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

* 1. **)**

import java.util.\*;

class EvenNo {

private int n;

public EvenNo(int n){

this.n = n;

}

public void printNo() {

System.out.println("Even numbers are:");

for(int i=2; i<=n; i+=2){

System.out.print(i+" ");

}

}

}

public class Main {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter a number:");

int n=sc.nextInt();

EvenNo obj=new EvenNo(n);

obj.printNo();

}

}

**1.2 ) Rectangle.java**

class Rectangle {

private float l;

private float b;

public Rectangle(){

this.l = 0;

this.b = 0; }

public Rectangle(float l, float b) {

this.l = l;

this.b = b; }

public float getLength(){

return l; }

public float getBreadth(){

return b; }

public void setLength(float l){

this.l = l; }

public void setBreadth(float b){

this.b = b; }

public float area(){

return l\*b; }

public void info(){

System.out.println("Rectangle Details:");

System.out.println("Length: "+l);

System.out.println("Breadth: "+b);

System.out.println("Area: "+area());

}

}

**TestRectangle.java**

import java.util.Scanner;

public class TestRectangle{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

System.out.println("Enter Length:");

float l=sc.nextFloat();

System.out.println("Enter Breadth:");

float b=sc.nextFloat();

Rectangle rectangle=new Rectangle(l,b);

rectangle.info();

}

}

**1.3 )** import java.util.\*;

class Book {

private String t;

private double p;

public Book(){

this.t = "";

this.p = 0.0;

}

public String getTitle(){

return t;

}

public void setTitle(String t){

this.t = t;

}

public double getPrice(){

return p;

}

public void setPrice(double p){

this.p = p;

}

}

public class BookStore{

public static Book[] createBooks(int n){

Scanner sc=new Scanner(System.in);

Book[] books=new Book[n];

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

books[i]=new Book();

System.out.println("Enter title:");

String t=sc.next();

System.out.println("Enter price:");

double p=sc.nextDouble();

books[i].setTitle(t);

books[i].setPrice(p);

}

return books;

}

public static void showBooks(Book[] books){

System.out.println("Book Title & Price:");

for(Book book : books){

System.out.printf("%s\tRs %.2f%n",book.getTitle(), book.getPrice());

}

}

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

System.out.println("Enter n");

int n=sc.nextInt();

Book[] books = createBooks(n);

showBooks(books);

}

}

**1.4 ) Rectangle.java**

class Rectangle {

private double l;

private double b;

public Rectangle(){

this.l = 1.0;

this.b = 1.0; }

public Rectangle(double l, double b) {

this.setLength(l);

this.setBreadth(b); }

public double getLength(){

return l; }

public double getBreadth(){

return b; }

public void setLength(double l){

if(l>0.0 && l<20.0){

this.l=l;

} else{

throw new IllegalArgumentException("Lenght must be b/w 0.0 to 20.0");

}

}

public void setBreadth(double b){

if(b>0.0 && b<20.0){

this.b=b;

} else{

throw new IllegalArgumentException("Breadth must be b/w 0.0 to 20.0");

}

}

public double area(){

return l\*b; }

public double perimeter(){

return 2\*(l+b);

}

public void info(){

System.out.println("Rectangle Details:");

System.out.println("Length: "+l);

System.out.println("Breadth: "+b);

System.out.println("Area: "+area());

System.out.println("Perimeter: "+perimeter());

}

}

**TestRectangle.java**

import java.util.Scanner;

public class TestRectangle{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

System.out.println("Enter Length:");

float l=sc.nextFloat();

System.out.println("Enter Breadth:");

float b=sc.nextFloat();

Rectangle rectangle=new Rectangle(l,b);

rectangle.info();

}

}

**1.5 )** class Date {

private int day;

private int month;

private int year;

public Date(){

this.day=1;

this.month=1;

this.year=2000;

}

public Date(int day, int month, int year){

if(isValidDate(day, month, year)) {

this.day = day;

this.month = month;

this.year = year;

} else {

throw new IllegalArgumentException("Invalid Date");

}

}

private boolean isValidDate(int day, int month, int year){

if(month < 1 || month > 12) return false;

if(day < 1 || day > daysInMonth(month, year)) return false;

return year >= 1;

}

private int daysInMonth(int month, int year){

switch(month){

case 2: return(isLeapYear(year)) ? 29:28;

case 4: case 6: case 9: case 11: return 30;

default: return 31;

}

}

private boolean isLeapYear(int year){

return (year%4==0 && year%100!=0) || (year%400==0);

}

public void addDays(int daysToAdd){

day += daysToAdd;

while(day > daysInMonth(month, year)){

day -= daysInMonth(month, year);

month++;

if(month > 12){

month=1;

year++;

}

}

}

@Override

public String toString(){

return day+"/"+month+"/"+year;

}

}

class Employee{

private int employeeNumber;

private String employeeName;

private Date joiningDate;

public Employee(int employeeNumber, String employeeName, Date joiningDate) {

this.employeeNumber = employeeNumber;

this.employeeName = employeeName;

this.joiningDate = joiningDate;

}

public void displayInfo() {

System.out.println("Employee Number:"+employeeNumber);

System.out.println("Employee Name:"+employeeName);

System.out.println("Joining Date:"+joiningDate);

}

}

public class Main {

public static void main(String[] args){

Employee emp1=new Employee(1,"John",new Date(12, 9, 2015));

Employee emp2=new Employee(2,"Rohan", new Date(5, 4, 2019));

Employee emp3=new Employee(3,"Raj",new Date(9, 6, 2020));

Employee emp4=new Employee(4, "Amit", new Date(25, 12, 2021));

