**Problem Statement 1: Classes and Objects in Java**

**Name : Subash A**

**Employee : 2604516**

**1.1 Write a program to list all, even numbers less than or equal to the number n. Take the value of n as input from the user.**

**package** Project1;

**import** java.util.Scanner;

**public** **class** Even\_Numbers {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner ip=**new** Scanner(System.***in***);

System.***out***.println("enter the n value");

**int** n=ip.nextInt();

**int** i;

**for**(i=1;i<=n;i++)

{

**if**(i%2==0)

System.***out***.println(i);

}

ip.close();

}

}

**1.2 Define a class Rectangle with its length and breadth.**

**package** Project1;

**import** java.util.Scanner;

**public** **class** LengthBreadth {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the length of the rectangle: ");

**double** length = scanner.nextDouble();

System.***out***.print("Enter the breadth of the rectangle: ");

**double** breadth = scanner.nextDouble();

scanner.close();

Rectangle rectangle = **new** Rectangle(length, breadth);

rectangle.displayInfo();

}

}

**class** Rectangle {

**private** **double** length;

**private** **double** breadth;

**public** Rectangle() {

**this**.length = 0;

**this**.breadth = 0;

}

**public** Rectangle(**double** length, **double** breadth) {

**this**.length = length;

**this**.breadth = breadth;

}

**public** **double** getLength() {

**return** length;

}

**public** **double** getBreadth() {

**return** breadth;

}

**public** **void** setLength(**double** length) {

**this**.length = length;

}

**public** **void** setBreadth(**double** breadth) {

**this**.breadth = breadth;

}

**public** **double** calculateArea() {

**return** length \* breadth;

}

**public** **void** displayInfo() {

System.***out***.println("Length: " + length);

System.***out***.println("Breadth: " + breadth);

System.***out***.println("Area: " + calculateArea());

}

}

**1.3 Create a class Book which describes its book\_title and book\_price. Follow the below steps,**

**a. Use getter and setter methods to get & set the Books description.**

**b. Implement createBooks and showBooks methods to create n objects of Book in anarray.**

**c. Display the books along with its descriptionas follows, Book Title Price Java Programming Rs 350.50 Let Us C Rs.200.00**

**d. Note: createBooks & showBooks should not be member functions of Book class.**

**package** Project1;

**import** java.util.Scanner;

**public** **class** ClassBook {

**public** **static** **void** main(String[] args) {

Book[] books = BookUtils.*createBooks*(2) ;

BookUtils.*showBooks*(books);

}

}

**class** BookUtils {

**public** **static** Book[] createBooks(**int** n) {

**try** (Scanner scanner = **new** Scanner(System.***in***)) {

Book[] books = **new** Book[n];

**for** (**int** i = 0; i < n; i++) {

System.***out***.print("Enter the title of book " + (i + 1) + ": ");

String title = scanner.nextLine();

System.***out***.print("Enter the price of book " + (i + 1) + ": ");

**double** price = scanner.nextDouble();

scanner.nextLine();

books[i] = **new** Book(title, price);

}

**return** books;

}

}

**public** **static** **void** showBooks(Book[] books) {

**for** (Book book : books) {

System.***out***.println("Book Title: " + book.getBookTitle());

System.***out***.println("Rs. " + book.getBookPrice());

}

}

}

**class** Book {

**private** String bookTitle;

**private** **double** bookPrice;

**public** Book() {

**this**.bookTitle = "";

**this**.bookPrice = 0.0;

}

**public** Book(String bookTitle, **double** bookPrice) {

**this**.bookTitle = bookTitle;

**this**.bookPrice = bookPrice;

}

**public** String getBookTitle() {

**return** bookTitle;

}

**public** **double** getBookPrice() {

**return** bookPrice;

}

**public** **void** setBookTitle(String bookTitle) {

**this**.bookTitle = bookTitle;

}

**public** **void** setBookPrice(**double** bookPrice) {

**this**.bookPrice = bookPrice;

}

}

**1.4 Modify the program, which is createdin sub problem 1.2 asfollows, a. The class has attributes lengthandwidth, each of which defaults to 1. b. Itshould have memberfunctions that calculate the perimeter and area of the rectangle. c. Itshould have set and get functions for both lengthand width. d. The set functions should verify that length and width are each floating-point number larger than 0.0 and lessthan 20.0**

