Course Code : CSE2008   Course Title : Programming in java

Lab sheet 1

Programs for Module 1

1. In an Examination hall students of second and third year sit in odd and even number positions consecutively. The teacher asks one of the students to find the sum of even and odd positions separately. The total seating capacity of an exam hall is 100. Develop a software application to perform the above operations. (Task01 – L1)

Solution:

```
public class Odd_Even
{
public static void main(String args[])
{
int sumeven=0, sumodd=0;
for(int i=1;i<=100;i++)
        {
                if(i%2==0)
                {
                sumeven = sumeven+i;
                }
                else
                {
                sumodd = sumodd+i;
                }
         }
System.out.println("The sum of even positions is "+ sumeven);
System.out.println("The sum of odd positions is "+ sumodd);
}
}
```

Output

```
The sum of even positions is 2550
The sum of odd positions is 2500
```

2. A Teacher gives a box containing 5 chits to a student and asks him to pick one chit. Each chit contains a arithmetic operation to be performed by the student. The student can pick only one chit at a time. The 5 chits contain basic arithmetic operations like addition, subtraction, multiplication,division and modulus. Demonstrate the same situation using java program. (Task01-L1)

Solution:

```
import java.util.Scanner;
```

```
public class ArithmeticDemo {
        public static void main(String[] args) {
                int a,b,c=0;
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the values of a,b");
                a=sc.nextInt();
                b=sc.nextInt();
                System.out.println("1:Addition\n2:Substraction\n3:Multiplication\
n4:Division");
                System.out.println("Enter choice");
                int ch=sc.nextInt();
                switch(ch)
                {
                case 1: c=a+b;
                                break;
                case 2: c=a-b;
                                break;
                case 3: c=a*b;
                                break;
                case 4: if(b!=0)
                        {
                        c=a/b;
                        break;
                        }
                        else
                        {
                        System.out.println("Division is not possible");
                        break;
                        }
                default: System.out.println("Invalid choice");
                        break;
                }
                System.out.println("The result is "+c);
        }
        }
```

Output:

```
Enter the values of a,b
20 3
1:Addition
2:Substraction
3:Multiplication
4:Division
```

```
Enter choice
3
The result is 60
```

3. Swiggy App is celebrating its birthday, so it wants to give 80% discount for those customers who have order number which is prime and others will get 50% discount. Help the application to identify the discounted rate for customers. Write suitable java program. (Task01-L2)

Solution

```
import java.util.Scanner;
public class Prime {
public static void main(String[] args) {
        int n, i;
        boolean flag=false;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter order number");
        n = sc.nextInt();
        if(n==1)
                System.out.println("The customer will get 50% discount");
        else
        {
                for(i=2;i<n;i++)
                {
                        if(n%i==0)
                        {
                                System.out.println("The customer will get 50% discount");
                                flag=true;
                                break;
                        }
                }
                if(flag==false)
                        System.out.println("The customer will get 80% discount");
        }
        }
}
```

Output

```
Enter order number
9
The customer will get 50% discount
```

4. Shyam has coding competition in his school, he has to draw the following pattern. Suggest a solution to Shyam for winning the competition. (Task01 – L2)

```
    *
   ***
  *****
 *******
*********
```

Solution

```
import java.util.Scanner;
public class Pattern
{
public static void main(String args[]){
int n, s, i, j;
Scanner scan = new Scanner(System.in);
System.out.println("Enter number of rows: ");
n=scan.nextInt();
for(i = 1; i <= n; i++)
{
//for loop for displaying space
for(s = i; s < n; s++)
{
System.out.print(" ");
}
//for loop to display star equal to row number
for(j = 1; j <= (2 * i - 1); j++)
{
System.out.print("*");
}
// ending line after each row
System.out.println();
}
}
}
```

Output

```
Enter number of rows:
5
          *
         ***
        *****
       *******
      *********
```

5. Define a class called CommandLine, develop a program in Java that accepts two floats and integers as its command line arguments and perform addition, multiplication, subtraction and division and display the result. (Task02 – L1)

Solution:

```
class CommandLine{
public static void main(String args[]){
  int a,b;
  float c,d;
  a=Integer.parseInt(args[0]);
  b=Integer.parseInt(args[1]);
  c=Float.parseFloat(args[2]);
  d=Float.parseFloat(args[3]);
  System.out.println("Integer Arithmetic Operations\n");
  System.out.println("Addition is " +(a+b));
  System.out.println("Subtraction is " +(a-b));
  System.out.println("Division is " +(a/b));
  System.out.println("Multiplication is " +(a*b));

  System.out.println("Real Arithmetic Operations\n");
  System.out.println("Addition is " +(c+d));
  System.out.println("Subtraction is " +(c-d));
  System.out.println("Division is " +(c/d));
```

```
   System.out.println("Multiplication is " +(c*d));
}}
```

