

LAB ASSIGNMENT-2

Q1. Define a class 'Person' with two instance variables : name and age. and two member methods : setData() : set the details of the person
displayData() : display the details of the person.

Now, create two objects of class person and initialize one object value directly (by using the dot(.) operator name: "Rohan" and age : 20). Accept your name and name through keyboard and set them to another object using setData() method. Now, display both the member variables using the displayData() method. Also, check who is younger.

Ans

```
import java.util.*;
```

```
class Person
```

```
{
```

```
    String name;
```

```
    int age;
```

```
    void setData()
```

```
    {
```

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

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

```
        this.name = sc.next();
```

```
        System.out.print("Enter age :");
```

```
        this.age = sc.nextInt();
```

```
    }
```

```
        void displayData()
        {
            System.out.println(name + " " +
            age);
        }
    }

    public class testQ1
    {
        public static void main (String[] args)
        {
            person p1 = new Person();
            p1.name = "Rohan";
            p1.age = 20;
            Person p2 = new Person();
            p2.setData();
            p1.displayData();
            p2.displayData();
            if (p1.age < p2.age)
            {
                System.out.println(p1.nameage + " is younger");
            }
            else
            {
                System.out.println(p2.name + " is
                younger");
            }
        }
    }
}
```

Output:

Enter name: Rakesh

Enter age: 26

Rahul is younger. ~~than~~

Q2: Define a class called Complex with instance variables real, img, and instance method

- void setData()
- void display()
- public Complex add(Complex c1, Complex c2)

Write a ~~new~~ main method in a separate class to implement the addition of two complex no.s with the given method signature as above.

Ans.

```
class Complex
```

```
{
```

```
    int real;
```

```
    int img;
```

```
    void setData()
```

```
{
```

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

```
        System.out.print("Enter real value:");
```

```
        this.real = sc.nextInt();
```

```
        System.out.print("Enter imaginary value:");
```

```
        this.img = sc.nextInt();
```

```
}
```

```
    void display()
```

```
{
```

```
        System.out.println(real + "+" + img + "i");
```

```
}
```

```
    public Complex add(Complex c1, Complex c2)
```

```
{
```

```
        Complex sum = new Complex();
```



```
        sum.real = c1.real + c2.real;
        sum.img = c1.img + c2.img;
        return sum;
    }
}

public class Q2
{
    public static void main (String[] args)
    {
        Complex c1 = new Complex();
        c1.setData();
        Complex c2 = new Complex();
        c2.setData();
        c1.display();
        c2.display();
        System.out.println("Sum is : ");
        Complex sum = new Complex();
        sum = sum.add(c1, c2);
        System.out.println("Sum is :");
        sum.display();
        sc.close();
    }
}
```

output : Enter real value : 5
Enter imaginary value : 3
Enter real value : 7
Enter imaginary value : 8

5+3i
7+8i
Sum is 12+11i

Question 3: In a supermarket each product is having minimum details like prodId, price, quantity. This is used during the billing process. Keeping this in mind prepare a class named

'Product' having member variable:-

- prodId, price, quantity
- a static variable totalPrice.

Initialize the value of product through parameterized constructor. It consists of a display() method to display the value of instance variables.

A person went to market to buy 5 different products. Using the above mentioned class, display the details of products that the person has purchased. Also determine how much total amount the person will pay for the purchase of 5 products.

Ans.

```
import java.util.*;
```

```
class Product
```

```
{
```

```
    int prodId, price, quantity;
```

```
    static int totalPrice;
```

```
    Product (int pid, int p, int q)
```

```
{
```

```
        prodId = pid;
```

```
        price = p;
```

```
        quantity = q;
```

```
}  
void display()  
{  
    System.out.println(prodId + " " + price + "  
    + quantity);  
}  
}  
public class Q3  
{  
    public static void main (String[] args)  
    {  
        Product p[] = new Product[5];  
        for (int i=0; i<5; i++)  
        {  
            System.out.println("Enter product ID,  
            price and Quantity:");  
            int pid = sc.nextInt();  
            int p = sc.nextInt();  
            int q = sc.nextInt();  
            p[i] = new Product(pid, p, q);  
        }  
        for (int i=0; i<5; i++)  
        {  
            p[i].display();  
        }  
        System.out
```

```
for (int i=0; i<5; i++)  
{  
    Product.totalprice += p[i].price * p[i].  
        quantity;  
}  
System.out.println("Total price is " +  
    Person.totalprice);  
}
```


