Department: Computer Science and Engineering	***Course Type: Professional Core - Integrated***
***Course Title**: OOP WITH JAVA*	***Course Code: 21CS35***
L-T-P: 3-0-2	***Credits: 4***
Total Contact Hours: 48Hours	***Duration of SEE: 3Hours***
***SEE Marks**: 50*	***CIE Marks: 50***

COURSE DESCRIPTION

This course provides an in-depth knowledge of Object-Oriented application development using Java programming language. It discusses the programming concepts like multithreading, Interfaces, Exception Handling, GUI development and event Handling in JavaFX and, the use of the Java's Collection framework to solve real-world computing problems.

PREREQUISITES

- Student should have the prior knowledge of C

COURSE OBJECTIVES

- To understand and apply the basic Object-Oriented features of Java.
- To understand and apply the concept of Inheritance, Packages and Interfaces
- To understand and apply Exception Handling and, develop Multithreaded Java Applications.
- To Understand and use Wrapper classes, autoboxing, unboxing and Java Collection Framework.
- To develop GUI using JavaFX.

COURSE CONTENTS

UNIT - I **10Hours**

An Overview of Java --Object-Oriented Programming, Two Paradigms, Abstraction, The Three OOP Principles, A First Simple Program, Entering the Program, Compiling the Program, A Closer Look at the First Sample Program

Introducing Classes: Class Fundamentals, Declaring Objects, A Closer Look at new, Assigning Object Reference Variables, Introducing Methods, Constructors, Parameterized Constructors, The this Keyword, Instance Variable Hiding, Garbage Collection, The finalize( ) Method, A Stack Class.

A Closer Look at Methods and Classes: Overloading Methods, Overloading Constructors, Using Objects as Parameters, A Closer Look at Argument Passing, Returning Objects, Introducing Access Control, Understanding static, Introducing final, Arrays, Exploring the String Class, Using Command-Line Arguments, Varargs: Variable-Length Arguments.

UNIT - II **09 Hours**

Inheritance: Inheritance Basics, Using super, Creating a Multilevel Hierarchy, When Constructors Are Executed, Method Overriding, Dynamic Method Dispatch, Why Overridden Methods? Using Abstract Classes, Using final with Inheritance, The Object Class.

Packages and Interfaces: Packages, Access Protection, Importing Packages, Interfaces, Defining an Interfaces, Default Interface Methods, Use static Methods in an Interface.

UNIT - III **10Hours**

Exception Handling: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, multiple catch Clauses, Nested try Statements, throw, throws, finally. Exception Handling: Java's Built-in exceptions, Creating Your Own Exception Subclasses, Chained Exceptions.

Multi-Threaded Programming:

The Java thread model, The main thread, Creating thread, creating multiple threads, Thread priorities, Synchronization. Interthread Communication.

UNIT - IV **10 Hours**

Type Wrappers: Character, Boolean, Numeric type wrappers. Autoboxing: Autoboxing and Methods, Autoboxing / Unboxing occurs in expressions, Autoboxing/Unboxing Boolean and Character values, Autoboxing / Unboxing helps prevents errors.

The Collections Framework: Collections Overview, The Collection Interfaces: the collection interface, the List interface, the Set interface. The Collection Classes: The ArrayList Class, The LinkedList Class, The HashSet Class. Accessing a Collection via an Iterator -Using an Iterator, The For-Each Alternative to Iterators.

UNIT - V **09 Hours**

GUI Programming and Event Handling: Introducing JavaFx GUI Programming, Exploring JavaFx controls, Introducing JavaFX Menus

Event Handling: Event handling mechanisms, The Delegation Event model.

Lab Program List

1. Create a class namely Account with the data members(Accno : integer, name :String, Phone_No: integer, balance_amt:float), and following methods :

a. CreateAccount() method to create an account.
b. Deposit() method to deposit amount to an account.
c. Withdraw() method which gets the amount to be withdrawn from his/her account.
d. PrintAccount() method to display account details.