Output:

```
➢  javac CommandLine.java
➢  java CommandLine 10 20 6.5 4.5
   Integer Arithmetic Operations
   Addition: 30
   Subtraction: -10
   Multiplication: 200
   Division: 0

   Real Arithmetic Operations
   Addition: 12
   Subtraction: 2.0
   Multiplication: 29.25
   Division: 1.444
```

6. A teacher gave a project to student in the classroom to find the area of different shapes. A student has to find the area of different shapes based on the choice of different parameters and display the results. Demonstrate the same using Java Program (Task02 – L1)

Solution:

```
public class Students
{
   void area(int x)
   {
      System.out.println("The area of the square is "+Math.pow(x, 2)+" sq units");
   }
   void area(int x, int y)
   {
      System.out.println("The area of the rectangle is "+ (x*y) +" sq units");
   }
   void area(double x)
   {
      double z = 3.14 * x * x;
      System.out.println("The area of the circle is "+z+" sq units");
   }
    public static void main(String args[])
```

```
        {
          Students ob = new Students();
          ob.area(5);
          ob.area(11,12);
          ob.area(2.5);
        }
}
```

Output:

```
The area of the square is 25.0 sq units
The area of the rectangle is 132 sq units
The area of the circle is 19.625 sq units
```

7. Create a class named 'Rectangle' with two data members- length and breadth and a method to calculate the area which is 'length*breadth'. The class has three constructors which are (Task02 – L1)

i -having no parameter - values of both length and breadth are assigned zero.
ii - having two numbers as parameters - the two numbers are assigned as length and breadth respectively.
iii - having one number as parameter - both length and breadth are assigned that number.
Now, create objects of the 'Rectangle' class having none, one and two parameters and print their areas.

Solution

```
import java.util.*;
class Rectangle
{
int length;
int breadth;
void area() // method
{
int area=length*breadth;
System.out.println(area);
}
Rectangle() // constructor with no parameter
{
length=0;
breadth=0;
}
Rectangle(int len)// constructor with one parameter
{
length=len;
breadth=len;
}
Rectangle(int l,int b) // constructor with two parameters
{
length=l;
breadth=b;
}
Rectangle(Rectangle r) //copy constructor
{
length=r.length;
breadth=r.breadth;
}
public static void main(String args[])
{
int len, br;
Rectangle r1= new Rectangle();
System.out.print("Area of object r1 : ");
r1.area();
Scanner input= new Scanner(System.in);
System.out.println("Enter the length and breadth");
len=input.nextInt();
br=input.nextInt();
Rectangle r2= new Rectangle(len, br);
System.out.print("Area of object r2 : ");
r2.area();
Rectangle r3= new Rectangle(50);
System.out.print("Area of object r3 : ");
r3.area();
Rectangle r4= new Rectangle(r2);
System.out.print("Area of object r4 : ");
r4.area();}}
```

Output:

```
Area of object r1 : 0
Enter the length and breadth
2 3
Area of object r2 : 6
Area of object r3 : 2500
Area of object r4 : 6
```

8. AVR apartments uses a common water tank to supply water for every block. The D block uses 20 litres of water and E block uses 40 litres of water every time. Each time the water level of the tank reduces based on the usage of each block The maintenance person fills the tank thrice with 50litres each time. Display the current water level in the tank.

Solution:

```
import java.util.Scanner;
class Dblock
{
   int u;
   Dblock(int usage) //parameterized constructor
   {
      u = usage;
   }
}
class Eblock //parameterized constructor
{
   int u;
   Eblock(int usage)
   {
      u=usage;
   }
}
public class Labexercise3
{
   public static void main(String args[])
   {
   Scanner s = new Scanner(System.in);
   int maxlevel;
   int currentlevel=0;
   System.out.println("Enter maximum level of tank");
   maxlevel = s.nextInt();
   int choice;
   currentlevel=maxlevel;
   while(currentlevel>= 150)
   {
   System.out.println("Which block used water 1. D Block 2. E Block");
   choice=s.nextInt();
   switch(choice)
   {
      case 1:
         Dblock obj1 = new Dblock(20);
         currentlevel = currentlevel-obj1.u;
```

```
            break;
        case 2:
            Eblock obj2 = new Eblock(40);
            currentlevel= currentlevel-obj2.u;
            break;
        default:
            System.out.println("Invalid choice");

    }
    System.out.println("Currentlevel water level is "+ currentlevel);
    if(currentlevel<=150)
        System.out.println("Call Facility Manager to fill water tank");
    else
        System.out.println("Don't Call Facility Manager to fill water tank");
    }
}
}
```

Output

```
Enter maximum level of tank
200
Which block used water 1. D Block 2. E Block
2
Currentlevel water level is 160
Don't Call Facility Manager to fill water tank
Which block used water 1. D Block 2. E Block
1
Currentlevel water level is 140
Call Facility Manager to fill water tank
```

9. The company XYZ has decided to give some bonus only for employees which belong to IT department. Hence design an application where you can set same bonus amount and department for some employee objects . Depending upon the employee's salary calculate the final salary with bonus. Display company information and employee information separately. Demonstrate the same using java program.

