**Experiment: 3**

PART B

(PART A: TO BE COMPLETED AND SUBMITTED BY STUDENTS)

Students must execute all the programs, write executed code in the workbook, and submit part B of experiment 3 on the student portal. The filename should be **OOPJ\_batch\_rollno\_experimentno. Example: OOPJ\_A1\_A001\_P1**

|  |  |
| --- | --- |
| **Roll No.: C126** | **Name: Rushabh Shah** |
| **Prog/Yr/Sem: BTI Sem 8** | **Batch: 2021-2027** |
| **Date of Experiment: 25-01-2025** | **Date of Submission: 25-01-2025** |

**Aim:** **To study Class, object, methods and constructor.**

**Tasks:**

|  |  |
| --- | --- |
| 1 | Write a Java program to define a class LibraryBook with attributes such as title, author, and price. Accept input for these attributes from the user and include a method displayDetails() to print the details. |
| 2 | Create a class named Eligible that has data members as number of classes held and attended, as well as methods: input() and compute(). The input() accepts the data from user and compute() calculates the percentage and verifies that a student having 80% or more than displays a message as “ student permitted to appear in the exam” otherwise “student not permitted to appear in the exam”. Create an instance of a class that invokes the both the methods. |
| 3 | Write a Java program to create a class called Account with instance variables accountNumber and balance. Implement a parameterized constructor that initializes these variables with validation:   * accountNumber should be non-null and non-empty. * balance should be non-negative. * Print an error message if the validation fails. |
| 4 | WAP to compute the area of the room by illustrating the concept of constructor overloading where the length and breadth of the room passed as parameter through constructor which are same in one constructor and different in another constructor. Further use a method that computes the area of the room. |
| 5 | Design a Java program that utilizes method overloading to create a calculator capable of performing arithmetic operations. |
| 6 | A computer science professor written the java code as created a class called “MyClass” with a method named “myMethod” that is overloaded with four different visibility modifiers: public, private, protected and default (package-private). Demonstrate the above concept with method overloading under various visibility modifiers. |
| 7 | Develop a Java program to define a class Student with attributes including name, rollNumber, and marks for three subjects. Take user input for these attributes, and implement two methods: average() to calculate the average of the marks, and printInfo() to display the student's details in a formatted manner. |
| 8 | Write a Java program to create a class called Dog with instance variables name and color. Implement a parameterized constructor that takes name and color as parameters and initializes the instance variables. Print the values of the variables. |
| 9 | WAP to create a class named as Bird which consist the three constructor: first constructor displays the name of the bird with no-arguments. Second constructor takes the name of the bird as a parameter and third constructor takes both name and age of the bird as a parameter. Create three instances of class and display the names of the bird. |

