

[]: [Advance Programming]

Programme: B.Tech (CSE)
Course: Core

Year: II
Credits: 4

Semester:III
Hours: 3

Course Context and Overview:

This course teaches in major object-oriented programming skills essential for software development using Java programming language as a vehicle. Participants learn the fundamental concepts of object-oriented software engineering and development, such as Object and classes, abstraction, inheritance and polymorphism. This course offers a balanced treatment of OOP theory and practice for developing secure and robust codes (Using exception handling) with debugging strategies and unit testing. The advance topics like Collection Framework, Multithreading, and JDBC.

Prerequisites Courses: Computer Programming, Data Structure

Course Outcomes (COs):

| On completion of this course, the students will have the ability to: |
|--|
| CO1: Know and use basic programming constructs for object-oriented problem solving (e.g., classes, polymorphism, inheritance, interfaces) |
| CO2: Learn to go beyond basic programming skills by employing generics, error and exception handling and testing. |
| CO3: Appreciate the role of algorithms and data structures in problem solving and software design (e.g., objected-oriented design, lists, files, searching and sorting) |
| CO4: Design and implement the software solution for a given problem using Object Oriented Programming Paradigm. |

Course Topics:

| Topics | Lecture Hours | |
|--|---------------|---|
| UNIT – I A. Software Development Process | | |
| 1.1 Stake holders such as programmer, System Analyst, System Designer, Tester, project manager, Engineer Architect. | 1 | 4 |
| 1.2 System development Requirements | | |
| 1.3 Feasibility Study | 1 | |
| 1.4 System Design (Low Level and High Level) | | |
| 1.5 Coding and Integration (Top down and Bottom up approach) | 1 | |
| 1.6 Debugging and Testing | | |
| 1.7 Implementation and post implementation | 1 | |
| A case Study to explain the Software development process | | |
| B. Programming Paradigms | | 5 |

| | | |
|--|---|---|
| 1.8 Functional Programming 1.9 Object Oriented Programming | 2 | |
| C. Principles of Object Oriented Programming and problem solving | | |
| 1.10 Classes and Objects. 1.11 Link and Associations. 1.12 Polymorphism (Function binding and overriding) 1.13 Class hierarchy (Inheritance) 1.14 Abstraction and Data hiding 1.15 Unified Modeling Language(UML) Class Diagrams | 3 | |
| Unit-II A. OOP Language Basics | | |
| 2.1 Data types, variables, scope and life time of variables, operators, and expressions. | 1 | |
| 1.2 Control statements, type conversion and costing, simple java program. | 1 | 5 |
| 1.3 Concepts of classes, objects, constructors, methods, access control. | 2 | |
| 2.4 this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, string handling. | 2 | |
| Lab on Classes and Objects | 2 | 4 |
| B. Inheritance | | |
| 2.5 Hierarchical abstractions, Base class object, subclass, subtype, forms of inheritance- specialization, specification, construction, extension, limitation. | 1 | 4 |
| 2.6 Benefits of inheritance, costs of inheritance. Member access rules, super uses, using final with inheritance. | 1 | |
| 2.7 Run time polymorphism- method overriding, abstract classes. | 1 | |
| A case Study to explain the Inheritance Concept | 1 | |
| Lab on Inheritance | 1 | 2 |
| Unit- III A. Package and Interfaces | | |
| 3.1 Defining, Creating and Accessing a Package, Understanding CLASSPATH. | 1 | 4 |
| 3.2 Importing packages, differences between classes and interfaces, defining an interface, implementing interface. | 1 | |
| 3.3 Applying interfaces, variables in interface and extending interfaces. | 1 | |
| 2.4 Exploring packages | 1 | |
| Lab on packages and Interfaces | 2 | 2 |
| B. Exception handling (Secure and Robust coding) | | |
| 3.5 Concepts of exception handling, benefits of exception handling, exception hierarchy. | 1 | 4 |
| 3.6 Usage of try, catch, throw, throws and finally, built in exceptions. | 1 | |
| 3.7 Creating own exception sub classes, nested try. | 1 | |

| | | |
|--|---|---|
| A case Study to explain the Exception handling Concept | 1 | |
| Lab on Exception Handling | 1 | 2 |
| Unit –IV A. Multithreading | | |
| 4.1 Differences between multi threading and multitasking, thread life cycle. | 1 | 4 |
| 4.2 Creating threads, synchronizing threads, daemon threads. | 1 | |
| 4.3 Thread Priority, killing threads | 1 | |
| An example based on Multithreading | 1 | |
| B. Abstract data types , data structures, and algorithms | | |
| 4.4 ArrayList, LinkedList, Stack, HashTable etc. | 2 | 6 |
| 4.5 Set, HashSet, HashMap etc. | 1 | |
| 4.6 Wrapper classes- Number, Integer, Character, Float, Double, Byte etc. | 2 | |
| A case Study to explain the Collection Framework concept | 1 | |
| Lab on Data structures | 1 | 2 |
| C. Data Base Connectivity | | |
| 4.7 Introduction to JDBC, creating connection, registering DBMS driver. | 2 | 4 |
| 4.8 Types of JDBC drivers, executing queries. | 2 | |
| Lab on Database Connectivity | 1 | 2 |

Textbook references (IEEE format):**Text Book:**

- Herbert Schildt. , The Complete Reference JAVA2, Fifth Edition, McGRAW-HILL.

Reference books:

- Timothy C Lethbridge and Robert Laganier, Object Oriented Software Engineering, TATA McGRAW-HIL Edition.

Evaluation Methods:

| Item | Weightage |
|---------|-----------|
| Quiz1 | 20 |
| Quiz2 | |
| Midterm | 30 |
| Endterm | 50 |

Prepared By: Mukesh K Jadon (Consultation with Mr. Nirmal and Dr. Padmanabh)

Date: August 1, 2017