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

**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 org.day2assignment;

import java.util.Scanner;

public class evenupton {

int n;

void input() {

Scanner sc = new Scanner(System.in);

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

n = sc.nextInt();

}

void output() {

for (int i=1; i<=n;i++) {

if(i%2==0) {

System.out.println(i);

}

}

}

public static void main(String[] args) {

evenupton ob = new evenupton();

ob.input();

ob.output();

}

}

**Input:**

enter a number:

20

**Output:**

2

4

6

8

10

12

14

16

18

20

**1.2 Define a class Rectangle with its length and breadth. Follow the below steps,**

**a. Provide appropriate constructor(s), which gives facility of constructing Rectangle object with default values of length and breadth as 0 or passing value of length and breadth externally to constructor.**

**package** org.day2assignment;

**public** **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;

}

**b. Provide appropriate accessor & mutator methods to Rectangle class.**

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

**return** Length;

}

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

**this**.Length = length;

}

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

**return** Breadth;

}

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

**this**.Breadth = breadth;

}

**c. Provide methods to calculate area & to display all information of Rectangle.**

**public** **double** caluculatearea() {

**return** Length\*Breadth;

}

**public** **void** display() {

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

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

System.***out***.println("the area of rectangle:"+caluculatearea());

}

}

**d. Design different class TestRectangle class in a separate source file, which will contain main method. From this main method, create a Rectangle object by taking all necessary information from the user and calculate respective area of rectangle objects and display it.**

package org.day2assignment;

import java.util.Scanner;

public class Testrectangle {

public static void main(String[] args) {

Rectangle rect = new Rectangle();

rect.setLength(4);

rect.setBreadth(5);

System.*out*.println(rect.caluculatearea());

rect.display();

}

}

**output:**

20.0

the length of rectangle:4.0

the breadth of rectangle:5.0

the area of rectangle:20.0

**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.**

package org.day2assignment;

import java.util.Scanner;

class Book {

private String book\_title;

private double book\_price;

public String getBookTitle() {

return book\_title;

}

public void setBookTitle(String book\_title) {

this.book\_title = book\_title;

}

public double getBookPrice() {

return book\_price;

}

public void setBookPrice(double book\_price) {

this.book\_price = book\_price;

}

}

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

**package** org.day2assignment;

**import** java.util.Scanner;

**public** **class** BookStore {

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

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

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

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

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

Book book = **new** Book();

System.***out***.print("Enter book title: ");

String title = scanner.nextLine();

book.setBookTitle(title);

System.***out***.print("Enter book price: ");

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

scanner.nextLine();

book.setBookPrice(price);

books[i] = book;

}

**return** books;

}

**c. Display the books along with its description**

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

System.***out***.println("Book Title Price");

System.***out***.println("---------------------------------------");

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

System.***out***.printf("%-25s Rs %.2f\n", book.getBookTitle(), book.getBookPrice());

}

}

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

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

System.***out***.print("Enter the number of books to create: ");

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

scanner.nextLine();

Book[] books = *createBooks*(n);

*showBooks*(books);

}

}

**Output:**

Enter the number of books to create: 2

Enter details for book 1

Enter book title: basic java

Enter book price: 2500

Enter details for book 2

Enter book title: c

Enter book price: 1500

Book Title Price

---------------------------------------

basic java Rs 2500.00

c Rs 1500.00

**1.4 Modify the program, which is created in sub problem 1.2 as follows,**

a. The class has attributes length and width, each of which defaults to 1.

b. It should have member functions that calculate the perimeter and area of the rectangle.

c. It should have set and get functions for both length and width.

d. The set functions should verify that length and width are each floating-point number larger

than 0.0 and less than 20.0.

**package** org.day2assignment;

**public** **class** Rectangle1 {

**private** **double** Length;

**private** **double** Breadth;

**public** Rectangle1() {

**this**.Length=1;

**this**.Breadth=1;

}

**public** Rectangle1(**double** length, **double** breadth) {

**if** (length > 0.0 && length < 20.0 && breadth > 0.0 && breadth < 20.0) {

**this**.Length = length;

**this**.Breadth = breadth;

}

}

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

**return** Length;

}

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

**this**.Length = length;

}

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

**return** Breadth;

}

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

**this**.Breadth = breadth;

}

**public** **double** caluculatearea() {

**return** Length\*Breadth;

}

**public** **double** perimeter() {

**return** 2\*(Length+Breadth);

}

**public** **void** display() {

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

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

System.***out***.println("the area of rectangle:"+caluculatearea());

System.***out***.println("the perimeter of rectangle:"+perimeter());

}

}

**package** org.day2assignment;

**import** java.util.Scanner;

**public** **class** Testrectangle1 {

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

Rectangle1 rect = **new** Rectangle1(-1,21);

// rect.setLength(-1);

// rect.setBreadth(21);

System.***out***.println(rect.caluculatearea());

System.***out***.println(rect.perimeter());

rect.display();

