**Level 1: Easy**

1. Write a program to print the area of a rectangle by creating a class named 'Area'

having two methods. The first method named 'setDim' takes the length and breadth

of the rectangle as parameters and the second method named 'getArea' returns the

area of the rectangle. The length and breadth of the rectangle are entered through

the keyboard.

**package** com.oopsassignment;

**import** java.util.\*;

**class** Area {

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

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

**public** **void** setDim(**double** length, **double** breadth) {

**this**.length = length;

**this**.breadth = breadth;

}

**public** **double** getArea() {

**return** length \* breadth;

}

}

**public** **class** Easyy1 {

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

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

Area rectangle = **new** Area();

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

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

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

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

rectangle.setDim(length, breadth);

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

}

}

Output:

Enter length of rectangle: 4

Enter breadth of rectangle: 5

Area of Rectangle: 20.0

2. Write a program to print the area and perimeter of a triangle having sides of 3, 4,

and 5 units by creating a class named 'Triangle' without any parameter in its

constructor.

**package** com.oopsassignment;

**class** Triangle {

**private** **double** side1 = 3;

**private** **double** side2 = 4;

**private** **double** side3 = 5;

**public** **double** getPerimeter() {

**return** side1 + side2 + side3;

}

**public** **double** getArea() {

**double** s = getPerimeter() / 2;

**return** Math.*sqrt*(s \* (s - side1) \* (s - side2) \* (s - side3));

}

}

**public** **class** Eassy2 {

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

Triangle triangle = **new** Triangle();

System.***out***.println("Perimeter of Triangle: " + triangle.getPerimeter());

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

}

}

Output:

Perimeter of Triangle: 12.0

Area of Triangle: 6.0

3. Write a program to print the area of two rectangles having sides (4,5) and (5,8)

respectively by creating a class named 'Rectangle' with a method named 'Area' which

returns the area and length and breadth passed as parameters to its constructor.

**public** **class** Easy\_3 {

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

Rectangle rect1 = **new** Rectangle(4, 5);

Rectangle rect2 = **new** Rectangle(5, 8);

System.***out***.println("Area of Rectangle 1: " + rect1.getArea());

System.***out***.println("Area of Rectangle 2: " + rect2.getArea());

}

}

Output:

Area of Rectangle 1: 20.0

Area of Rectangle 2: 40.0

4. Print the average of three numbers entered by the user by creating a class named

'Average' having a method to calculate and print the average.

**package** com.oopsassignment;

**import** java.util.\*;

**class** Average {

**public** **void** calculateAverage(**double** num1, **double** num2, **double** num3) {

**double** average = (num1 + num2 + num3) / 3;

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

}

}

**public** **class** Easy\_4 {

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

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

Average avg = **new** Average();

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

**double** num1 = sc.nextDouble();

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

**double** num2 = sc.nextDouble();

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

**double** num3 = sc.nextDouble();

avg.calculateAverage(num1, num2, num3);

}

}

Output:

Enter first number: 4

Enter second number: 3

Enter third number: 5

Average: 4.0

5. Write a class called Crop and help the farmer to separate the vegetables on his farm.

The farm contains vegetables like carrots, Brinjal, and Potato. Separate the

vegetables and print the result in the following format using constructor overloading

C 15

P 25

B 30

**package** com.oopsassignment;

**class** Crop {

**private** **char** vegetableType;

**private** **int** quantity;

**public** Crop(**char** vegetableType, **int** quantity) {

**this**.vegetableType = vegetableType;

**this**.quantity = quantity;

}

**public** **void** printDetails() {

System.***out***.println(vegetableType + " " + quantity);

}

}

**public** **class** Easy\_5 {

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

Crop carrot = **new** Crop('C', 15);

Crop potato = **new** Crop('P', 25);

Crop brinjal = **new** Crop('B', 30);

carrot.printDetails();

potato.printDetails();

brinjal.printDetails();

}

}

Output:

C 15

P 25

B 30

Level 2: Medium

1. Write a program by creating an 'Employee' class having the following methods and

print the final salary.