Solution

```
import java.util.*;
public class Employee {
    double fixedsal;
    double finalsalary;
    int eid;
    static String company_name,dept_name;
    static double bonus;
    static
    {
    company_name = "XYZ";
    dept_name="IT";
    bonus=10.0;
    }
    void calculatesalary()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter your fixed salary");
        fixedsal = s.nextDouble();
        System.out.println("Enter your Id");
        eid= s.nextInt();
        finalsalary = fixedsal + (bonus/100)*fixedsal;
    }
    static void displaycompanydata()
    {
    System.out.println("Company Name : " + company_name);
    System.out.println("Department Name : " + dept_name);
    System.out.println("Bonus : " + bonus + "%");
    }
    void displayemployeedata()
    {
        System.out.println("Id is : "+eid);
        System.out.println("Final Salary with Bonus : " + finalsalary);

    }
    public static void main(String args[]) {
    displaycompanydata();
    Employee e1 = new Employee();
    Employee e2 = new Employee();
    e1.calculatesalary();
    e1.displayemployeedata();
    e2.calculatesalary();
    e2.displayemployeedata();

    }
}
```

Output

```
Company Name : XYZ
Department Name : IT
Bonus : 10.0%
Enter your fixed salary
40000
Enter your Id
111
Id is : 111
Final Salary with Bonus : 44000.0
Enter your fixed salary
60000
Enter your Id
222
Id is : 222
Final Salary with Bonus : 66000.0
```

10. The edubuddy is online learning platform and students can enroll in online courses only when they have access-key for the course. Design a java application which checks if the access-key entered by the student is valid or invalid and display suitable message. Demonstrate the scenario for two student objects.

Solution

```
import java.util.Scanner;
class Students {
    public int id, key;
    private int accesskey = 1310;
    public void checkaccess(int key)
    {
        if(key == accesskey)
        {
            System.out.println("You are provided the access for this course");
            System.out.println("Duration of access : 3 months");
        }
        else
        {
           System.out.println("Access Key is wrong ");
           System.out.println("You are not provided the access for this course");
        }
    }

}
public class Access
    {
```

```
    public static void main(String args[]) {
    Students s1 = new Students();
    Students s2 = new Students();
    Scanner s = new Scanner(System.in);
    System.out.println("Enter your id and key");
    s1.id = s.nextInt();
    s1.key = s.nextInt();
    s1.checkaccess(s1.key);
    System.out.println("Enter your id and key");
    s2.id = s.nextInt();
    s2.key = s.nextInt();
    s2.checkaccess(s2.key);
}
}
```

Output

```
Enter your id and key
122
1310
You are provided the access for this course
Duration of access : 3 months
Enter your id and key
124
7878
Access Key is wrong
You are not provided the access for this course
```

Course Code : CSE2008   Course Title : Programming in java
Lab sheet 2

Programs for Module 2

1. XYZ shopping mall wants to distribute lucky coupon for its customers. Each lucky coupon has some prize amount. The mall executive wants to pick 100 order numbers randomly. The order number lies between 1000 and 9999. Design a java application which helps the executive to determine the winners of lucky coupon. The customer is winner if order number is palindrome and if it belongs to those 100 randomly picked one.

Program

```
import java.util.Random;
public class Array {
    public static void main(String args[]) {
    int min = 1000;
    int max = 9999;
    int ordernum[] =  new int[100];
    Random r = new Random();
    for(int i =0; i<100;i++)
    {
    ordernum[i] = r.nextInt((max - min) + 1) + min;
    }
    for(int i=0; i < ordernum.length; i++)
    {
    int number = ordernum[i];
    int reversedNumber = 0;
    int temp=0;
    while(number > 0)
    {
        temp = number % 10;
        number = number / 10;
        reversedNumber = reversedNumber * 10 + temp;
    }
    if(ordernum[i] == reversedNumber)
    System.out.println("The customer with order number"+" "+ordernum[i]+" "+ "won the lucky
coupon");

    }
    }
}
```

Output

```
The customer with order number 4224 won the lucky coupon
The customer with order number 8888 won the lucky coupon
```

2. Mr Ram has gathered the requirements of sports accessories for Team A and Team B as shown below. The cost of sports accessories are given as below. Design a java application to find the total cost of accessories for Team A and Team B separately

