists before documentation testing, Security testing, Website testing: Web page fundamentals.

#### UNIT-III

**No. of Hours: 5**

Automation Testing: Techniques and methods, Seminar on popular tools like Winrunner and Rational ROBO, Hands on experience on these tools.

#### UNIT-IV

**No. of Hours: 5**

Test Planning: Test Phases, Resource, manpower requirements, test strategy, test schedule, bug reporting mechanisms, metrics and statistics writing and tracking test cases, Introduction to automated bug tracking and test case management systems.

#### UNIT-V

**No. of Hours: 10**

Usability: Importance and Impact on SDLC, Generations of User Interfaces, The Usability Engineering Lifecycle, Usability Heuristics, Usability Testing, Usability Assessment Methods beyond Testing, Interface Standards, International User Interfaces. CMM: Process, Need for Process Improvement & Standards, Assessment, Improvement and Compliance against Matured Processes, Software Quality tradeoffs, Introduction: CMM Level I to V, Case Studies.

**Text Books:**

1. Software testing Ron Patton SAMS Publishing
2. CMM Level 5 unleashed

**Reference Books:**

1. Effective methods of software testing' Wiley Publishers, William E. Perry

**Online Resources:**

## CS-6313: SOFTWARE TESTING & QUALITY ASSURANCE - PRACTICAL

---

### Course Outcomes (COs):

**CO1:** Develop the tester's perspective resulting in software quality improvement

**CO2:** Learn modern techniques of manual and automated testing

**CO3:** Acquire skills of creating, managing and tracking the test cases.

**CO4:** Define and comply with custom specific quality process in the company.

**CO5:** Acquire proficiency in tools for configuration management, test case management and automated testing.

---

### Course Contents

#### UNIT-I No. of Hours: 8

1. To Test a basic data structure programs
2. To Test a Sample Website Using LoadRunner

#### UNIT-II No. of Hours: 8

3. To Perform Automation Testing on a Sample Website using Selenium and JMeter
4. To Test a Sample Android Application Using Bitbar

#### UNIT-III No. of Hours: 4

5. To Perform Functional Testing and Usability Testing on a Website

#### UNIT-IV No. of Hours: 2

6. To Perform Automation Testing on a Desktop Application

#### UNIT-V No. of Hours: 2

7. To Manage Defects using a Defect Management Tool

#### Text Books:

1. Software testing Ron Patton SAMS Publishing
2. CMM Level 5 unleashed

#### Reference Books:

1. Effective methods of software testing' Wiley Publishers, William E. Perry

#### Online Resources:

## CS-6518: CLOUD COMPUTING

---

### Course Outcomes (COs):

- CO1:** Understand the concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing
- CO2:** Understand the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, multicloud.
- CO3:** : Developing In-depth Knowledge of Cloud Service Models and Virtualization and design cloud computing solutions according to the application domain.
- CO4:** Implement IaaS, SaaS, PaaS through AWS, Google Cloud, Salesforce and other service providers.
- CO5:** Learn the core issues and latest trends and technologies of cloud computing.
- 

### Course Contents

#### UNIT-I

**No. of Hours: 10**

Cloud Computing definition, Types of cloud, Evolution of Cloud Computing, Applications cloud computing, Cluster Computing, Major Players in Cloud Computing, Issues and challenges in Cloud, Cloud stakeholders, SLAs, Economics of the Cloud  
Cloud Models: Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud. Advantages, disadvantages and applications of the deployment models.

#### UNIT-II

**No. of Hours: 8**

Basics of Virtualization, Types of Virtualization, Virtualization using virtualbox, Virtualization Tools and Mechanisms, Creating virtual machines in AWS, Virtualization for Data-center Automation, Scalability, Load balancing, Server Management, Fault Tolerance.

#### UNIT-III

**No. of Hours: 12**

Types of Cloud services: Infrastructure as a Service: Compute Services, clusters, HPC, Data Storage services and its categories- File storage, Block storage, Object storage, applications utilizing cloud storage, Network Services.  
Platform as a Service: Understand and develop application using PAAS. Software as a Service: Applications, working, development methodology, advantages and disadvantages.

#### UNIT-IV

**No. of Hours: 8**

Database as a Service, Functions as a service-Serverless Computing, Microservices Introduction to MapReduce, HDFS, Hadoop Framework. Devops, Containers, Kubernetes.

#### UNIT-V

**No. of Hours: 8**

Security issues in cloud, AI and Machine Learning in Cloud, Fog Computing, Edge Computing, Green Cloud.  
Case Studies:

Hypervisors – Xen, KVM , VMWare, Virtual Box, Hyper-V.