}

}

**Output:**

0.0

0.0

the length of rectangle:0.0

the breadth of rectangle:0.0

the area of rectangle:0.0

the perimeter of rectangle:0.0

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

**Follow the below steps,**

a. Provide a constructor that 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 to perform error checking on the initializer values for data

members day, month, and year.

c. Provide a member function to add an integer in a date to obtain a new date.

d. Design separate class Employee which will have following information.

e. Provide appropriate constructor(s)& methods to this class. Provide main function which will create 5 objects of Employee class and display employee information

**package** org.day2assignment;

**class** Date1 {

**private** **int** day;

**private** **int** month;

**private** **int** year;

**public** Date1() {

**this**.day = 1;

**this**.month = 1;

**this**.year = 2000;

}

**public** Date1(**int** day, **int** month, **int** year) {

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

**this**.day = day;

**this**.month = month;

**this**.year = year;

} **else** {

System.***out***.println("Invalid date. Setting to default values.");

**this**.day = 1;

**this**.month = 1;

**this**.year = 2000;

}

}

**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** **true**;

}

**private** **int** daysInMonth(**int** month, **int** year) {

**switch** (month) {

**case** 1: **case** 3: **case** 5: **case** 7: **case** 8: **case** 10: **case** 12: **return** 31;

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

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

**default**: **return** 0;

}

}

**private** **boolean** isLeapYear(**int** year) {

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

}

**public** **void** addDays(**int** daysToAdd) {

**while** (daysToAdd > 0) {

**if** (day + daysToAdd <= daysInMonth(month, year)) {

day += daysToAdd;

**break**;

} **else** {

daysToAdd -= (daysInMonth(month, year) - day + 1);

day = 1;

**if** (month == 12) {

month = 1;

year++;

} **else** {

month++;

}

}

}

}

**public** **void** display() {

System.***out***.println(day + "/" + month + "/" + year);

}

}

**class** Employee1 {

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

String employeeName;

Date1 joiningDate;

// Constructor to initialize Employee1

**public** Employee1(**int** employeeNumber, String employeeName, Date1 joiningDate) {

**this**.employeeNumber = employeeNumber;

**this**.employeeName = employeeName;

**this**.joiningDate = joiningDate;

}

// Method to display employee information

**public** **void** displayEmployeeInfo() {

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

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

System.***out***.print("Joining Date: ");

joiningDate.display();

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

}

}

**package** org.day2assignment;

**public** **class** Main3 {

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

Employee1 emp1 = **new** Employee1(101, "John Doe", **new** Date1(8,12,2002));

Employee1 emp2 = **new** Employee1(102, "Jane Smith", **new** Date1(6,5,2008));

Employee1 emp3 = **new** Employee1(103, "Robert Brown", **new** Date1(18,9,2001));

Employee1 emp4 = **new** Employee1(104, "Emily Davis", **new** Date1(25,2,2010));

Employee1 emp5 = **new** Employee1(105, "David Wilson", **new** Date1(30, 10, 2021));

emp1.displayEmployeeInfo();

emp2.displayEmployeeInfo();

emp3.displayEmployeeInfo();

emp4.displayEmployeeInfo();

emp5.displayEmployeeInfo();

emp1.joiningDate.addDays(50);

System.***out***.println("Updated Joining Date for " + emp1.employeeName + ": ");

emp1.joiningDate.display();

}

}

**Output:**

Employee Number: 101

Employee Name: John Doe

Joining Date: 8/12/2002

Employee Number: 102

Employee Name: Jane Smith

Joining Date: 6/5/2008

Employee Number: 103

Employee Name: Robert Brown

Joining Date: 18/9/2001

Employee Number: 104

Employee Name: Emily Davis

Joining Date: 25/2/2010

Employee Number: 105

Employee Name: David Wilson

Joining Date: 30/10/2021

Updated Joining Date for John Doe:

27/1/2003

**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:**

**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.**

**package** org.day2assignment;

**public** **class** Vehicle {

String manufacturer,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:"+manufacturer);

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

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

}

}

**Subclasses:**

**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().**

**package** org.day2assignment;

**public** **class** Car **extends** Vehicle {

**public** **double** seatingcapacity;

**public** **double** getSeatingcapacity() {

**return** seatingcapacity;

}

**public** **void** setSeatingcapacity(**double** seatingcapacity) {

**this**.seatingcapacity = seatingcapacity;

}

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

System.***out***.println("the seating capacity:"+seatingcapacity);

}

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

System.***out***.println("accelerante the vehicle");

}

**public** **void** Break()

{

System.***out***.println("break stops the vehicle");

}

}

**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().**

**package** org.day2assignment;

**public** **class** Motorcycle **extends** Vehicle {

**public** **double** enginecapacity;

**public** **double** getEnginecapacity() {

**return** enginecapacity;

}

**public** **void** setEnginecapacity(**double** enginecapacity) {

**this**.enginecapacity = enginecapacity;

}

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

System.***out***.println("the engine capacity:"+enginecapacity);