Teams	Balls	Bats	Gloves
Team -A	12	45	15
Team -B	15	38	17

Equipment Name	Cost
Balls	9$
Bats	80$
Gloves	60$

Program

```
import java.util.Scanner;
public class Lab3
{
public static void main(String args[])
{
int itemlist[][] = new int[2][3];
int totalcost[]= new int[2];
int cost[] = {9,80,60};
Scanner s = new Scanner(System.in);
for(int i=0;i<2;i++)
{
System.out.println("Enter the quantity of balls, bats and gloves for Team" + (i+1));
for(int j=0;j<3;j++)
{
itemlist[i][j]= s.nextInt();
}
}
for(int i=0;i<2;i++)
{
for(int j=0;j<3;j++)
{
totalcost[i]=totalcost[i]+ itemlist[i][j]*cost[j];
}
}
System.out.println("The total bill for Team A and Team B is as follows");
System.out.println("For Team A " + totalcost[0]);
System.out.println("For Team B " + totalcost[1]);
}
}
```

Output

```
Enter the quantity of balls, bats and gloves for Team1
2 3 4
Enter the quantity of balls, bats and gloves for Team2
1 2 4
The total bill for Team A and Team B is as follows
For Team A 498
For Team B 4609
```

3. Mr John is working as data entry operator in company XYZ. He wants to gather the information about the new employees joining the organization. Design a java application to read and display the information about n employees.

Program

```
import java.util.Scanner;
class Employee
{
  int id,age;
  String name;
  long salary;
void getData()
{
 Scanner sc = new Scanner(System.in);
 System.out.println("Enter Employee Id : ");
 id = sc.nextInt();
 System.out.println("Enter Employee Name : ");
 name = sc.next();
 System.out.print("Enter Employee Age : ");
 age = sc.nextInt();
 System.out.print("Enter Employee Salary : ");
 salary = sc.nextLong();
}
void putData()
{
System.out.println(id + "\t\t" +name + "\t\t" +age + "\t\t"+salary);
}
}
public class EmployeeObj
{
public static void main(String args[])
{
  int n;
  Scanner s =  new Scanner(System.in);
  System.out.println("Enter the number of employees");
  n=s.nextInt();
  Employee[] Emp = new Employee[n];
  int i;
  for(i=0;i<n;i++)
  Emp[i] = new Employee();  // Allocating memory to each object
  for(i=0;i<n;i++)
  {
   System.out.println("Enter details of Employee " + (i+1));
   Emp[i].getData();
  }
  System.out.println("Details of Employees\n");
  for(i=0;i<n;i++)
  Emp[i].putData();
}
}
```

Output

```
Enter the number of employees
2
Enter details of Employee 1
Enter Employee Id :
111
Enter Employee Name :
ram
Enter Employee Age : 33
Enter Employee Salary : 45000
Enter details of Employee 2
Enter Employee Id :
222
Enter Employee Name :
Sham
Enter Employee Age : 45
Enter Employee Salary : 78800
Details of Employees

111             ram             33              45000
222             Sham            45              78800
```

4. Mr John is office executive in Engineering college. He wants to count the number of admissions on each day branch wise. Design a java application which counts the number of students admitted in CSE, ISE and ECE branch depending upon the roll number allocated to the student. The format of roll number is as follows. Example : 20221CSE001.

RollNumber Format : year-1-branch name-last three digits of roll number

Program

```
import java.util.*;
public class Stringprogram1 {
    public static void main(String args[]) {
    int countcse,countece,countise;
    countcse=0;
    countece=0;
    countise=0;
    Scanner s = new Scanner(System.in);
    String rollnumbers[] = new String[10];

    System.out.println("Enter the roll number of 10 students");
    for(int i = 0; i<10;i++)
    {
        rollnumbers[i] = s.next();
    }
    for(int i = 0; i<10; i++)
    {
        String str = rollnumbers[i].toLowerCase();
        String substringstr = str.substring(5,8);

        if(substringstr.equals("cse"))
        countcse = countcse+1;
        else if(substringstr.equals("ise"))
        countise = countise+1;
        else
        countece = countece+1;

    }
    System.out.println("Total CSE students " + countcse);
    System.out.println("Total ISE students " + countise);
    System.out.println("Total ECE students " + countece);
    }
}
```

Output

```
Enter the roll number of 10 students
20221cse001
20221cse002
20221cse003
20221cse004
20221cse005
20221cse006
20221CSE007
20221ece001
20221ece002
20221ise200
Total CSE students 7
Total ISE students 1
Total ECE students 2
```

5. XYZ bank wants to generate a 16 digit transaction password for the customers during each transaction. The transaction password is constructed based on the username and some random digits. Design a java application which generates a 16 digit transaction password using following rule

Note : Transaction Password = first half of user name + random digits + second half of user name and total length of Transaction Password is 16. The entered username should not contain any blank spaces.