2. All the banks operating in India are controlled by RBI. RBI has set a well defined guidelines (e.g. minimum interest rate, minimum balance allowed, maximum withdrawal limit etc) which all banks must follow. For example, suppose RBI has set minimum interest rate applicable to a saving bank account to be 4% annually; however, banks are free to use 4%

interest rate or to set any rates above it. Write a JAVA program to implement bank functionality in the above scenario and demonstrate the dynamic polymorphism concept. Note: Create few classes namely Customer, Account, RBI (Base Class) and few derived classes (SBI, ICICI, PNB etc). Assume and implement required member variables and functions in each class.

3. Write a Java Program that does the following related to Inheritance:
   a. Create an abstract class called Vehicle which contains the year_of_manufacture‟ data memberand two abstract methods "getData()‟ and "putData()‟ with a constructor.
   b. Create two derived classes "TwoWheeler" and "FourWheeler" and implement the abstract methods. Make "FourWheeler" as final class.
   c. Create class "MyTwoWheeler‟ which is a sub-class of "TwoWheeler" and demonstrate the use of super keyword to initialize data members of "MyTwoWheeler".
4. Write a Java program to create an abstract class namely Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.
5. Write a program to add additional functionals like mod() and squrt() for performing modulus and squreroot operation to an existing class called "Calculator" which performs basic functions like add(), sub(), mul() and div().
   a. Initialize the class's data members by using the parameterized constructor.
   b. Define an overridden method called displayResults() to display the result of mathematical operation.
   c. Demonstrate the use of reusability by extending the existing class.
   d. Demonstrate the concept of Dynamic Method Dispatch (or run time polymorphism)
6. Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
7. Write a Java program that correctly implements the producer – consumer problem using the concept of inter-thread communication
8. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.

9. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.
10. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in selected color. Initially, there is no message shown.
11. Write a program to create an ArrayList with the name arrlist and type "String". Add strings "AAA","BBB","CCC","DDD","EEE" to the arrlist. Add another string "XXX" to first position, remove the string "AAA", update "BBB" to "AAA" and display the arrlist.
12. Develop a Java program to create a linked list of 5 names of programming languages using appropriate collection class. Remove the first and last element of the linked list and print the remaining names iterating the list.

TEXT BOOKS

1. Herbert Schildt, —Java – The Complete Reference –, 9th Edition, 2014, Oracle Press.

REFERENCE BOOKS

1. Y. Daniel Liang, —Introduction to JAVA Programming, 6thEdition, 2007, Pearson Education,
2. Stephanie Bodoff et al, —The J2EE Tutorial, 2nd Edition, 2004, Pearson Education.
3. Head First Java, O'Reilly Publication, 2005.

TEACHING METHODS

- Black Board/Power Point Presentations
- Demonstration of Applications through IDE

ASSESSMENT METHODS

Parameter	**Marks**
Three Internals (Average of best of two)	30
Programming Assignment based Test	10
Rubrics for the evaluation of Programming Assignments	10
Total	**50**
Final Examination will be conducted for 100 marks (SEE)	

COURSE OUTCOMES

Sl. No.	COURSE OUTCOMES	BL
CO1	Apply the object-oriented principles like Encapsulation, Abstraction inheritance and polymorphism to solve the real-world problems	L3
CO2	Demonstrate with examples the usage of Inheritance, Interfaces, packages and Exception handling	L3
CO3	Implement multithreaded Java applications, demonstrate exception handling in Java	L3
CO4	Use the Java's Collection framework to solve computing real-world problems.	L4
CO5	Use JavaFX for interactive UI development and event handling	L4

CO-PO MAPPING

	PROGRAM OUTCOMES												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3		3						1	1			3	2	
CO2	2		3						1	1			3	2	
CO3	2		3						1	1			3	2	
CO4	2		3						1	1			3	2	
CO5	2		3						1	1			3	2	
Correlation level	2		3						1	1			3	2	