• 'getInfo()' which takes the salary, and number of hours of work per day of the

employee as a parameter

• 'AddSal()' which adds Rs.800 to the salary of the employee if it is less than

Rs.40,000.

• 'AddWork()' which adds Rs.400 to the salary of an employee if the number of

hours of work per day is more than 6 hours.

**package** com.oopsassignment;

**import** java.util.\*;

**class** Employee {

**private** **double** salary;

**private** **int** workHours;

**public** **void** getInfo(**double** salary, **int** workHours) {

**this**.salary = salary;

**this**.workHours = workHours;

}

**public** **void** AddSal() {

**if** (salary < 40000) {

salary += 800;

}

}

**public** **void** AddWork() {

**if** (workHours > 6) {

salary += 400;

}

}

**public** **void** printSalary() {

System.***out***.println("Final Salary: " + salary);

}

}

**public** **class** Medium\_1 {

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

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

Employee emp = **new** Employee();

System.***out***.print("Enter salary: ");

**double** salary = sc.nextDouble();

System.***out***.print("Enter work hours per day: ");

**int** workHours = sc.nextInt();

emp.getInfo(salary, workHours);

emp.AddSal();

emp.AddWork();

emp.printSalary();

}

}

Output:

Enter salary: 25000

Enter work hours per day: 8

Final Salary: 26200.0

2. The machine has two main components: A built-in cash register and several

dispensers to hold and release the products.

Define class cashRegister with the following descriptions :

• Private Members: cashOnHand of type integer

• Public Members: A default constructor cashRegister() sets the cash in the register

to 100.

• A constructor cashRegister(int) sets the cash in the register to a specific amount.

• A function getCurrentBalance() which returns value of cashOnHand

• A function acceptAmount(int) to receive the amount deposited by the customer

and update the amount in the register.

class CashRegister {

    private int cashOnHand;

    public CashRegister() {

        this.cashOnHand = 100;

    }

    public CashRegister(int amount) {

        this.cashOnHand = amount;

    }

    public int getCurrentBalance() {

        return cashOnHand;

    }

    public void acceptAmount(int amount) {

        cashOnHand += amount;

    }

}

public class Medium\_2 {

    public static void main(String[] args) {

        CashRegister register = new CashRegister();

        System.out.println("Initial Balance: " + register.getCurrentBalance());

        register.acceptAmount(500);

        System.out.println("Balance after accepting amount: " + register.getCurrentBalance());

    }

}

Output:

Initial Balance: 100

Balance after accepting amount: 600

3. Create a class called Singer. It should be designed as follows: It contains:

• Three private instance variables: name(String), email (String), and gender (char

of either ‘m’ or ‘f’).

• One constructor to initialize the name, email, and gender with the given values.

And, a class called MusicAlbum is designed as follows: It contains:

• Four private instance variables: name (String), Singer of the album (of the class

Singer you have just created), price (double), and qtyInStock (int).

Assuming that each MusicAlbum is composed by one Singer. Perform the following:

• One constructor which constructs an instance with the values given.

• Getters and setters: getName(), getSinger(), getPrice(), setPrice(),

getQtyInStock(), setQtyInStock().

There is no setter for name and author.

Write the class MusicAlbumInfo(which uses the“Singer” class written earlier).

Try:

• Printing the album name, price and qtyinstock from MusicAlbumInfo object.

• After obtaining the “Singer” object, print the Singer (name, email & gender) of

the MusicAlbum.