Program:

```
import java.util.*;
public class Stringprogram1 {
    public static void main(String args[]) {
    Scanner s = new Scanner(System.in);
    Random r = new Random();
    StringBuffer tpassword =  new StringBuffer();
    int min=0;
    int max=9;
    String name,firsthalf,secondhalf;
    System.out.println("Enter username : ");
    name = s.next();
    int len = name.length();
    System.out.println("Length of username is " + len);
    firsthalf = name.substring(0,len/2);
    secondhalf = name.substring(len/2,len);
    tpassword.append(firsthalf);
    for(int i = 0;i< 16-len; i++)
    {
    int num = r.nextInt((max - min) + 1) + min;
    tpassword.append(num);
    }
    tpassword.append(secondhalf);

    System.out.println("16 digits transaction password is : " + tpassword);
    }
}
```

Output

```
Enter username :
reshmashet
Length of username is 11
16 digits transaction password is : resha68585mashet
```

6. The Royal Sea International provides Passenger Ships and Container Ships. Based on the requirement of the user display suitable ship and its details. Design a Java application based on inheritance for the same.

Program

```
import java.util.Scanner;
class Vehicle
{
    String vehiclename;
    void display()
    {
        System.out.println("Vehicle Name is :" + vehiclename);
    }
}
class WaterVehicle extends Vehicle
{
    int length_of_waterline;
    int length;
    void display()
    {
        super.display();
        System.out.println("Length of Water line :" + length_of_waterline);
        System.out.println("Length :"+ length);
    }
}
class Passengership extends WaterVehicle
{
    int no_of_decks, no_of_crew,no_of_lifeboats;
    String restaurant_name;
    String hospital_name;

    Passengership( int decks, int crew, int lifeboat, String r)
    {
        no_of_decks = decks;
        no_of_crew = crew;
        no_of_lifeboats = lifeboat;
        restaurant_name = r;
    }
    void display()
    {
        super.display();
        System.out.println("No of decks :" + no_of_decks);
        System.out.println("No of crew members :" + no_of_crew);
        System.out.println("No of Life Boats :" + no_of_lifeboats);
        System.out.println("Restaurant Name :" + restaurant_name);
    }
}
class Containership extends WaterVehicle
{
    int no_of_holds;
    String container_material;
    String container_type;
    Containership ( int holds, String m, String t)
    {
        no_of_holds= holds;
        container_material = m;
        container_type = t;
    }
    void display()
    {
        super.display();
        System.out.println("No of Holds :" + no_of_holds);
        System.out.println("Container Material :" + container_material);
        System.out.println("Container Type :" + container_type);
    }
}
public class InheritanceDemo
{
    public static void main(String args[])
    {
        Scanner s =  new Scanner(System.in);
        System.out.println("Enter your purpose 1. Travel on Ship 2. Transportation of Goods");
        int ch  = s.nextInt();
        if(ch==1)
        {
            Passengership ship = new Passengership( 10, 100,1000,"Royal Food");
            ship.vehiclename = "Oasis of Seas";
            ship.length_of_waterline = 1000;
            ship.length = 1200;
            ship.display();
        }
        else
        {
            Containership ship = new Containership(6,"Steel","Refrigerated");
            ship.vehiclename = "MSC Oscar";
            ship.length_of_waterline = 2000;
            ship.length = 2400;
            ship.display();
        }
    }
}
```

Output

```
Enter your purpose 1. Travel on Ship 2. Transportation of Goods
1
Vehicle Name is :Oasis of Seas
Length of Water line :1000
Length :1200
No of decks :10
No of crew members :100
```

No of Life Boats :1000
Restaurant Name :Royal Food

7. The shopping website named Shopify wants to offer supercoins for the customer based on thier purchases. The elite customer earns 2 supercoins for each 100 Rs spent and icon customer earns 4 supercoins for each 100 Rs spent. Design a java application which calculates the total super coins collected yearly by the customer on the basis of amount spent in each month.

Program

```
import java.util.Scanner;
class EliteCustomer
{
int no_of_supercoins;
double[] amount_spent = new double[12];
double total_amount=0;

void calculatetotalamount()
{
   Scanner s = new Scanner(System.in);
   int i;
   for(i=0;i<12;i++)
   {
      System.out.println("Enter the amount spent on orders in Month " + (i+1));
      amount_spent[i] = s.nextDouble();
   }
   for(i=0;i<12;i++)
   {
      total_amount = total_amount+ amount_spent[i];
   }
   System.out.println("Total amount spent in a year is " + total_amount);
}
void calculatesupercoins()
{
   System.out.println("You are eligible for 2 supercoins per 100 Rs spent");
   no_of_supercoins = (int)(total_amount/100)*2;
   System.out.println("The number of supercoins earned is " + no_of_supercoins);
}
}
class IconCustomer extends EliteCustomer
{
void calculatesupercoins() //overridden method
{
   System.out.println("You are eligible for 4 supercoins per 100 Rs spent");
   no_of_supercoins = (int)(total_amount/100)*4;
   System.out.println("The number of supercoins earned is " + no_of_supercoins);
}
}
public class SuperCoins
{
    public static void main(String args[])
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the customer type 1. Elite or 2. Icon");
        int ch = s.nextInt();
        if(ch==1)
        {
            EliteCustomer customer = new EliteCustomer();
            customer.calculatetotalamount();
            customer.calculatesupercoins();
        }
        else
        {
            IconCustomer customer = new IconCustomer();
            customer.calculatetotalamount();
            customer.calculatesupercoins();
        }
    }
}
```

