University of Colombo School of Computing
IS 2104 - Rapid Application Development

Lab Sheet 07

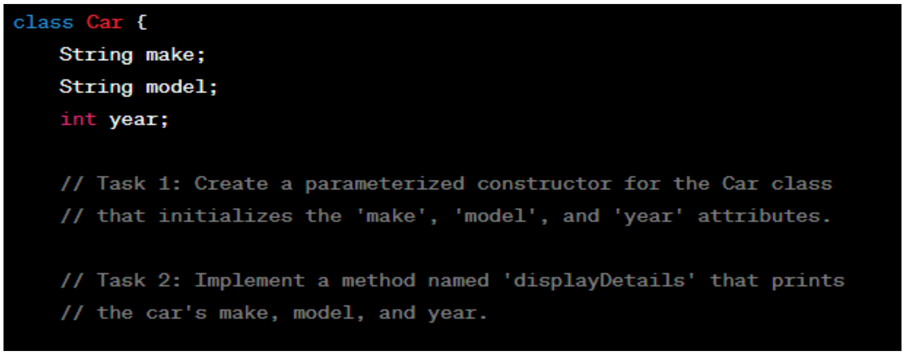
Activity 01

Create a Java class called "**Calculator**" with overloaded methods for performing basic mathematical operations (addition, subtraction, multiplication, and division). Implement methods for both integers and floating-point numbers.

- Create a Java class named "Calculator."
- Define overloaded methods for each of the basic mathematical operations (addition, subtraction, multiplication, and division). You will need two methods for each operation, one for integers and one for floating-point numbers.
- Repeat the above step for subtraction, multiplication, and division, creating overloaded methods for both integers and floating-point numbers.
- Inside each method, perform the corresponding mathematical operation and return the result.
- In your main method or a separate class, create an instance of the "Calculator" class and use it to perform various mathematical operations on both integers and floating-point numbers.

Activity 02

Complete following tasks.



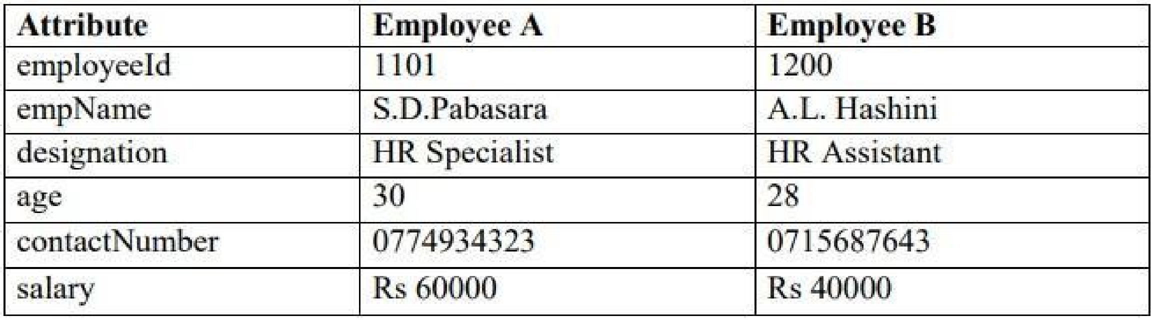
```
class Car {
    String make;
    String model;
    int year;

    // Task 1: Create a parameterized constructor for the Car class
    // that initializes the 'make', 'model', and 'year' attributes.

    // Task 2: Implement a method named 'displayDetails' that prints
    // the car's make, model, and year.
```

Task 3 - Create a method to display the details of the cars.

Task 4 - Print three car details using the method that you created.

Activity 03

(a) Create a class to store details of employees and show all the details of the employees using a suitable method.
(b) Set salary variable value as 40000.
(c) Use another constructor to set salaries other than 40000.



Attribute	Employee A	Employee B
employeeId	1101	1200
empName	S.D.Pabasara	A.L. Hashini
designation	HR Specialist	HR Assistant
age	30	28
contactNumber	0774934323	0715687643
salary	Rs 60000	Rs 40000

Activity 04

Write a Java program to create an abstract class Shape3D with abstract methods **calculateVolume()** and **calculateSurfaceArea()**. Create subclasses Sphere and Cube that extend the Shape3D class and implement the respective methods to calculate the volume and surface area of each shape.

Activity 05

Create a Java class called Person to represent individuals. The class has private instance variables name and age. You have to implement a parameterized constructor to initialize these variables using the "**this**" keyword and provide a method to display the person's details.

I. Implement a parameterized constructor in the Person class that takes name and age as parameters and uses the "this" keyword to initialize the instance variables.

II. Create a method named displayDetails in the Person class that prints the person's name and age.

III. Write a code snippet to create an instance of the Person class, set the name and age using the constructor, and then call the displayDetails method to print the person's details.

Activity 06

You are designing a Java class hierarchy for a zoo management system. You have a superclass called Animal with private instance variables name and species. Implement a parameterized constructor in the Animal class to initialize these variables. Additionally, create a subclass called Zookeeper that inherits from Animal. The Zookeeper class should have an additional private instance variable employeeID.

IV. Write the code for the parameterized constructor in the Animal class that takes name and species as parameters and uses the "this" keyword to initialize the instance variables.

V. Implement getters and setters for the name, species, and employeeID variables in the Animal and Zookeeper classes.

VI. Write a code snippet to create an instance of the Zookeeper class, set the name, species, and employeeID using the setters, and then retrieve and print these values using the getters.

Activity 07

Write a Java program that includes a Person class with a constructor, getters, and setters for firstName and lastName, a Rebel class that extends Person and adds a rebelId variable with corresponding getters and setters, and a PersonDemo class with a main method. In the main method, create an instance of the Rebel class with the first name 'Elon', last name ‘zuckerberg' and rebel ID 'R231.' Then, print out the first name, last name, and rebel ID of the rebel object.