   
**package** Project1;

**import** java.util.Scanner;

**public** **class** Perimeter {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the length of the rectangle: ");

**double** length = scanner.nextDouble();

System.***out***.print("Enter the width of the rectangle: ");

**double** width = scanner.nextDouble();

scanner.close();

**try** {

Rectangle rectangle = **new** Rectangle(length, width);

rectangle.displayInfo();

} **catch** (IllegalArgumentException e) {

System.***out***.println(e.getMessage());

}

}

}

**class** Rectangle1 {

**private** **double** length;

**private** **double** width;

**public** Rectangle1() {

**this**.length = 1;

**this**.width = 1;

}

**public** **void** Rectangle12(**double** length, **double** width) {

setLength(length);

setWidth(width);

}

**public** **double** getLength() {

**return** length;

}

**public** **double** getWidth() {

**return** width;

}

**public** **void** setLength(**double** length) {

**if** (length > 0 && length < 20.0) {

**this**.length = length;

} **else** {

**throw** **new** IllegalArgumentException("Length must be between 0 and 20.0");

}

}

**public** **void** setWidth(**double** width) {

**if** (width > 0 && width < 20.0) {

**this**.width = width;

} **else** {

**throw** **new** IllegalArgumentException("Width must be between 0 and 20.0");

}

}

**public** **double** calculateArea() {

**return** length \* width;

}

**public** **double** calculatePerimeter() {

**return** 2 \* (length + width);

}

**public** **void** displayInfo() {

System.***out***.println("Length: " + length);

System.***out***.println("Width: " + width);

System.***out***.println("Area: " + calculateArea());

System.***out***.println("Perimeter: " + calculatePerimeter());

}

}

**1.5 Create a class Date with day, month, and year attributes for manipulating dates. Follow the below steps,**

**a. Provide a constructorthat enables an object of this class to be initialized when it is declared (You can select any default values for the day, month & year, e.g., your birth date).**

**b. Provide the necessary functionality toperform error checking on the initializer values for data members day, month, and year.**

**c. Provide a member function to add an integerin a date to obtain a new date. d. Designseparate class Employee which will have following information. Refer below table,**

**e. Provide appropriate constructor(s)& methodsto this class. Provide main function which willcreate 5 objects of Employee class and display employee information.**   
**package** Project1;

**import** java.time.LocalDate;

**import** java.time.format.DateTimeParseException;

**import** java.util.Scanner;

**public** **class** Classdata {

**public** **static** **void** main(String[] args) {

**try** (Scanner scanner = **new** Scanner(System.***in***)) {

Employee[] employees = **new** Employee[5];

**for** (**int** i = 0; i < 5; i++) {

System.***out***.println("Enter details for employee " + (i + 1));

System.***out***.print("Enter employee number: ");

**while** (!scanner.hasNextInt()) {

System.***out***.println("Invalid input. Please enter a valid employee number.");

scanner.next();

}

**int** employeeNumber = scanner.nextInt();

scanner.nextLine();

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

String employeeName = scanner.nextLine();

System.***out***.print("Enter joining date (YYYY-MM-DD): ");

String dateInput = scanner.nextLine();

**try** {

LocalDate joiningDate = LocalDate.*parse*(dateInput);

employees[i] = **new** Employee(employeeNumber, employeeName, joiningDate);

} **catch** (DateTimeParseException e) {

System.***out***.println("Invalid date format. Please enter the details again.");

i--;

}

}

**for** (Employee employee : employees) {

System.***out***.println("\n" + employee);