Output

```
Enter the customer type 1. Elite or 2. Icon
2
Enter the amount spent on orders in Month 1
1000
Enter the amount spent on orders in Month 2
1000
Enter the amount spent on orders in Month 3
1000
Enter the amount spent on orders in Month 4
1000
Enter the amount spent on orders in Month 5
0
Enter the amount spent on orders in Month 6
0
Enter the amount spent on orders in Month 7
0
Enter the amount spent on orders in Month 8
2000
Enter the amount spent on orders in Month 9
4000
Enter the amount spent on orders in Month 10
0
Enter the amount spent on orders in Month 11
0
```

```
Enter the amount spent on orders in Month 12
0
Total amount spent in a year is 10000.0
You are eligible for 4 supercoins per 100 Rs spent
The number of supercoins earned is 400
```

Package

1. Create a package named "Presidency". Under this create two packages named "Employee" and "Student". Under the employee package create a class called "EmployeeDetails" having required member fields and methods. Under the Student package create a class called "StudentDetails" having required member fields and methods. Demonstrate the above by creating objects of StudentDetails and EmployeeDetails inside another class which resides in another package.Hint: You can assume the relevant fields and methods to be written inside the EmployeeDetails and StudentDetails class.

//Presidency.java

```
package presidency;

import presidency.employee.EmployeeDetails;
import presidency.student.StudentDetails;

public class Presidency {

    public static void main(String[] args) {
        EmployeeDetails empDetails[] = new EmployeeDetails[5];
        StudentDetails stuDetails[] = new StudentDetails[5];
        for(int i =0; i<5;i++){
            empDetails[i] = new EmployeeDetails();
            empDetails[i].setEmployeeDetails();
        }
        for(int i=0;i<5;i++){
            stuDetails[i] = new StudentDetails();
            stuDetails[i].setStudentDetails();
        }
        for(int i=0;i<5;i++)
            empDetails[i].getEmployeeDetails();

        for(int i=0;i<5;i++)
            stuDetails[i].getStudentDetails();

    }

}
```

//EmployeeDetails.java

```
package presidency.employee;

import java.util.Scanner;

public class EmployeeDetails {
    String empName;
    int empId;
    long empPhoneNumber;
    Scanner scan = new Scanner(System.in);
    public void setEmployeeDetails(){
        System.out.println("Enter the name of the Employee");
        empName = scan.nextLine();
        System.out.println("Enter the ID of the Employee");
        empId = scan.nextInt();
        System.out.println("Enter the phone number of the Employee");
        empPhoneNumber  = scan.nextLong();
    }

    public void getEmployeeDetails(){
        System.out.println("Emp Id: "+empId);
        System.out.println("Emp Name: "+ empName);
        System.out.println("Emp Phone Number: "+ empPhoneNumber);
    }
}
```

//StudentDetails.java

```
package presidency.student;

import java.util.Scanner;

public class StudentDetails {
    String studentId;
    String studentName;
    int semester;
    Scanner scan = new Scanner(System.in);
    public void setStudentDetails(){
        System.out.println("Enter the studnet's Id");
        studentId = scan.nextLine();
        System.out.println("Enter the name of the Student");
        studentName = scan.nextLine();
        System.out.println("Enter the semester the student is in");
        semester = scan.nextInt();
    }
    public void getStudentDetails(){
        System.out.println("Student Id:"+ studentId);
        System.out.println("Student's Name: "+ studentName);
        System.out.println("Semester :"+ semester);
    }
}
```

Packages & Interface

2. Tom is participating in a coding contest, he will win the contest if he performs the following task: Design and develop an application for Banking in Java using the concept of Interface. Follow the following instructions for developing the application.

a. Create an interface IAccount which contains the functions such as balanceEnquiry(int accountNumber), depositAmount(int AccountNumber) and withdrawAmount(int AccountNumber) in package pkgintf.

b. Create a class Account which implements IAccount in package pkgbase.

c. Create a class SavingsAccount which inherits Account class in package pkgder

d. Create a class CurrentAccount which inherits Account class in package pkgder

e. Include a createAccount() method that generates account number in sequence and another method showAccountDetails() to display the account information.

f. Create a Banking in package pkgtest to demonstrate the application using a menu driven program which gives the choices to operate. Create array for both types of accounts and manage the same.

