Teaching Guidelines for

Concepts of Programming, Operating System & Software Engineering

Diploma in Advanced Computing (e-DAC)

May 2021

Duration: 36 theory hours + 28 lab hours (**64 hours**)

Evaluation: 100 Marks
Weightage: Theory exam – 40%, Lab exam – 40%, Internal exam – 20%

Part I - Basic Programming Concepts

Duration: 10 theory hours + 10 lab hours (**20 hours**)

Objective: To introduce the fundamental programming concepts in Java.

Prerequisites: Knowledge of computer fundamentals

Text Book:

- Core and Advanced Java Black Book / Dreamtech Press

References:

- Java The Complete Reference by Herbert Schildt / McGraw Hill
- Core Java : Fundamentals - Volume 1 Gary Cornell, Cay S. Horstmann/ Pearson
- Programming in Java by Sachin Malhotra, Saurabh Choudhary / Oxford University Press

(Note: Each Session is of 2 hours)

Session 1: Getting Started

Lecture:

- Setup development environment (JRE, JDK, eclipse)
- Writing your first Java program
- About main () method
- Constructor in Java

Lab:

Write Java programs to:

- Print Hello World
- Add two numbers/binary numbers/characters
- Calculate compound interest
- Calculate power of a number
- Swap two numbers

Session 2: Object Oriented Concepts

Lecture:

- Class & Object
- Access Specifier
- Java Data Types, Primitives and Binary Literals

Lab:

Write Java programs to:

- Calculate area of rectangle
- Calculate area and circumference of circle using multiple classes
- Java program to find ASCII value of a character

Session 3: Operators

Lecture:

- Arithmetic Operator
- Relational Operator
- Logical Operator
- Unary Operator
- Ternary Operator
- Assignment Operator

Session 4: Conditional and Looping Statements

Lecture:

- If, else if, switch
- break & continue keyword
- for loop
- while loop
- do while loop
- static & final keyword
- Recursion

Lab:

Write Java programs to:

- Display prime numbers between 1 and 100 or 1 and n
- Swap two variables without using the third variable
- Find the factorial of a number
- Check if a number is palindrome or not
- Print Fibonacci series till n
- Add two integer variables in 5 different ways using functions and control statement
- Find square root of a number without sqrt method
- Check Armstrong number
- Calculate grades of students using their marks
- Use switch case, recursion, print patterns, etc.

Session 5: Arrays

Lecture:

- Initializing an Array in Java
- Two dimensional array in java
- Java Variable Arguments explained
- Add, update, read array elements
- Sorting and searching in array
- Java String Array to String
- How to copy arrays in Java

Lab:

Write Java programs to:

- Calculate average of numbers using Array
- Reverse an array

- Sort an array in ascending order
- Convert char Array to String
- Add two Matrix using Multi-dimensional Arrays
- Sort strings in alphabetical order
- Find out the highest and second highest numbers in an array
- Concatenate two arrays

Part II - Operating System Concepts

Duration: 16 theory hours + 8 lab hours (**24 hours**)

Objective: To introduce Operating System concepts with Linux environment, and to learn Shell Programming

Prerequisites: Basic Knowledge of programming with object oriented concepts

Text Books:

- Operating Systems Principles by Abraham Silberschatz, Peter Galvin & Greg Gagne / Wiley

References:

- Modern Operating Systems by Andrew Tanenbaum & Herbert Bos/ Pearson
- Principles of Operating Systems by Naresh Chauhan / Oxford University Press
- Beginning Linux Programming by Neil Matthew & Richard Stones / Wrox
- Operating System : A Design-Oriented Approach by Charles Crowley / McGraw Hill

(Note: Each Session is of 2 hours)

Sessions 1 & 2:
Lecture:
Introduction to OS

- Evolution and components of Operating System
- Different from other application software
- Functionality and Services of Operating System
- Types of Operating System

Introduction to Linux

- Basics of File System types
- Commands associated with files/directories;
- Permissions (chmod,chown, etc)
- access control list

File Management

- Attributes and Operations on File Management
- File Access Methods
- Directory Structure

Lab:
Use various commands in Linux system: ls, cp, mv, lpr, sort, grep, cat, tac, more, head, tail, man, whatis, whereis, locate, find, diff, file, rm, mkdir, rmdir, cd, pwd, ln and ln –s, gzip, zip and unzip, tar and its variants, cal, bc, date, time, wc, touch, echo, who, finger, w, whoami, alias, unalias, touch, push, pop, jobs, ps, etc.

Session 3: Shell Programming

Lecture:

- Types of shells in Linux
- Shell Variables and Wild Card symbols
- Shell Meta characters
- Command line arguments
    - Read, Echo, decision loops, arithmetic expressions;

Lab:

Practice scripting on:

- Command line arguments
- Arithmetic in shell scripts
- Read and echo commands in shell scripts
- Taking decisions: if-then-fi , if-then-else-fi, case control structure

Session 4: Process

Lecture:

- Process States
- Preemptive and non-preemptive processes
- Process life cycle

Lab:

Create new system process using fork system call
Implement zombie and orphan processes

Sessions 5 & 6: Process scheduling algorithms with examples

Lecture:

- FCFS
- RR
- Shortest Job First
- Priority

Session 7: Threads

Lecture:

- Types of Threads - user and kernel threads
- Difference between Threads and Process

Session 8: Concurrency Control

Lecture:

- Deadlock Handling Strategies
- Deadlock Prevention
- Deadlock Avoidance

Part III - Software Engineering Concepts

Duration: 10 theory hours + 10 lab hours (**20 hours**)

Objective: To build knowledge of software development methodologies.

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
- User Stories Applied: For Agile Software Development by Mike Cohn / Addison Wesley

---

(Note: Each Session is of 2 hours)

Session 1:
Lecture

- Developing an application in a team
- Issues developers face when working in a team
- Introduction to code versioning system
- 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

Session 2:
Lecture

- Introduction to software engineering
- Software Development Life Cycles
- Requirements Engineering
- Design and Architectural Engineering
  - Design Models
  - UML
- Object Oriented Analysis and Design

Lab

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

Session 3:
Lecture

- Introduction to Agile development model
- Agile development components
- Benefits of Agile model
- Introduction to different tools used for agile web development
- 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

Session 4:
Lecture

- Introduction to software testing
- Principles of software testing
- Verification and validation
- Quality Assurance vs Quality Control vs Testing
- 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 and non-functional testing

Lab

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

Session 5:
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