Cloud Service Providers- AWS, Microsoft Azure, Github, Google Workspace, Salesforce

**Text Books:**

1. Buyya, Rajkumar, Christian Vecchiola, and S. Thamarai Selvi. Mastering cloud computing: foundations and applications programming. Newnes, 2013.
2. 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

**Reference Books:**

1. Cloud Computing “A Practical Approach” Anthony T. Velte, Toby J. Velte, Robert Elsenpeter. McGraw-Hill.
2. Kai Hwang, Geoffrey C Fox, Jack G Dongarra, “Distributed and Cloud Computing, From Parallel Processing to the Internet of Things”, Morgan Kaufmann Publishers, 2012.
3. John W.Rittinghouse and James F.Ransome, “Cloud Computing: Implementation, Management, and Security”, CRC Press, 2010.
4. Toby Velte, Anthony Velte, Robert Elsenpeter, “Cloud Computing, A Practical Approach”, TMH, 2009.
5. Kumar Saurabh, “ Cloud Computing – insights into New -Era Infrastructure”, Wiley India,2011.

**Online Resources:**

1. <https://www.mygreatlearning.com/cloud-computing/free-courses>
2. <https://aws.amazon.com/training/digital/>
3. <https://www.simplilearn.com/introduction-to-cloud-computing-basics-skillup>
4. <https://www.geeksforgeeks.org/>
5. <https://www.javatpoint.com/>

## CS-6518: CLOUD COMPUTING - PRACTICAL

---

### Course Outcomes (COs):

- CO1:** Understand the concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing.
- CO2:** Understand the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, multicloud.
- CO3:** Developing In-depth Knowledge of Cloud Service Models and Virtualization and design cloud computing solutions according to the application domain.
- CO4:** Implement IaaS, SaaS, PaaS through AWS, Google Cloud, Salesforce and other service providers.
- CO5:** Learn the core issues and latest trends and technologies of cloud computing.
- 

### Course Contents

#### UNIT-I

**No. of Hours: 6**

1. Install Virtualbox/VMware Workstation with different flavor of linux or windows OS on top of windows.
2. Install a C compiler in the virtual machine created using virtual box and execute Simple Programs.
3. Create a free account in AWS and create a virtual machine, configure it and run program on VM.

#### UNIT-II

**No. of Hours: 4**

1. Explore data storage, server management, cloud watch and load balancing tools of AWS. Attach data storage to VM.
2. Create account and explore the applications of GitHub.
3. Explore cloud platforms for Web Application development and Mobile App development.

#### UNIT-III

**No. of Hours: 4**

1. Create a website using Wordpress and host it on cloud.
2. Explore Google workspace.
3. Create a free account in Salesforce (if available) and explore its functionalities.

#### UNIT-IV

**No. of Hours: 6**

1. Develop a SAAS application in team and deploy it on cloud.
2. Explore microservices and serverless computing.
3. Understand DevOps concepts and implementation details. Write complete note on steps required for developing CI/CD pipelines.

#### UNIT-V

**No. of Hours: 4**

1. Explore security solutions in cloud.
2. Install Hadoop single node cluster and run simple applications like wordcount.
3. Study scalability through containerized applications and kubernetes.

**Text Books:**

1. Buyya, Rajkumar, Christian Vecchiola, and S. Thamarai Selvi. Mastering cloud computing: foundations and applications programming. Newnes, 2013.
2. 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

**Reference Books:**

1. Cloud Computing "A Practical Approach" Anthony T. Velte, Toby J. Velte, Robert Elsenpeter. McGraw-Hill.
2. Kai Hwang, Geoffrey C Fox, Jack G Dongarra, "Distributed and Cloud Computing, From Parallel Processing to the Internet of Things", Morgan Kaufmann Publishers, 2012.
3. John W.Rittinghouse and James F.Ransome, "Cloud Computing: Implementation, Management, and Security", CRC Press, 2010.
4. Toby Velte, Anthony Velte, Robert Elsenpeter, "Cloud Computing, A Practical Approach", TMH, 2009.
5. Kumar Saurabh, " Cloud Computing – insights into New -Era Infrastructure", Wiley India,2011.

**Online Resources:**

1. <https://www.mygreatlearning.com/cloud-computing/free-courses>
2. <https://aws.amazon.com/training/digital/>
3. <https://www.simplilearn.com/introduction-to-cloud-computing-basics-skillup>
4. <https://www.geeksforgeeks.org/>
5. <https://www.javatpoint.com/>