//IAccount.java

```
package pkgintf;

public interface IAccount {
    public void depositAmount(int accountNumber);
    public void balanceEnquiry(int accountNumber);
    public void withdrawAmount(int accountNumber);
}


//Banking.java   //Main

package banking;

import java.util.Scanner;
import pkgbase.Account;
import pkgder.CurrentAccount;
import pkgder.SavingsAccount;

public class Banking {
    public static int saCount=0;
    public static int caCount=0;

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        SavingsAccount savingsAccount[] = new SavingsAccount[10];
        CurrentAccount currentAccount[] = new CurrentAccount[10];
        int choice;
        int ch=0;
        boolean contBanking = true;
    while(contBanking){
    do{
       System.out.println("Create Account 1. Saving Account 2. Current Account 3.
More Options");
      choice = scan.nextInt();
      switch(choice){
        case 1:  savingsAccount[saCount]= new SavingsAccount();
             savingsAccount[saCount].createAccount();
             saCount++;
             break;
        case 2: currentAccount[caCount] = new CurrentAccount();
            currentAccount[caCount].createAccount();
             caCount++;
             break;
      }


    }while(choice<=2);

    do{
      int account=0;
          System.out.println("1. Savings Acc Deposit 2. Savings Acc Withdraw 3.
Savings Acc BalanceEnquiry 4.Back");
       System.out.println("Enter your choice of Banking");
       ch = scan.nextInt();
       if(ch<=3){
         System.out.println("Enter the Account Number");
         account = scan.nextInt();
       }
       switch(ch){
         case 1: for(int i=0;i<saCount;i++){
               savingsAccount[i].depositAmount(account);
             }
             break;
         case 2: for(int i=0;i<saCount;i++){
               savingsAccount[i].withdrawAmount(account);
             }
             break;
         case 3: for(int i=0;i<saCount;i++){
               savingsAccount[i].balanceEnquiry(account);
             }
             break;
       }
    }while(ch<=3);


    do{
      int account=0;
          System.out.println("1. Current Acc Deposit 2. Current Acc Withdraw 3.
Current Acc BalanceEnquiry 4.Back");
      System.out.println("Enter your choice of Banking");
       ch = scan.nextInt();
```

```
            if(ch<=3){
               System.out.println("Enter the Account Number");
               account = scan.nextInt();
            }

            switch(ch){
                case 1: System.out.println("Number of CA "+caCount);
                         for(int i=0;i<caCount;i++){
                            currentAccount[i].depositAmount(account);
                        }
                        break;
                case 2: for(int i=0;i<caCount;i++){
                            currentAccount[i].withdrawAmount(account);
                        }
                        break;
                case 3: for(int i=0;i<caCount;i++){
                            currentAccount[i].balanceEnquiry(account);
                        }
                        break;
            }
        }while(ch<=3);

              System.out.println("Would you like to continue ?? Input 1 to continue... Any
other to exit");
            int num = scan.nextInt();
            if(num!=1)
                contBanking = false;
        }
    }

}
```

//Account.java

```
package pkgbase;

import java.util.Scanner;
import pkgintf.IAccount;

public class Account implements IAccount {
    protected int accountNumber;
    protected String accountHolderName;
    protected long accountBalance;
    protected double rateOfInterest;
    protected boolean overdraft;

    Scanner scan = new Scanner(System.in);
    public void showAccountDetails(){
        System.out.println("Account Number :" + accountNumber);
        System.out.println("Account Holder Name :"+ accountHolderName);
        System.out.println("Account Balance :" + accountBalance);
        System.out.println("Rate of Interest for your Acc : " +rateOfInterest);
        if(overdraft)
        System.out.println("Overdraft Allowed");
        else{
            System.out.println("Overdraft not allowed");
        }
    }
    @Override
    public void depositAmount(int accountNumber) {
        if(accountNumber==this.accountNumber){
            System.out.println("Enter amount to be deposited");
            int amount = scan.nextInt();
            accountBalance +=amount;
        }
    }

    @Override
    public void balanceEnquiry(int accountNumber) {
        if(accountNumber == this.accountNumber)
            System.out.println("Balnce in Account"+ accountBalance);
    }

    @Override
    public void withdrawAmount(int accountNumber) {
        if(accountNumber == this.accountNumber){
            System.out.println("Enter the amount to withdraw");
            int amount = scan.nextInt();
            if(this.accountNumber<50000 && amount<=accountBalance){
                accountBalance = accountBalance-amount;
                System.out.println("Balance in Account" + accountBalance);
            }
            else{
                                System.out.println("Not enough Balance, your balance is
"+accountBalance);
            }
            if(this.accountNumber>=50000){
                accountBalance = accountBalance-amount;
                System.out.println("Balance in Account" + accountBalance);
            }

    }
  }

}
```