Q4. Define a class Deposit. The instance variable of the class Deposit are mentioned below.

| <u>Instance variable</u> | <u>Datatype</u> |
|--------------------------|-----------------|
| Principal | Long |
| Time | Integer |
| Rate | Double |
| TotalAmt | Double |

Initialize the instance variables Principal, Time, Rate through constructor. Constructors are overloaded with following prototypes:

Constructor 1: Deposit()

Constructor 2: Deposit(long, int, double)

Constructor 3: Deposit(long, int)

Constructor 4: Deposit(long, double)

Apart from constructor, the other instance variable methods are:

i) display(): to display the value of instance variables.

ii) calcAmt(): to calculate the total amount.

$$\text{totalAmt} = \text{Principal} + \frac{\text{Principal} \times \text{Rate} \times \text{Time}}{100}$$

Ans. class Deposit

```
{  
    long principal;  
    int Time;  
    double Rate;  
    double totalAmt;  
    Deposit()  
    {  
        this.principal = 0;  
        this.rate = 0;  
        this.totalAmt = 0.0;  
    }  
}
```

Deposit (long principal, int time, double rate)

```
{  
    this.principal = principal;  
    this.time = time;  
    this.totalAmt = totalAmt;  
    this.rate = rate;  
    CalcAmt();  
}
```

Deposit (long Principal, int time)

```
{  
    this.principal = principal;  
    this.time = time;  
    this.rate = 7;  
    CalcAmt();  
}
```

Deposit (long Principal, double rate)

```
{  
    this.principal = principal;  
    this.rate = rate;  
    this this.time = 3;  
    CalcAmt();  
}
```

void display()

```
{  
    System.out.println("Principal is : " + principal);  
    System.out.println("Time is " + time);  
    System.out.println("Rate is " + rate);  
    System.out.println("Total Amount is " + totalAmt);  
}
```

```
void calcAmt()  
{  
    this.totalAmt = principal + (principal * rate * time) / 100;  
}  
}  
public class Q4  
{  
    public static void main (String[] args)  
    {  
        Deposit d1 = new Deposit ();  
        Deposit d2 = new Deposit (10000, 5.7, 1);  
        Deposit d3 = new Deposit (20000, 2);  
        Deposit d4 = new Deposit (5000, 7.1);  
        System.out.println ("Deposit 1: " + d1.display());  
        System.out.println ("Deposit 2: " + d2.display());  
        System.out.println ("Deposit 3: " + d3.display());  
        System.out.println ("Deposit 4: " + d4.display());  
    }  
}
```


Q5: Define a base class Person with instance variable name, age. The instance variables are initialized through constructors. The prototype of constructor is as below: Person (String, int).

Define a derived class Employee with instance variable Eid, salary. The instance variables are initialized through constructors. The prototype of the constructor is as below: Employee (String, int, int, double). Another instance method of Employee class is empDisplay() to display the information of employee details.

Ans. import java.util.*;

```
class Person
```

```
{
```

```
    String name;
```

```
    int age;
```

```
    Person (String name, int age)
```

```
{
```

```
        this.name = name;
```

```
        this.age = age;
```

```
}
```

```
}
```

```
class Employee extends Person
```

```
{
```

```
    int Eid;
```

```
    double salary;
```

```
    Employee (String name, int age, int int Eid,  
              double salary)
```

```
{
```

```
    super (name, age);
```

```
    this.Eid = Eid;
```



```
        this.salary = salary;
    }
    void display empDisplay()
    {
        S.o.pln("Employee ID: " + this.Eid);
        S.o.pln("Name" + this.name);
        S.o.pln("Age" + this.age);
        S.o.pln("Salary" + this.salary);
    }
}

public class QS
{
    public static void main (String[] args)
    {
        Scanner sc = new Scanner (System.in);
        System.out.println("Enter Employee ID:");
        int eid = sc.nextInt();
        System.out.println("Enter name:");
        String n = sc.next();
        System.out.println("Enter age:");
        int a = sc.nextInt();
        System.out.println("Enter salary");
        double sal = sc.nextDouble();
        Employee el = new Employee(n, a, eid, sal);
        el.empempDisplay();
    }
}
```

Q6. Create an abstract class Marks with three instance variable (markICP, markDSA and percentage) and an abstract method getPercentage(). Create two classes: CSE with instance variable algoDesign, and NonCSE with instance variable ~~engMech~~ engMechanics. Both classes inherit the abstract class Marks and override the abstract method getPercentage(). The constructor of class CSE takes the marks in three subjects (markICP, markDSA and algoDesign) as its parameter and the constructor of class NonCSE takes the marks in three subjects (MarkICP, MarkDSA, and engMechanics) as its parameter. Create an object for each of the two classes and print the percentage of marks for both students.