**package** com.oopsassignment;

**import** java.util.\*;

**class** Singer {

**private** String name;

**private** String email;

**private** **char** gender;

**public** Singer(String name, String email, **char** gender) {

**this**.name = name;

**this**.email = email;

**this**.gender = gender;

}

**public** String getName() {

**return** name;

}

**public** String getEmail() {

**return** email;

}

**public** **char** getGender() {

**return** gender;

}

}

**class** MusicAlbum {

**private** String name;

**private** Singer singer;

**private** **double** price;

**private** **int** qtyInStock;

**public** MusicAlbum(String name, Singer singer, **double** price, **int** qtyInStock) {

**this**.name = name;

**this**.singer = singer;

**this**.price = price;

**this**.qtyInStock = qtyInStock;

}

**public** String getName() {

**return** name;

}

**public** Singer getSinger() {

**return** singer;

}

**public** **double** getPrice() {

**return** price;

}

**public** **void** setPrice(**double** price) {

**this**.price = price;

}

**public** **int** getQtyInStock() {

**return** qtyInStock;

}

**public** **void** setQtyInStock(**int** qtyInStock) {

**this**.qtyInStock = qtyInStock;

}

}

**public** **class** Medium\_3 {

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

Singer singer = **new** Singer("AR", "ar@example.com", 'm');

MusicAlbum album = **new** MusicAlbum("Greatest Hits", singer, 20.99, 60);

System.***out***.println("Album Name: " + album.getName());

System.***out***.println("Price: " + album.getPrice());

System.***out***.println("Quantity in Stock: " + album.getQtyInStock());

System.***out***.println("Singer Name: " + album.getSinger().getName());

System.***out***.println("Singer Email: " + album.getSinger().getEmail());

System.***out***.println("Singer Gender: " + album.getSinger().getGender());

}

}

Output:

Album Name: Greatest Hits

Price: 20.99

Quantity in Stock: 60

Singer Name: AR

Singer Email: ar@example.com

Singer Gender: m

4. Define a class in with following description:

Private Members

A data member Flight number of type integer

A data member Destination of type string

A data member Distance of type float

A data member Fuel of type float

A member function CALFUEL() to calculate the value of Fuel as per the following

criteria

Distance Fuel

<=1000 500

more than 1000 and <=2000 1100

more than 2000 2200

Public Members

A function FEEDINFO() to allow user to enter values for Flight Number, Destination,

Distance & call function CALFUEL() to calculate the quantity of Fuel

A function SHOWINFO() to allow user to view the content of all the data members

**package** com.oopsassignment;

**import** java.util.\*;

**class** Flight {

**private** **int** flightNumber;

**private** String destination;

**private** **float** distance;

**private** **float** fuel;

**private** **void** CALFUEL() {

**if** (distance <= 1000) {

fuel = 500;

} **else** **if** (distance > 1000 && distance <= 2000) {

fuel = 1100;

} **else** {

fuel = 2200;

}

}

**public** **void** FEEDINFO() {

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

System.***out***.print("Enter Flight Number: ");

flightNumber = sc.nextInt();

sc.nextLine();

System.***out***.print("Enter Destination: ");

destination = sc.nextLine();

System.***out***.print("Enter Distance: ");

distance = sc.nextFloat();

CALFUEL();

}

**public** **void** SHOWINFO() {

System.***out***.println("Flight Number: " + flightNumber);

System.***out***.println("Destination: " + destination);

System.***out***.println("Distance: " + distance);

System.***out***.println("Fuel Required: " + fuel);

}

}

**public** **class** Medium\_4 {

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

Flight flight = **new** Flight();

flight.FEEDINFO();

flight.SHOWINFO();

}

}

Output:

Enter Flight Number: 50

Enter Destination: chennai

Enter Distance: 800

Flight Number: 50

Destination: chennai

Distance: 800.0

Fuel Required: 500.0

Hard:

1.Assume you see a tollbooth at a highway. It is rule that all vehicles passing by the booth are expected to pay the following amount: In few cases, at times some vehicle goes by without paying toll. The tollbooth keeps track of the number of vehicle that have passed by, and of the total amount of money collected. Implement this tollbooth with a class called tollbooth. The two data items are a type unsigned int array to hold the number of Vehicles, and a type double to hold the total amount of money collected. A constructor initializes both of these to 0.A member function called payingVehicle() increments the car total and adds 0.50 to the cash total. Another function, called nopayVehicle(), increments the Vehicle total but adds nothing to the cash total. Finally, a member function called display() displays the two totals. Include a program to test this class. This program should allow the user to push one key to count a paying Vehicle, and another to count a non-paying Vehicle. Develop application must print out the total Vehicles and total cash and then exit.