**Executed Code, Input and Output**

|  |  |
| --- | --- |
|  | Write a Java program to define a class LibraryBook with attributes such as title, author, and price. Accept input for these attributes from the user and include a method displayDetails() to print the details. |
| **Executed Code: -**  import java.util.Scanner;  class main{  public static void main(String[] args){  Scanner sc = new Scanner(System.in);  System.out.print("Enter title: ");  String title = sc.nextLine();  System.out.print("Enter Author: ");  String author = sc.nextLine();  System.out.print("Enter Price: ");  Double price = sc.nextDouble();  LibraryBook book1 = new LibraryBook(title,author,price);  book1.displayDetails();  }  }  public class LibraryBook{  String title, author;  double price;  LibraryBook(String title,String author,double price){  this.title = title;  this.author = author;  this.price = price;  }  public void displayDetails(){  System.out.println("Title: "+ title);  System.out.println("Author: "+ author);  System.out.println("Price: "+ price);  }  }  **Input Output: -** | |
|  | Create a class named Eligible that has data members as number of classes held and attended, as well as methods: input() and compute(). The input() accepts the data from user and compute() calculates the percentage and verifies that a student having 80% or more than displays a message as “ student permitted to appear in the exam” otherwise “student not permitted to appear in the exam”. Create an instance of a class that invokes the both the methods. |
| **Executed Code: -**  import java.util.Scanner;  class main{  public static void main(String[] args){  Eligible person1 = new Eligible();  person1.input();  person1.compute();  }  }  public class Eligible{  float held,attended;  public void input(){  Scanner sc = new Scanner(System.in);  System.out.print("Enter number of classes held : ");  held = sc.nextFloat();  System.out.print("Enter number of classes attended : ");  attended = sc.nextFloat();  }  public void compute(){  float atten = (attended/held) \* 100;  if(atten >80){  System.out.println("Student is Permitted");  }  else{  System.out.println("Student is not Permitted ");  }  }  }  **Input Output: -** | |
|  | Write a Java program to create a class called Account with instance variables accountNumber and balance. Implement a parameterized constructor that initializes these variables with validation:   * accountNumber should be non-null and non-empty. * balance should be non-negative. * Print an error message if the validation fails. |
| **Executed Code: -**  import java.util.Scanner;  class Main {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  System.out.print("Enter Account Number: ");  String accountNumber = sc.nextLine();    System.out.print("Enter Initial Balance: ");  double balance = sc.nextDouble();    Account account = new Account(accountNumber, balance);  account.displayDetails();  }  }  public class Account {  String accountNumber;  double balance;  public Account(String accountNumber, double balance) {  if (accountNumber == null || accountNumber.isEmpty()) {  System.out.println("Account number cannot be null or empty.");  return;  }  if (balance < 0) {  System.out.println("Balance cannot be negative.");  return;  }  this.accountNumber = accountNumber;  this.balance = balance;  }  public void displayDetails() {  System.out.println("Account Number: " + accountNumber);  System.out.println("Account Balance: " + balance);  }  }  **Input Output: -** | |
|  | WAP to compute the area of the room by illustrating the concept of constructor overloading where the length and breadth of the room passed as parameter through constructor which are same in one constructor and different in another constructor. Further use a method that computes the area of the room. |
| **Executed Code: -**  import java.util.Scanner;  class Main {  public static void main(String[] args) {  Room room1 = new Room(50);  Room room2 = new Room(10,20);  System.out.println("Area of Room1 is "+room1.Area());  System.out.print("Area of Room2 is "+room2.Area());  }  }  public class Room {  double length,breadth;  public Room(double side) {  this.length = side;  this.breadth = side;  }  public Room(double length,double breadth) {  this.length = length;  this.breadth = breadth;  }  public double Area(){  return length\*breadth;  }  }  **Input Output: -** | |
|  | Design a Java program that utilizes method overloading to create a calculator capable of performing arithmetic operations. |
| **Executed Code: -**  class main{  public static void main (String[]args) {  Calculator calc = new Calculator();  System.out.println("Addition (int): " + calc.add(10, 5));  System.out.println("Addition (double): " + calc.add(10.5, 5.5));  System.out.println("Subtraction (int): " + calc.subtract(10, 5));  System.out.println("Subtraction (double): " + calc.subtract(10.5, 5.5));  System.out.println("Multiplication (int): " + calc.multiply(10, 5));  System.out.println("Multiplication (double): " + calc.multiply(10.5, 5.5));  System.out.println("Division (int): " + calc.divide(10, 5));  System.out.println("Division (double): " + calc.divide(10.5, 5.5));  }  }  public class Calculator {  public int add(int a, int b) {  return a + b;  }  public double add(double a, double b) {  return a + b;  }  public int subtract(int a, int b) {  return a - b;  }  public double subtract(double a, double b) {  return a - b;  }  public int multiply(int a, int b) {  return a \* b;  }  public double multiply(double a, double b) {  return a \* b;  }  public double divide(int a, int b) {  return (double) a / b;  }  public double divide(double a, double b) {  return a / b;  }  }  **Input Output: -** | |
|  | A computer science professor written the java code as created a class called “MyClass” with a method named “myMethod” that is overloaded with four different visibility modifiers: public, private, protected and default (package-private). Demonstrate the above concept with method overloading under various visibility modifiers. |
| **Executed Code: -**  class q6 {  public void myMethod() {  System.out.println("This is a public myMethod.");  }  private void myMethod(int x) {  System.out.println("This is a private myMethod with parameter: " + x);  }  protected void myMethod(String s) {  System.out.println("This is a protected myMethod with parameter: " + s);  }  void myMethod(double d) {  System.out.println("This is a default (package-private) myMethod with parameter: " + d);  }  public static void main(String[] args) {  q6 obj = new q6();    obj.myMethod();  obj.myMethod(10);  obj.myMethod("Hello");  obj.myMethod(3.14);  }  }  **Input Output: -** | |

**Question 7:** Develop a Java program to define a class Student with attributes including name, rollNumber, and marks for three subjects. Take user input for these attributes, and implement two methods: average() to calculate the average of the marks, and printInfo() to display the student's details in a formatted manner.