}

}

}

}

**class** Employee {

**private** **int** employeeNumber;

**private** String employeeName;

**private** LocalDate joiningDate;

**public** Employee(**int** employeeNumber, String employeeName, LocalDate joiningDate) {

**this**.employeeNumber = employeeNumber;

**this**.employeeName = employeeName;

**this**.joiningDate = joiningDate;

}

**public** **int** getEmployeeNumber() {

**return** employeeNumber;

}

**public** String getEmployeeName() {

**return** employeeName;

}

**public** LocalDate getJoiningDate() {

**return** joiningDate;

}

**public** String toString() {

**return** "Employee Number: " + employeeNumber + "\nEmployee Name: " + employeeName + "\nJoining Date: " + joiningDate;}}

**Problem Statement 2:**

**Encapsulation and Inheritance in Java OOPs**

**Design a Java program to manage different types of vehicles (Car, Motorcycle, Truck) with the following features:**

**Problem Statement 2:**

**Main Class:**    
**package** Project1;

**import** Project1.Car1;

**import** Project1.Motorcycle;

**import** Project1.Truck1;

**public** **class** TestVehicle1 {

**public** **static** **void** main(String[] args) {

Car1 c = **new** Car1();

c.setManufacturer("BMW");

c.setModel("VERSION 2R");

c.setYear(2016);

c.setSeatingCapacity(5);

c.displayDetails();

c.accelerate();

c.brake();

Truck1 t = **new** Truck1();

t.setManufacturer("ROLLS ROYCE");

t.setModel("LPT 348");

t.setYear(2018);

t.setCargoCapacity(38000);

t.displayDetails();

t.loadCargo();

t.unloadCargo();

Motorcycle m = **new** Motorcycle();

m.setManufacturer("NINJA");

m.setModel("ROYAL ENFIELD HIMALAYAN");

m.setYear(2010);

m.setEngineCapacity(125);

m.displayDetails();

m.startEngine();

m.stopEngine();

}

}

**Vehicle Class:**

**• Create a superclass Vehicle with attributes such as manufacturer, model, year.**

**• Include methods to get and set these attributes (getManufacturer(), getModel(), getYear(), setManufacturer(String), setModel(String), setYear(int)).**

**• Implement a method displayDetails() to display all attributes of the vehicle.**   
Vehicle Class:    
**package** Project1;

**public** **class** Vehicle1 {

String manufacturer;

String model;

**int** year;

**public** String getManufacturer() {

**return** manufacturer;

}

**public** **void** setManufacturer(String manufacturer) {

**this**.manufacturer = manufacturer;

}

**public** String getModel() {

**return** model;

}

**public** **void** setModel(String model) {

**this**.model = model;

}

**public** **int** getYear() {

**return** year;

}

**public** **void** setYear(**int** year) {

**this**.year = year;

}

**public** **void** displayDetails() {

System.***out***.println("Manufacturer: "+getManufacturer());

System.***out***.println("Model: "+getModel());

System.***out***.println("Year: "+getYear());

}

}

**Car Class:**

**• Extend Vehicle and include an additional attribute seatingCapacity.**

**• Implement methods getSeatingCapacity() and setSeatingCapacity(int) to manipulate this attribute.**

**• Override the displayDetails() method to include seatingCapacity. • Include specific operations such as accelerate(), brake()**   
Car Class:

**package** Project1;

**public** **class** Car1 {

**int** seatingCapacity;

**public** **int** getSeatingCapacity() {

**return** seatingCapacity;}

**public** **void** setSeatingCapacity(**int** seatingCapacity) {

**this**.seatingCapacity = seatingCapacity;

}

**public** **void** accelerate() {

System.***out***.println("Car applies to accelerate");

}

**public** **void** brake() {

System.***out***.println("Car applies brake");

}

**public** **void** displayDetails() {

// super.displayDetails();

System.***out***.println("Car Seating Capacity: " + getSeatingCapacity());