**package** oops.practice;

**import** java.util.\*;

**class** Tollbooth {

**private** **int** totalVehicles;

**private** **double** totalCash;

**public** Tollbooth() {

totalVehicles = 0;

totalCash = 0.0;

}

**public** **void** payingVehicle() {

totalVehicles++;

totalCash += 0.50;

}

**public** **void** nopayVehicle() {

totalVehicles++;

}

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

System.***out***.println("Total Vehicles: " + totalVehicles);

System.***out***.println("Total Cash: ₹" + totalCash);

}

}

**public** **class** Toll {

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

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

Tollbooth tollbooth = **new** Tollbooth();

**while** (**true**) {

System.***out***.println("\nEnter choice:");

System.***out***.println("1 - Paying Veh");

System.***out***.println("2 - Non paying Veh");

System.***out***.println("3 - total");

System.***out***.print("Choice: ");

**int** choice = sc.nextInt();

**if** (choice == 1) {

tollbooth.payingVehicle();

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

} **else** **if** (choice == 2) {

tollbooth.nopayVehicle();

System.***out***.println("Non-paying vehicle");

} **else** **if** (choice == 3) {

tollbooth.display();

**break**;

} **else** {

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

}

}

sc.close();

}

}

Output:

Enter choice:

1 - Paying Veh

2 - Non paying Veh

3 - total

Choice:

1

Paying vehicle

Enter choice:

1 - Paying Veh

2 - Non paying Veh

3 - total

Choice: 2

Non-paying vehicle

Enter choice:

1 - Paying Veh

2 - Non paying Veh

3 - total

Choice: 3

Total Vehicles: 2

Total Cash: ₹0.5

1. Declare a class BrowsingCentre. Declare a class Member having memId, memName, MemLocation. PrimeMembers are charged weekly rent regardless of the number of hours. NormalMembers are charged hourly. Design a class such that it has components like memberProfile, getUsage, setBill based on the usage. Define appropriate constructors, the Main method to store array of ‘N’ PrimeMembers and NormalMembers accessing their pay details. A method to find and display details of HourlyMembers who are paying more than 2,000/per month. (1 hour charge is Rs.50) Provide code for this scenario.

**package** oops.practice;

**import** java.util.\*;

**class** Member {

**int** memid;

String memName;

String memLocation;

**public** Member(**int** memid, String memName, String memLocation) {

**this**.memid = memid;

**this**.memName = memName;

**this**.memLocation = memLocation;

}

**public** **void** memberProfile() {

System.***out***.println("ID: " + memid + " Name: " + memName + " Location: " + memLocation);

}

**public** **double** getUsage() {

**return** 0;

}

**public** **double** setBill() {

**return** 0;

}

}

**class** PrimeMember **extends** Member {

**double** weekRent;

**public** PrimeMember(**int** memid, String memName, String memLocation, **double** weekRent) {

**super**(memid, memName, memLocation);

**this**.weekRent = weekRent;

}

@Override

**public** **void** memberProfile() {

**super**.memberProfile();

System.***out***.println("Prime Member Weekly Rent: Rs " + weekRent + " Monthly Bill: Rs " + setBill());

}

@Override

**public** **double** setBill() {

**return** weekRent \* 4;

}

}

**class** NormalMember **extends** Member {

**int** hour;

**double** rate = 50;

**public** NormalMember(**int** memid, String memName, String memLocation, **int** hour) {

**super**(memid, memName, memLocation);

**this**.hour = hour;

}

@Override

**public** **void** memberProfile() {

**super**.memberProfile();

System.***out***.println("Normal Member Hours Used: " + hour + " Monthly Bill: Rs " + setBill());

