Teaching Guidelines for

Software Development Methodologies

PG-DAC September 2022

Duration: **80 hours** (38 theory hours + 34 lab hours + 8 revision/practice hours)

Objective: To build knowledge of Software development methodologies.

Evaluation: 100 marks
Weightage: Theory exam – 40%, Lab exam – 40%, Internals – 20%

Text Book:

- Software Engineering by Chandramouli / Pearson

References:

- Software engineering by Ian Sommerville / Pearson
- Clean Code: A Handbook of Agile Software Craftsmanship by Robert C. Martin / Prentice Hall
- The Mythical Man-Month: Essays on Software Engineering by Frederick P. Brooks Jr. / Addison Wesley
- User Stories Applied: For Agile Software Development by Mike Cohn / Addison Wesley
- DevOps: Continuous Delivery, Integration, and Deployment with DevOps by Sricharan Vadapalli / Packt
- Git for Teams by Emma Westby / O'Reilly

(Note: Each Session is of 2 hours)

Git (8 hours)

Sessions 1 & 2
Lecture

- Developing an application in a team
- Issues developers face when working in a team
- Introduction to code versioning system
- History of code versioning system
  - Different tools available for versioning
  - Software development workflow
- Introduction to git
- Introduction to git repository and git structure
- Adding code to git
- Creating and merging different git branches

Lab

- Create a local git repository
- Commit the initial code
- Update the code
- Use git commands to
  - Get the updated files
  - List the changes
  - Create branch
  - Merge branch

Software Engineering (18 hours)

Sessions 3, 4 & 5
Lecture

- Introduction to software engineering
  - Software Process
  - Software Process Model
  - Software Product
- Importance of Software engineering
- Software Development Life Cycles
- Requirements Engineering
  - Types of Requirements
  - Steps involved in Requirements Engineering
  - Requirement Analysis Modelling
- Design and Architectural Engineering
  - Characteristics of Good Design
  - Function Oriented vs Object Oriented System
  - Modularity, Cohesion, Coupling, Layering
  - Design Models
  - UML
- Coding
  - Programming Principles
  - Coding Conventions
- Object Oriented Analysis and Design

Lab (4 hours)

- Prepare software requirement specification for the final project
- Create the initial use-cases, activity diagram and ER diagram for the final project

Sessions 6 & 7
Lecture

- Introduction to Agile development model
- Agile development components
- Benefits of Agile
- Introduction to different tools used for agile web development
- Scrum and Extreme Programming
- Introduction to Atlassian Jira
  - Add Project
  - Add Tasks and sub-tasks
  - Create sprints with tasks
- Case study of developing web application using agile methodology

Lab

- Create different sprints in Atlassian Jira for different features

DevOps (20 hours)

Sessions 8 & 9
Lecture

- Introduction to Microservices
- Microservices Architecture
- Fragmentation of business requirement
- Deployment pattern

- API gateway
- Service Discovery
- Database Management for Microservices

Lab

- Create Microservices

Sessions 10 & 11
Lecture

- Introduction to DevOps
- DevOps ecosystem
- DevOps phases
- Introduction to containerisation
- Introduction to docker
- Creating docker images using Dockerfile
- Container life cycle

Lab

- Install and configure docker
- Create docker image using Dockerfile
- Start docker container
- Connect to docker container
- Copy the website code to the container
- Use docker management commands to
  - List the images
  - List the containers
  - Start and stop container
  - Remove container and image

Session 12
Lecture

- Introduction to YAML
- Introduction to Docker Swarm and Docker Stack
- Introduction to Kubernetes
- Creating Kubernetes cluster
- Creating service in Kubernetes
- Deploying an application using dashboard

Lab

- Configure Kubernetes
- Configure Kubernetes Dashboard
- Setup a Kubernetes cluster
- Access application using Kubernetes service
- Deploy the website using Dashboard

Testing & Integration (18 hours)

Session 13
Lecture

- Introduction to software testing
- Why testing code is important
- Verification and validation
- Quality Assurance vs Quality Control vs Testing

- Principles of software testing

Assignment

- Read more testing concepts used in the industry

Session 14

Lecture

- Introduction to STLC and V Model
- Types of testing: manual and automation
- Tools used for automation testing
- Introduction to testing methods: white-box, black-box and grey-box
- Introduction to functional testing: (* students are supposed to learn the concepts)
- Introduction to non-functional testing: (* students are supposed to learn theconcepts)

Lab

- Create a test plan for project
- Document the use cases
- Create test case document for different sprints (designed in SE)

Sessions 15 & 16

Lecture

- Introduction to Selenium (use Eclipse IDE)
- Load web driver
- Create selense commands: locators: by ID, name, class, tag name, XPath
- Add interactions: text box, radio button selection, check box selection, drop down item selection, keyboard actions, mouse actions, multi select

Lab

- Download and configure Selenium
- Create a test suite
- Add commands and interactions

Session 17

Lecture

- Introduction to delivery pipeline
- Introduction to Jenkins
- Jenkins management
- Adding slave node to Jenkins
- Building a delivery pipeline
- Selenium integration with Jenkins

Lab

- Install and configure Jenkins
- Build a pipeline job using Jenkins
- Create a maven project for Selenium
- Add Selenium test suite in the project
- Integrate it with Jenkins

Cloud (8 hours)

Session 18

Lecture

- Introduction to Cloud
- Introduction to Virtualization
- Virtualization types: type1, type2

- Cloud Computing, Cloud SPI Model, Cloud Computing Types (Public, Private and Hybrid), Cloud Security (SLA and IAM).
- Virtualization, Hardware Virtualization, Para-Virtualization, Cloning, Snapshot and Template
- Containerization, Operating System Virtualization

Lab

- Create and configure VM using VBox
- Deploy code on VM

Session 19

Lecture

- Cloud architecture
- Service models: IaaS, PaaS, SaaS
- Deployment models: Private, Public, Hybrid
- Services provided by Cloud (Compute, Database, Developer Tools, Storage, Media,
- Mobile, Web, Security, Integration etc.)
- Cloud development best practices
- Introduction to AWS
- Services provided by AWS: EC2, Lambda, S3

Lab

- Create AWS EC2 instance
    - Add Storage, Tag Instance, Review Instance Launch
    - Set up an Apache web server on your EC2 instance
    - Clean up your EC2 Instance