}

**public** **void** setManufacturer(String string) {

System.***out***.println("Car applies manufactureer");

}

**public** **void** setModel(String string) {

System.***out***.println("Car model");

}

**public** **void** setYear(**int** i) {

System.***out***.println("Car applied year");}}

**Motorcycle Class:**

**• Extend Vehicle and include an additional attribute engineCapacity.**

**• Implement methods getEngineCapacity() and setEngineCapacity(double) to manipulate this attribute.**

**• Override the displayDetails() method to include engineCapacity.**

**• Include specific operations such as startEngine(), stopEngine()**

Motorcycle Class:    
   
**package** Project1;

**public** **class** Motorcycle {

**double** engineCapacity;

**public** **double** getEngineCapacity() {

**return** engineCapacity;

}

**public** **void** setEngineCapacity(**double** engineCapacity) {

**this**.engineCapacity = engineCapacity;

}

**public** **void** startEngine() {

System.***out***.println("Motorcycle engine started");

}

**public** **void** stopEngine() {

System.***out***.println("Motorcycle engine stoped");

}

**public** **void** displayDetails() {

// super.displayDetails();

System.***out***.println("Motorcycle engine Capacity: "+getEngineCapacity());

}

**public** **void** setManufacturer(String string) {

System.***out***.println("Motorcycle manufacture");

}

**public** **void** setModel(String string) {

System.***out***.println("Motorcycle model");

}

**public** **void** setYear(**int** i) {

System.***out***.println("Motorcycle year");

}

}

**Truck Class:**

**• Extend Vehicle and include an additional attribute cargoCapacity.**

**• Implement methods getCargoCapacity() and setCargoCapacity(double) to manipulate this attribute.**

**• Override the displayDetails() method to include cargoCapacity.**

**• Include specific operations such as loadCargo(), unloadCargo().** 

Truck Class:    
   
**package** Project1;

**public** **class** Truck1 {

**double** cargoCapacity;

**public** **double** CargoCapacity() {

**return** cargoCapacity;

}

**public** **void** setCargoCapacity(**double** cargoCapacity) {

**this**.cargoCapacity = cargoCapacity;

}

**public** **void** loadCargo() {

System.***out***.println("Cargo is loading");

}

**public** **void** unloadCargo() {

System.***out***.println("Cargo is unloading");

}

**private** String getCargoCapacity() {

**return** **null**;

}

**public** **void** setManufacturer(String string) {

System.***out***.println("Cargo Manufacturer");

}

**public** **void** setModel(String string) {

System.***out***.println("Cargo model");

}

**public** **void** setYear(**int** i) {

System.***out***.println("Cargo year");

}

**public** **void** displayDetails() {

System.***out***.println("Truck Cargo Capacity: " + getCargoCapacity());}}

**Problem Statement 3:**

**Abstraction in Java OOPs Design a Java program to calculate areas of different shapes (Circle, Rectangle, Triangle):**

**• Define an abstract class Shape with an abstract method calculateArea().**

**• Implement classes for each shape extending Shape and provide necessary attributes (radius and sideLength for circle, rectangle, and triangle respectively).**

**• Use abstraction to ensure that each shape class implements its own logic to calculate the area based on its specific attributes.**   
Shape class    
**package** Project1;

**public** **abstract** **class** Subash\_shape\_class {

**abstract** **double** calculateArea();

}

Circle class    
**package** Project1;