}

@Override

**public** **double** getUsage() {

**return** hour;

}

@Override

**public** **double** setBill() {

**return** hour \* rate;

}

}

**public** **class** BrowsingCentre {

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

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

System.***out***.println("Enter number of members:");

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

Member[] members = **new** Member[n];

NormalMember[] highpayer = **new** NormalMember[n];

**int** count = 0;

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

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

System.***out***.println("Enter ID:");

**int** memid = sc.nextInt();

sc.nextLine();

System.***out***.println("Enter Name:");

String memName = sc.nextLine();

System.***out***.println("Enter Location:");

String memLocation = sc.nextLine();

System.***out***.println("1-prime 2-normal");

**int** type = sc.nextInt();

**if** (type==1) {

System.***out***.print("weekly rent: ");

**double** weeklyRent = sc.nextDouble();

members[i] = **new** PrimeMember(memid, memName, memLocation, weeklyRent);

} **else** **if** (type==2) {

System.***out***.print("hour in a Month: ");

**int** hoursUsed = sc.nextInt();

NormalMember nm = **new** NormalMember(memid, memName, memLocation, hoursUsed);

members[i] = nm;

**if** (nm.setBill()>2000) {

highpayer[count++] = nm;

}

} **else** {

System.***out***.println("invalid type");

}

}

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

System.***out***.println("ID: " + highpayer[i].memid + " Name: " + highpayer[i].memName);

}

sc.close();

}

}

Output:

Enter number of members:

2

Enter details for member 1

Enter ID:

1

Enter Name:

david

Enter Location:

salem

1-prime 2-normal

2500

invalid type

Enter details for member 2

Enter ID:

2

Enter Name:

john

Enter Location:

cbe

1-prime 2-normal

2

hour in a Month: 500

ID: 2 Name: john

3.Assume you see a ParkingLot at a mall. It is a rule that all vehicles parked are expected to pay the following amount: In few cases, at times some vehicle goes by without paying the bill. The parking lot system keeps track of the number of vehicle that have passed by, and of the total amount of money collected. Implement this with a class called ParkingLot. The two data items are a type unsigned int array to hold the number of Vehicles, and a type double to hold the total amount of money collected. A constructor initializes both of these to 0.A member function called payingVehicle() increments the car total and adds 0.50 to the cash total. Another function, called nopayVehicle(), increments the Vehicle total but adds nothing to the cash total. Finally, a member function called display() displays the two totals. Include a program to test this class. This program should allow the user to push one key to count a paying Vehicle, and another to count a non-paying Vehicle. Develop application must print out the total Vehicles and total cash and then exit.

**package** oops.practice;

**import** java.util.\*;

**class** Tollbooth {

**private** **int** totalVehicles;

**private** **double** totalCash;

**public** Tollbooth() {

totalVehicles = 0;

totalCash = 0.0;

}

**public** **void** payingVehicle() {

totalVehicles++;

totalCash += 0.50;

}

**public** **void** nopayVehicle() {

totalVehicles++;

}

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

System.***out***.println("Total Vehicles: " + totalVehicles);

System.***out***.println("Total Cash: ₹" + totalCash);

}

}

**public** **class** Toll {

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

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

Tollbooth tollbooth = **new** Tollbooth();

**while** (**true**) {

System.***out***.println("\nEnter choice:");

System.***out***.println("1 - Paying Veh");

System.***out***.println("2 - Non paying Veh");

System.***out***.println("3 - total");

System.***out***.print("Choice: ");

**int** choice = sc.nextInt();

**if** (choice == 1) {

tollbooth.payingVehicle();

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

} **else** **if** (choice == 2) {

tollbooth.nopayVehicle();

System.***out***.println("Non-paying vehicle");

} **else** **if** (choice == 3) {

tollbooth.display();

**break**;

} **else** {

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

}

}

sc.close();