Employee emp5=new Employee(5, "Ghost", new Date(1, 7, 2023));

emp1.displayInfo();

emp2.displayInfo();

emp3.displayInfo();

emp4.displayInfo();

emp5.displayInfo();

}

}

**2 )** class Vehicle {

private String manufacturer;

private String model;

private int year;

public Vehicle(String manufacturer, String model, int year) {

this.manufacturer = manufacturer;

this.model = model;

this.year = 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: " + manufacturer);

System.out.println("Model: " + model);

System.out.println("Year: " + year);

}

}

class Car extends Vehicle {

private int seatingCapacity;

public Car(String manufacturer, String model, int year, int seatingCapacity) {

super(manufacturer, model, year);

this.seatingCapacity = seatingCapacity;

}

public int getSeatingCapacity() {

return seatingCapacity;

}

public void setSeatingCapacity(int seatingCapacity) {

this.seatingCapacity = seatingCapacity;

}

@Override

public void displayDetails() {

super.displayDetails();

System.out.println("Seating Capacity: " + seatingCapacity);

}

public void accelerate() {

System.out.println("The car is accelerating.");

}

public void brake() {

System.out.println("The car is braking.");

}

}

class Motorcycle extends Vehicle {

private double engineCapacity;

public Motorcycle(String manufacturer, String model, int year, double engineCapacity) {

super(manufacturer, model, year);

this.engineCapacity = engineCapacity;

}

public double getEngineCapacity() {

return engineCapacity;

}

public void setEngineCapacity(double engineCapacity) {

this.engineCapacity = engineCapacity;

}

@Override

public void displayDetails() {

super.displayDetails();

System.out.println("Engine Capacity: " + engineCapacity + "cc");

}

public void startEngine() {

System.out.println("The motorcycle engine is starting.");

}

public void stopEngine() {

System.out.println("The motorcycle engine is stopping.");

}

}

class Truck extends Vehicle {

private double cargoCapacity;

public Truck(String manufacturer, String model, int year, double cargoCapacity) {

super(manufacturer, model, year);

this.cargoCapacity = cargoCapacity;

}

public double getCargoCapacity() {

return cargoCapacity;

}

public void setCargoCapacity(double cargoCapacity) {

this.cargoCapacity = cargoCapacity;

}

@Override

public void displayDetails() {

super.displayDetails();

System.out.println("Cargo Capacity: " + cargoCapacity + " tons");

}

public void loadCargo() {

System.out.println("The truck is loading cargo.");

}

public void unloadCargo() {

System.out.println("The truck is unloading cargo.");

}

}

public class Main {

public static void main(String[] args) {

Car car = new Car("Toyota", "Innova", 2022, 5);

Motorcycle motorcycle = new Motorcycle("Kawasaki", "Zx-10r", 2020, 750);

Truck truck = new Truck("Volvo", "FH16", 2019, 25);

System.out.println("Car Details:");

car.displayDetails();

car.accelerate();

car.brake();

System.out.println();

System.out.println("Motorcycle Details:");

motorcycle.displayDetails();

motorcycle.startEngine();

motorcycle.stopEngine();

System.out.println();

System.out.println("Truck Details:");

truck.displayDetails();

truck.loadCargo();

truck.unloadCargo();

}

}

**3 )** abstract class Shape{

abstract double calculateArea();

}

class Circle extends Shape {

private double r;

public Circle(double r) {

this.r = r;

}

@Override

double calculateArea() {

return Math.PI \* r \* r;

}

}

class Rectangle extends Shape {

private double l;

private double w;

public Rectangle(double l, double w) {

this.l = l;

this.w = w;

}

@Override

double calculateArea() {

return l \* w;

}

}

class Triangle extends Shape {

private double b;

private double h;

public Triangle(double b, double h) {

this.b = b;

this.h = h;

}

@Override

double calculateArea() {

return 0.5 \* b \* h;

}

}

public class Main {

public static void main(String[] args) {

Shape circle = new Circle(5);

Shape rectangle = new Rectangle(4, 6);

Shape triangle = new Triangle(3, 7);

System.out.println("Area of Circle: " + circle.calculateArea());

System.out.println("Area of Rectangle: " + rectangle.calculateArea());

System.out.println("Area of Triangle: " + triangle.calculateArea());

}

}

4 ) class Employee {

private String name;

private int point;

private static int employeeCount = 0;

public Employee(String name, int point) {

this.name = name;

this.point = point;

employeeCount++;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getPoint() {

return point;

}

public void setPoint(int point) {

this.point = point;

}

public static int getEmployeeCount() {

return employeeCount;

}

}

class PerformanceRating {

public static final int OUTSTANDING = 5;

public static final int GOOD = 4;

public static final int AVERAGE = 3;

public static final int POOR = 2;

public static int calculate(Employee e) {

int points = e.getPoint();

if (points >= 90) {

return OUTSTANDING;

} else if (points >= 75) {

return GOOD;

} else if (points >= 50) {

return AVERAGE;

} else {

return POOR;

}

}

}

public class PerformanceCalculator {

public static void main(String[] args) {

Employee emp1 = new Employee("Ram", 95);

Employee emp2 = new Employee("Amit", 80);

Employee emp3 = new Employee("Rohan", 45);

System.out.println("Total employees: " +Employee.getEmployeeCount()+" & rating is");

System.out.println(emp1.getName() + " Rating: " + PerformanceRating.calculate(emp1));

System.out.println(emp2.getName() + " Rating: " + PerformanceRating.calculate(emp2));

System.out.println(emp3.getName() + " Rating: " + PerformanceRating.calculate(emp3));

}

}