}

**public** **void** startengine() {

System.***out***.println("vehicle started");

}

**public** **void** s()

{

System.***out***.println("vehicle stops");

}

}

**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().**

**package** org.day2assignment;s

**public** **class** Truck **extends** Vehicle {

**public** **double** cargocapacity;

**public** **double** getCargocapacity() {

**return** cargocapacity;

}

**public** **void** setCargocapacity(**double** cargocapacity) {

**this**.cargocapacity = cargocapacity;

}

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

System.***out***.println("the cargo capacity:"+cargocapacity);

}

**public** **void** loadcargo()

{

System.***out***.println("cargo is loaded");

}

**public** **void** unloadcargo() {

System.***out***.println("cargo is unloaded");

}

}

**Main Class:**

**• Create instances of Car, Motorcycle, and Truck.**

**• Set values for their attributes and demonstrate the usage of methods like displayDetails(), accelerate(), brake(), startEngine(), stopEngine(), loadCargo(), unloadCargo().**

**• Ensure that each vehicle type behaves correctly based on its specific attributes and operations.**

package org.day2assignment;

public class Main {

public static void main(String[] args) {

Car ob = new Car();

Motorcycle ob1 = new Motorcycle();

Truck ob2 = new Truck();

ob.setSeatingcapacity(4);

ob1.setEnginecapacity(5000);

ob2.setCargocapacity(200);

ob.displayDetails();

ob.accelerate();

ob.Break();

ob1.displayDetails();

ob1.startengine();

ob1.stopengine();

ob2.displayDetails();

ob2.loadcargo();

ob2.unloadcargo();

}

}

**Output:**

the seating capacity:4.0

the engine capacity:5000.0

the cargo capacity:200.0

**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().**

**package** org.day2assignment;

**public** **abstract** **class** Shape{

**public** **abstract** **void** calculateArea();

}

**• Implement classes for each shape extending Shape and provide necessary attributes (radius and**

**sideLength for circle, rectangle, and triangle respectively).**

**class** Circle **extends** Shape{

**public** **double** pi=3.14;

**public** **double** radius;

**public** Circle(**double** radius){

**this**.radius = radius;

}

**public** **void** calculateArea(){

System.***out***.println("the area of circle:"+(pi\*radius\*radius));

}

}

**class** Rectangle2 **extends** Shape{

**public** **double** Length;

**public** **double** Breadth;

**public** Rectangle2(**double** length,**double** breadth){

**this**.Length = length;

**this**.Breadth = breadth;

}

**public** **void** calculateArea(){

System.***out***.println("the area of rectangle:"+(Length\*Breadth));

}

}

**class** Triangle **extends** Shape{

**public** **double** h=0.5;

**public** **double** Base;

**public** **double** Height;

**public** Triangle(**double** base,**double** height){

**this**.Base = base;

**this**.Height = height;

}

**public** **void** calculateArea(){

System.***out***.println("the area of triangle:"+(h\*Base\*Height));

}

}

**• Use abstraction to ensure that each shape class implements its own logic to calculate the area**

**based on its specific attributes**

**package** org.day2assignment;

**import** java.util.Scanner;

**public** **class** Main1 {

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

Circle ci = **new** Circle(5);

Rectangle2 rect = **new** Rectangle2(1,19);

Triangle tri = **new** Triangle(5.4,7.8);

ci.calculateArea();

rect.calculateArea();

tri.calculateArea();

}

}

**Output:**

the area of circle:78.5

the area of rectangle:19.0

the area of triangle:21.060000000000002

**Problem Statement 4: Implement the Static Classes and Methods in Java**

package org.day2assignment;

public class Employee {

private String name;

private int points;

public Employee(String name, int points) {

this.name = name;

this.points = points;

}

public String getname() {

return name;

}

public void setname(String name) {

this.name=name;

}

public int getpoints() {

return points;

}

public void setname(int points) {

this.points=points;

}

}

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 calculatePerformance(Employee e) {

int points = e.getpoints();

if (points>80 || points<100) {

return *outstanding*;

}

else if(points>60 || points <79) {

return *good*;

}

else if(points>50 || points<35) {

return *average*;

}

else if(points <35) {

return *poor*;

}

return points;

}

}

package org.day2assignment;

public class main2 {

private static int *employeeCount* = 0;

public static void main(String[] args) {

Employee[] employees = new Employee[]{

new Employee("John Doe", 95),

new Employee("Jane Smith", 7),

new Employee("Alice Johnson", 65)};

*employeeCount* = employees.length;

System.*out*.println("Total number of Employees: " + *employeeCount*+" and their ratings are");

for (Employee emp : employees) {

System.*out*.println(emp.getname() + " has performed with rating " + PerformanceRating.*calculatePerformance*(emp));

System.*out*.println("==================================================================");

}

}

}

**Output:**

Total number of Employees: 3 and their ratings are

Oliver has performed with rating 5

==================================================================

Jayden has performed with rating 2

==================================================================

Daniel has performed with rating 4

==================================================================