}

}

Output:

Enter choice:

1 - Paying Veh

2 - Non paying Veh

3 - total

Choice:

1

Paying vehicle

Enter choice:

1 - Paying Veh

2 - Non paying Veh

3 - total

Choice: 2

Non-paying vehicle

Enter choice:

1 - Paying Veh

2 - Non paying Veh

3 - total

Choice: 3

Total Vehicles: 2

Total Cash: ₹0.5

4. A rental machine shop, gives the construction machines to the mechanics in a rental

basics.Define a RentalMachine class which has drawMachine(), returnMachine(),

checkStaus() and reserveMachine() methods and necessary attributes. All the

methods are tagged with public in the following ways.

(a) Using drawMachine()-get the required Machine based on the name of the

machine.

(b) Using checkStatus()-user - Machine returned date details.

(c) Using reserveMachine()-block or reserve particular Machine for their account.

(d)Using returnMachine()-Return the drawn Machines. Implement this scenario and

create user friendly management application.

import java.util.\*;

class RentalMachine {

    private HashMap<String, Boolean> machines;

    private HashMap<String, String> reservationDetails;

    private HashMap<String, String> returnDates;

    public RentalMachine() {

        machines = new HashMap<>();

        reservationDetails = new HashMap<>();

        returnDates = new HashMap<>();

        machines.put("Excavator", true);

        machines.put("Bulldozer", true);

        machines.put("Crane", true);

    }

    public void drawMachine(String machineName) {

        if (machines.containsKey(machineName) && machines.get(machineName)) {

            machines.put(machineName, false);

            System.out.println(machineName + " has been rented.");

        } else {

            System.out.println(machineName + " is not available.");

        }

    }

    public void returnMachine(String machineName, String returnDate) {

        if (machines.containsKey(machineName)) {

            machines.put(machineName, true);

            returnDates.put(machineName, returnDate);

            System.out.println(machineName + " has been returned on " + returnDate);

        } else {

            System.out.println("Invalid machine name.");

        }

    }

    public void checkStatus(String machineName) {

        if (returnDates.containsKey(machineName)) {

            System.out.println(machineName + " was last returned on: " + returnDates.get(machineName));

        } else {

            System.out.println("No return record for " + machineName);

        }

    }

    public void reserveMachine(String machineName, String userName) {

        if (machines.containsKey(machineName) && machines.get(machineName)) {

            reservationDetails.put(machineName, userName);

            System.out.println(machineName + " has been reserved for " + userName);

        } else {

            System.out.println(machineName + " is currently unavailable for reservation.");

        }

    }

}

public class Hard\_4 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        RentalMachine rental = new RentalMachine();

        char choice;

        do {

            System.out.println("\nChoose an option:");

            System.out.println("1. Rent a machine");

            System.out.println("2. Return a machine");

            System.out.println("3. Check machine status");

            System.out.println("4. Reserve a machine");

            System.out.println("5. Exit");

            choice = sc.next().charAt(0);

            switch (choice) {

                case '1':

                    System.out.print("Enter machine name to rent: ");

                    String rentMachine = sc.next();

                    rental.drawMachine(rentMachine);

                    break;

                case '2':

                    System.out.print("Enter machine name to return: ");

                    String returnMachine = sc.next();

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

                    String returnDate = sc.next();

                    rental.returnMachine(returnMachine, returnDate);

                    break;

                case '3':

                    System.out.print("Enter machine name to check status: ");

                    String checkMachine = sc.next();

                    rental.checkStatus(checkMachine);

                    break;

                case '4':

                    System.out.print("Enter machine name to reserve: ");

                    String reserveMachine = sc.next();

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

                    String userName = sc.next();

                    rental.reserveMachine(reserveMachine, userName);

                    break;

                case '5':

                    System.out.println("Exiting...");

                    break;

                default:

                    System.out.println("Invalid choice. Please try again.");

            }

        } while (choice != '5');

    }

}