Ans.

abstract class Marks

```
{  
    int markICP, markDSA;  
    double percentage;  
    Marks (int ICP markICP, int markDSA)  
    {  
        this.markICP = markICP;  
        this.markDSA = markDSA;  
    }  
    abstract void getPercentage()  
}
```

class CSE extends Marks

```
{  
    int algoDesign;  
    CSE (int markICP, int markDSA, int algoDesign)  
    {  

```

```
        super(markICP, markDSA);
        this.algoDesign = algoDesign;
    }
    public void getPercentage()
    {
        percentage = ((markICP + markDSA + algoDesign) / 3.0) * 100;
        System.out.println("CSE student percentage " +
            percentage);
    }
}

class NonCSE extends Marks
{
    int engMechanics;
    public NonCSE(int markICP, int markDSA, int engMechanics)
    {
        super(markICP, markDSA);
        this.engMechanics = engMechanics;
    }
    public void getPercentage()
    {
        percentage = ((markICP + markDSA + engMechanics) / 3.0) * 100;
        System.out.println("Non CSE student percentage " +
            percentage);
    }
}

public class Q6
{
    public static void main(String args[])
    {
```



```
CSE cl = new CSE(84, 65, 90);
```

```
cl.getPercentage();
```

```
NonCSE ncl = new NonCSE(87, 54, 49);
```

```
ncl.getPercentage();
```

```
}
```

```
}
```


Q7. Define an interface DetailInfo to declare methods display() and count(). Another class Person contains a static data member maxcount, instance member name and method display() to display name of the person, count the no. of characters present in the name of the person.

Ans.

```
interface DetailInfo {
```

```
{  
    void display();
```

```
    void count();
```

```
}
```

```
class Person implements DetailInfo
```

```
{
```

```
    String name;
```

```
    static int maxCount;
```

```
    public Person (String name)
```

```
{
```

```
        this.name = name;
```

```
}
```

```
    public void count()
```

```
{
```

```
        maxCount = this.name.length();
```

```
}
```

```
    public void display()
```

```
{
```

```
        System.out.println("Name" + name + " " +
```

```
        "No. of characters present : " + maxCount);
```

```
}
```

```
}
```

```
public class Q7  
{  
    public static void main (String[] args)  
    {  
        Person p1 = new Person ("Washington");  
        p1.count();  
        p1.display();  
    }  
}
```

Q8. Design a package that contains two classes Student and Test. The Student class has data members as name, roll and instance methods inputDetails() and showDetails(). Similarly, the Test class has data members as mark1, mark2, and instance methods inputDetails(), showDetails(). Student is extended by ~~test~~ Test. Another package carry interface Sports with two attributes score1 and score2. Find the grand total mark and score in another class.

Ans. package P1;
class Student
{
String name;
int roll;
public void inputDetails (String name, int roll)
{
this.name = name;
this.roll = roll;
}
public void showDetails ()
{
System.out.println("Name" + name + " " +
"roll no " + roll);
}
}
public class Test extends Student
{
public int mark1;
public int mark2;

```
public void inputDetails (int m1, int m2)
{
    this.mark1 = m1;
    this.mark2 = m2;
}
public void showDetails()
{
    super.showDetails();
    System.out.println(" Mark1 : " + mark1 + " "
        + " Mark 2 : " + m2 mark2);
}
}
```

Package P2;

public interface Sports

```
{
    int score1 = 90;
    int score2 = 100;
}
```

import java.P1.*;

import java.P2.*;

public class Q8 extends Test implements Sports

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

```
{
    Test m1 = new Test();
    m1.inputDetails ("Rahul", 38);
    m1.inputDetails (90, 95);
}
```



```
m1.showDetails();  
int Totalmarks = m1.mark1 + m1.mark2;  
System.out.println("Total marks" + Totalmarks);  
int Totalscore = score1 + score2;  
System.out.println("Total score:" + Totalscore);  
}  
}
```