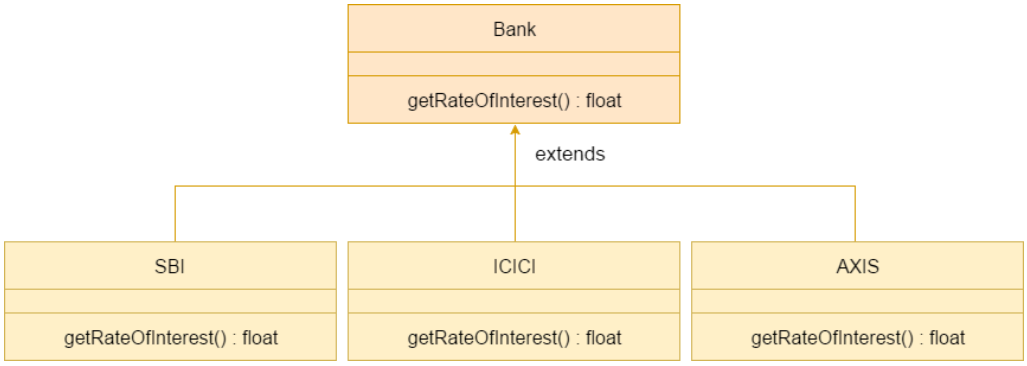
RAMAIAH Institute of Technology

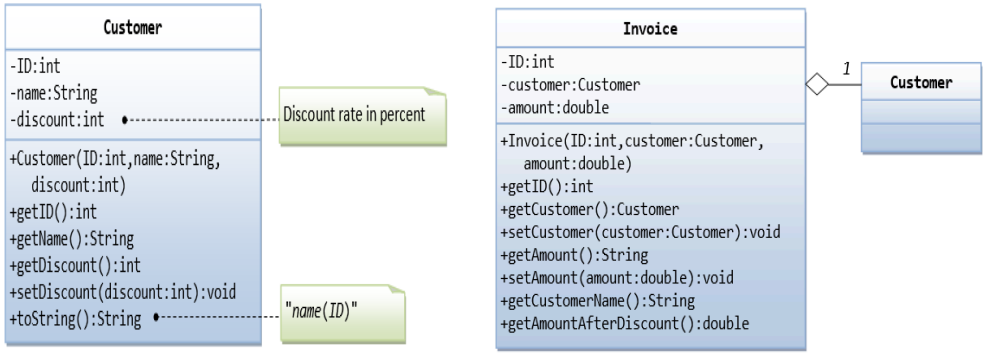
Department of Computer Science & Engineering

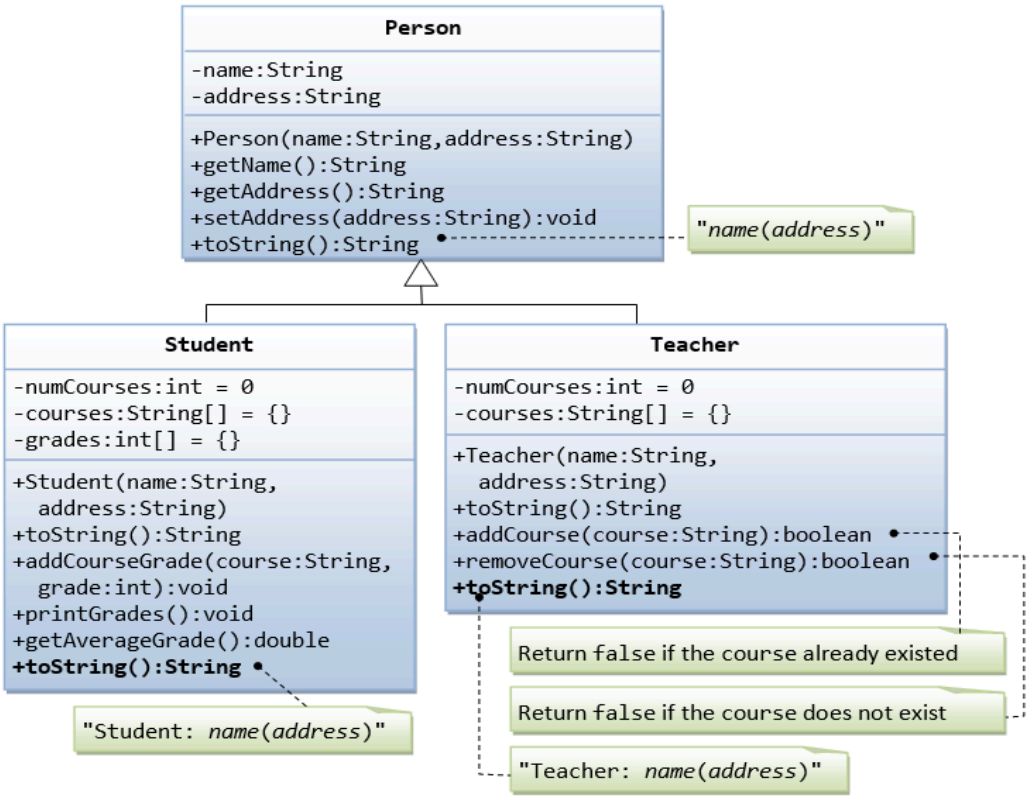
Course Code:	**CSL37**	**TERM: Nov 2023 – Mar 2024**	
Course Name:	**Object Oriented Programing Laboratory**	**Faculty In-charge:**	**Jamuna S Murthy**
Credits:	**0:0:1**	**SEMESTER :**	**III**