**Code:**

import java.util.Scanner;

public class q7{

public static void main(String[] args) {

Student student = new Student();

student.inputDetails();

student.printInfo();

}

}

class Student {

String name;

int rollNumber;

int marks1, marks2, marks3;

void inputDetails() {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter name: ");

name = scanner.nextLine();

System.out.print("Enter roll number: ");

rollNumber = scanner.nextInt();

System.out.print("Enter marks for subject 1: ");

marks1 = scanner.nextInt();

System.out.print("Enter marks for subject 2: ");

marks2 = scanner.nextInt();

System.out.print("Enter marks for subject 3: ");

marks3 = scanner.nextInt();

}

double average() {

return (marks1 + marks2 + marks3) / 3.0;

}

void printInfo() {

System.out.println("\nName: " + name);

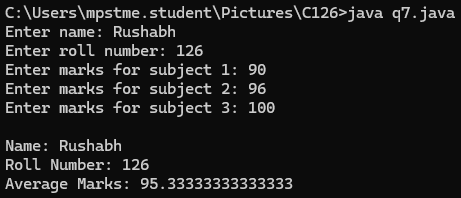
System.out.println("Roll Number: " + rollNumber);

System.out.println("Average Marks: " + average());

}

}

**Input/Output:**

****

**Question 8:** Write a Java program to create a class called Dog with instance variables name and color. Implement a parameterized constructor that takes name and color as parameters and initializes the instance variables. Print the values of the variables.

**Code:**

import java.util.Scanner;

class Dog {

String name;

String color;

public Dog(String name, String color) {

this.name = name;

this.color = color;

}

public void printDetails() {

System.out.println("\nDog Name: " + name);

System.out.println("Dog Color: " + color);

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter dog's name: ");

String dogName = scanner.nextLine();

System.out.print("Enter dog's color: ");

String dogColor = scanner.nextLine();

Dog myDog = new Dog(dogName, dogColor);

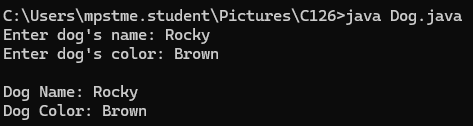
myDog.printDetails();

scanner.close();

}

}

**Input/Output:**

****

**Question 9:** WAP to create a class named as Bird which consist the three constructor: first constructor displays the name of the bird with no-arguments. Second constructor takes the name of the bird as a parameter and third constructor takes both name and age of the bird as a parameter. Create three instances of class and display the names of the bird.

**Code:**

class Bird {

String name;

int age;

public Bird() {

this.name = "Rocky";

}

public Bird(String name) {

this.name = name;

}

public Bird(String name, int age) {

this.name = name;

this.age = age;

}

public void displayInfo() {

System.out.println("\nBird Name: " + name);

if (age != 0) {

System.out.println("Bird Age: " + age);

}

}

public static void main(String[] args) {

Bird bird1 = new Bird();

bird1.displayInfo();

Bird bird2 = new Bird("Eagle");

bird2.displayInfo();

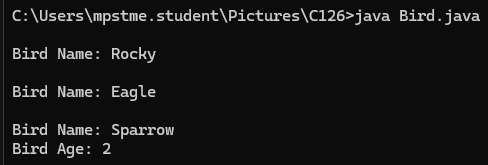
Bird bird3 = new Bird("Sparrow", 2);

bird3.displayInfo();

}

}

**Input/Output:**

****

**Conclusion (Learning Outcomes):** Reflect on the questions answered by you jot down your learnings about the Topic:

**Really good practice of OOPS concepts.**

**Questions:**

**1. What is a constructor? When is it invoked?**

A constructor is a special method used to initialize objects when they are created. It is automatically invoked using the `new` keyword.

**2. What is the difference between Constructor and method/function?**

- A constructor has the same name as the class, no return type, and is used to initialize objects.

- A method has any name, includes a return type, and is used to perform specific tasks after the object is created.

**3. What are the different types of Constructors available in Java? Define.**

- Default Constructor: A constructor that takes no arguments. If you don’t define one, Java provides it automatically.

- Parameterized Constructor: A constructor that takes parameters to set initial values for the object.