**public** **class** circle\_shape **extends** Subash\_shape\_class {

**int** radius;

**int** sidelength;

**public** circle\_shape(**int** radius, **int** sidelength) {

**super**();

**this**.radius = radius;

**this**.sidelength = sidelength;

}

**public** **double** calculateArea()

{

**return** 3.14\*radius\*radius;

}

}

Rectangle class

**package** Project1;

**public** **class** Rectangle\_shape {

**int** length;

**int** breadth;

**public** Rectangle\_shape(**int** length, **int** breadth) {

**super**();

**this**.length = length;

**this**.breadth = breadth;

}

**double** calculateArea()

{

**return** length\*breadth;

}

}

Triangle class

**package** Project1;

**public** **class** Triangle\_shape {

**int** base;

**int** height;

**public** Triangle\_shape(**int** base, **int** height) {

**super**();

**this**.base = base;

**this**.height = height;

}

**double** calculateArea()

{

**return** 0.5\*base\*height;

}

}

Main class  

**package** Project1;

**public** **class** Mainclass {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**double** area,area1,area2;

circle\_shape c=**new** circle\_shape(8,10);

Rectangle\_shape r=**new** Rectangle\_shape(12,20);

Triangle\_shape t=**new** Triangle\_shape(2,4);

area=c.calculateArea();

System.***out***.println("area of circle "+area);

area1=r.calculateArea();

System.***out***.println("area of rectangle "+area1);

area2=t.calculateArea();

System.***out***.println("area of triangle "+area2);

}

}

**Problem Statement 4:**

**Implement the Static Classes and Methods in Java VastMindz Company is creating a performance rating system to calculate the performance of each employee. Designthe below classestoachieve the same.**

**package** Project1;

**public** **class** Subash\_employee\_class {

**private** String name;

**private** **int** points;

**private** **static** **int** *employeeCount* = 0; // Static variable to count instances

// Constructor

**public** Subash\_employee\_class(String name, **int** points) {

**this**.name = name;

**this**.points = points;

*employeeCount*++; // Increment count when a new employee is created

}

// Getter and Setter for name

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

// Getter and Setter for points

**public** **int** getPoints() {

**return** points;

}

**public** **void** setPoints(**int** points) {

**this**.points = points;

}

// Static method to get the employee count

**public** **static** **int** getEmployeeCount() {

**return** *employeeCount*;

}}

Performance Rating class:

**package** Project1;

**public** **class** Subash\_performance\_rating {

**public** **static** **final** **int** ***OUTSTANDING*** = 5;

**public** **static** **final** **int** ***GOOD*** = 4;

**public** **static** **final** **int** ***AVERAGE*** = 3;

**public** **static** **final** **int** ***POOR*** = 2;

//Static method to calculate performance

**public** **static** **int** calculatePerformance(Subash\_employee\_class e) {

**int** points = e.getPoints();

**if** (points >= 90) {

**return** ***OUTSTANDING***;

} **else** **if** (points >= 75) {

**return** ***GOOD***;

} **else** **if** (points >= 50) {

**return** ***AVERAGE***;

} **else** {

**return** ***POOR***;

}

}

}

Develop the main class PerformanceCalculator.

**package** Project1;

**public** **class** Subash\_performance\_calc {

**public** **static** **void** main(String[] args) {

// Creating Employees instances

Subash\_employee\_class emp1 = **new** Subash\_employee\_class("Swetha", 90);

Subash\_employee\_class emp2 = **new** Subash\_employee\_class("Priya", 80);

Subash\_employee\_class emp3 = **new** Subash\_employee\_class("Neha", 60);

Subash\_employee\_class emp4 = **new** Subash\_employee\_class("Dhivya", 40);

// Calculating and printing performance ratings

System.***out***.println(emp1.getName() + "'s Performance Rating: " +

Subash\_performance\_rating .*calculatePerformance*(emp1));

System.***out***.println(emp2.getName() + "'s Performance Rating: " +

Subash\_performance\_rating .*calculatePerformance*(emp2));

System.***out***.println(emp3.getName() + "'s Performance Rating: " +

Subash\_performance\_rating .*calculatePerformance*(emp3));

System.***out***.println(emp4.getName() + "'s Performance Rating: " +

Subash\_performance\_rating .*calculatePerformance*(emp4));

// Printing total number of employee instances created

System.***out***.println("Total Employees Created: " + Subash\_employee\_class.*getEmployeeCount*());

}}