Design, develop, and implement the following programs in Java

	PART - A	CO	PO/PSO
1.	Write a java program that demonstrate Method overriding: a. Create a class Bank that provides method to get the rate of interest called "getRateOfinterest() which is of type float. b. The rate of interest varies according to banks. SBI, ICICI and AXIS banks could provide 8%, 7%, and 9% rate of interest. c. Create Inheritance mechanism where SBI, ICICI and AXIS bank child classes extends Bank parent class that contains getRateOfinterest() method. d. Override getRateOfinterest() method in the child classes with updated interests as 8%, 7%, and 9%. e. Finally create a driver class BankDetails that can display the rate of interest of all the banks. 	1	2,3,5/3
2.	Write a Java Program that does the following related to Inheritance: **1.** Create an abstract class called 'Vehicle' which contains the 'hashelmet','year of manufacture' and two abstract methods 'getData()' and 'putData()'. Demonstrate the error when attempt is made to create objects of 'Vehicle'. **2.** Have two derived classes 'TwoWheeler' and 'FourWheeler'. 'FourWheeler' is a final class. Demonstrate the error when attempt is made to inherit from 'FourWheeler'. **3.** Your abstract class should haveoverloadedconstructors that initializes 'hashelmet' and 'year of manufacture for TwoWheeler and FourWheeler respectively. **4.** 'TwoWheeler' has data elements 'Brand', 'Cost', 'EngineType' (possible values "2 stroke", "4 stroke"), and 'Color' which are private, protected, 'friendly/default' and public respectively. Demonstrate the various ways in which the two abstract methods can be dealt 'getData()' and 'putData()' can be dealt with by the derived classes, 'TwoWheeler' and 'FourWheeler'.	1	2,3,5/3

3.	Write a Java Program that does the following: a. Create an abstract class called 'Shape' which contains Two instance variables color (String) and filled (boolean). ● Two constructors: a no-arg (no-argument) constructor that initializes the color to "green" and filled to true, and a constructor that initializes the color and filled to the given values. ● Getter and setter for all the instance variables. By convention, the getter for a boolean variable xxx is called isXXX() (instead of getXxx() for all the other types). ● A toString() method that returns "A Shape with color of xxx and filled/Not filled". ● An abstract method getArea() ● Demonstrate the error when attempt is made to create objects of 'Shape'. b. Write two subclasses of Shape called Circle and Rectangle. Rectangle is a final class. Demonstrate the error when attempt is made to inherit from 'Rectangle'. c. Write a class called Square, as a subclass of Rectangle. Convince yourself that Square can be modeled as a subclass of Rectangle. Square has no instance variable, but inherits the instance variables width and length from its superclass Rectangle.	1	2,3,5/3
4.	Write a Java Program that does the following a. Create a superclass, Student, and two subclasses, Undergrad and Grad. b. The superclass Student should have the following data members: name, ID, grade, age c. The superclass, Student should have at least one method: Boolean isPassed (double grade) d. The purpose of the isPassed method is to take one parameter, grade (value between 0 and 100) and check whether the grade has passed the requirement for passing a course. In the Student class this method should be empty as an abstract method. e. The two subclasses, Grad and Undergrad, will inherit all data members of the Student class and override the method isPassed. For the UnderGrad class, if the grade is above 70.0, then isPassedreturns true, otherwise it returns false. For the Grad class, if the grade is above 80.0, then isPassed returns true, otherwise returns false. f. Demonstrate "final" keyword in the above class. g. Create a test class for your three classes. In the test class, create one Grad object and one Undergrad object. For each object, provide a grade and display the results of the isPassed method.	1	2,3,5/3
5.	Write a Java Program that does the following a. Create a super class called Car. The Car class has the following fields and methods. ● int speed; double regularPrice; String color; double getSalePrice(); b. Create a sub class of Car class and name it as Truck. The Truck class has the following fields and methods. ● int weight; double getSalePrice(); ● //Ifweight>2000,10% discount. Otherwise,20%discount. c. Create a subclass of Car class and name it as Ford. The Ford class has the following fields and methods ● int year; intmanufacturerDiscount; double getSalePrice(); ● //From the sale price computed from Carclass, subtract the manufacturer Discount. d. Create a subclass of Car class and name it as Sedan. The Sedan class has the following fields and methods. ● int length; double getSalePrice(); ● //If length>20feet, 5% discount, Otherwise, 10% discount.	1	2,3,5/3

	e. Create MyOwnAutoShop class which contains the main() method. Perform the following within the main() method. ● Create an instance of Sedan class and initialize all the fields with appropriate values. ● Use super(...) method in the constructor for initializing the fields of the superclass. ● Create an instance of the Ford class and initialize all the fields with appropriate values ● Use super(...) method in the constructor for initializing the fields of the super class. ● Create an instance of Car class and initialize all the fields with appropriate values. Display the sale prices of all instances.		
6.	Write a Java Program that implements the following ● Define a class SavingsAccount with following characteristics. ● Use a static variable annualInterestRate to store the annual interest rate for all account holders. ● Private data member savingsBalance indicating the amount the saver currently has on deposit. ● Method calculateMonthlyInterest to calculate the monthly interest as (savingsBalance * annualInterestRate / 12). After calculation, the interest should be added to savingsBalance. ● Static method modifyInterestRate to set annualInterestRate. ● Parameterized constructor with savingsBalance as an argument to set the value of that instance. ● Test the class SavingsAccount to instantiate two savingsAccount objects, saver1 and saver2, with balances of Rs.2000.00 and Rs3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month's interest and print the new balances for both savers.	1	2,3,5/3
7.	Write a Java Program that does the following ● The Customer class models a customer is design as shown in the class diagram. Write the codes for the Customer class and a test driver to test all the public methods.  ● The Invoice class, design as shown in the class diagram, composes a Customer instance (written earlier) as its member. Write the codes for the Invoice class and a test driver to test all the public methods.	1	2,3,5/3
8.	We are required to model students and teachers in our application. We can define a superclass called `Person` to store common properties such as `name` and `address`, and subclasses `Student` and `Teacher` for their specific properties. For students, we need to maintain the courses taken and their respective grades; add a course with grade, print all courses taken and the average grade. Assume that a student takes no more than 30 courses for the entire program. For teachers, we need to maintain the courses taught currently, and able to add or remove a course taught. Assume that a teacher teaches not more than 5 courses concurrently.	1	2,3,5/3



PART - B

1.	Write a JAVA program which does the following operations: a. An Interface class for Stack Operations b. A Class that implements the Stack Interface and creates a fixed length Stack. c. A Class that implements the Stack Interface and creates a Dynamic Length Stack. d. A Class that uses both the above Stacks through Interface reference and does the Stack operations that demonstrates the runtime binding.	1	2,3,5/3
2.	Write a Java Program to implement Packages by performing following operations: a. Create a class TwoDim which contains private members as x and y coordinates in package P1. Define the default constructor, a parameterized constructor and override toString() method to display the co-ordinates. b. Reuse the class TwoDim and in package P2 create another class ThreeDim, adding a new dimension as z as its private member. Define the constructors for the subclass and override toString() method in the subclass also. c. Write a driver code that imports both packages and usage of classes TwoDim and ThreeDim by creating objects.	1	2,3,5/3
3.	Write Java multithreaded program to implement y=sinx+cosx+tanx	2	2,3,5/3
4.	Write a Java program to create an applet with text box. We must type a number in text box first. Then if we press "J" key the reverse of the given number must be displayed on the status bar.	3	2,3,5/3
5.	Write a java program to throw a exception (checked) for an employee details. If an employee name is a number, a name exception must be thrown. If an employee age is greater than 50, an age exception must be thrown. Or else an object must be created for the entered employee details.	2	2,3,5/3
6.	Write a Java program to display multiplication table of 8 & 9 using shared resources "synchronized displayTable(int num)". The table should be displayed with 1 sec delay between every number. First print multiplication table of 8 and then 9.	2	2,3,5/3

7.	Write a Java program to implement "ADDTION" and "MULTIPLICATION" of two numbers using Lambda Expressions	2	2,3,5/3
8.	Write a java program to accept a string. Convert the string to uppercase. Count and output the number of double letter sequences that exist in the string. Sample Input: “SHE WAS FEEDING THE LITTLE RABBIT WITH AN APPLE” Sample Output: 4	3	2,3,5/3

Marks Distribution

Conduction and Result	Part	Write-up	Execution	Viva	Total	Change of Program
	A	5M	20M	7M	50	-8M
	B	3M	15M			

Course Coordinator **Reviewer** **HoD, Dept. of CSE**