//CurrentAccount.java

```
package pkgder;

import java.util.Scanner;
import pkgbase.Account;


public class CurrentAccount extends Account{
    String businessName;
    static int accNumGen = 50000;
    Scanner scan = new Scanner(System.in);
    public CurrentAccount() {
        rateOfInterest = 0;
        overdraft = true;
    }
    public void createAccount(){
        Scanner scan = new Scanner(System.in);
        accountNumber = accNumGen++;
        System.out.println("Enter the Account Holder Name");
        accountHolderName = scan.nextLine();
        showAccountDetails();
    }

}
```

//SavingsAccount.java

```
package pkgder;

import java.util.Scanner;
import pkgbase.Account;

public class SavingsAccount extends Account{
    static int accnumGen = 10000;
    public SavingsAccount() {
        rateOfInterest = 5.5;
        overdraft = false;
    }

    public void createAccount(){
        Scanner scan = new Scanner(System.in);
        accountNumber = accnumGen++;
        System.out.println("Enter the Account Holder Name");
        accountHolderName = scan.nextLine();
        showAccountDetails();
    }

}
```

Exception Handling

3. Shyam's teacher asked him to submit an assignment on division of integers , he wrote a java program to complete the assignment, his program abruptly ended at a condition. Identify the condition where Shyama program abruptly ended and handle the condition using suitable measures.

```
package shyamassignment;

import java.util.Scanner;


public class ShyamAssignment {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        int choice =1;
        while(choice ==1){
            System.out.println("Please input any number & a divisor to divide ");
            double number = scan.nextInt();
            double div = scan.nextInt();

            try{
                double result = number / div;
                System.out.println("result is: "+ result);
            }catch(ArithmeticException e){
                  System.out.println("Kindly input Another Divisor,You are trying to divide
by Zero");
            }

            System.out.println("Try Again? Input 1 for yes..");
            choice = scan.nextInt();

        }
    }

}
```

User Defined Exception

4. A **stack** is a basic data structure that can be logically thought of as a linear structure. Two operations that can be performed on a Stack are: Push operation which inserts an element into the stack. Pop operation which removes the last element that was added into the stack. It follows Last In First Out(LIFO) Order. Write a menu driven program in Java to perform the operations on an IntegerStack. Create custom exceptions to deal with the following situations

1.“EmptyStackException”, while trying to access from an empty stack.
2.“FullStackException” while trying to insert to a full stack.

```
package stack;

import java.util.Scanner;

class FullStackException extends Exception{
   int size;

   public FullStackException() {
   }

   public FullStackException(int size) {
      this.size = size+1;
   }
   public String toString(){
        return "FullStackException: The number of elements in the stack is : "+size+"
Can't insert";
   }

}
class EmptyStackException extends Exception{
   int size;
   public EmptyStackException() {
   }

   public EmptyStackException(int size) {
      this.size = size;
   }
   public String toString(){
         return "EmptyStackException: The number of elements in the stack is : "+
(size+1);
   }

}

public class Stack {
```

```
int top=-1;
final int maxElements = 3;
int numberStack[] = new int[maxElements];

public void push(int element){
   try{
      if(top==maxElements-1){
         throw new FullStackException(top);
      }
      else{
         numberStack[++top] = element;
      }
   }catch(FullStackException fse){
      System.out.println(fse);
   }

}
public void  pop(){
   int element=0;
   try{
   if(top==-1){
      throw new EmptyStackException(top);
   }
   else{
      System.out.println("Popped item is "+ numberStack[top--]);
   }

   }catch(EmptyStackException ese){
      System.out.println(ese);
   }

}

public void display(){
   try{
      if(top==-1){
         throw new EmptyStackException(top);
      }
      for(int i = top; i>=0 ; i--){
         System.out.println(numberStack[i]);
      }
   }catch(EmptyStackException ese){
      System.out.println(ese);
   }

}
public static void main(String[] args) {
   int choice =0;
   Scanner scan = new Scanner(System.in);
  Stack stack = new Stack();
  do{
          System.out.println("1. Push 2.Pop 3.Display 4. Exit");
          System.out.println("Enter your choice");
          choice = scan.nextInt();
          switch(choice){

             case 1:  System.out.println("Enter the element");
                    int element = scan.nextInt();
                    stack.push(element);
                    break;

             case 2: stack.pop();
                    break;

             case 3: System.out.println("Items in the stack are");
                    stack.display();
                    break;
          }

        }while(choice<=3);
      }

   }
```

Labsheet 4

Module 4

Question 1: With the help of Multithreading concept, create two separate threads, one thread titled "primeThread" by extending the Thread class, and the other titled "fiboThread" by implementing Runnable interface. "primeThread" will be responsible to print all the prime numbers from1 to 100 in a regular interval of 0.25 seconds."fiboThread" will be responsible to print fibonacci series of 20 numbers,in a regular interval of 0.5 seconds.
On executing this application, get the following info also.
a) Get the id,name and priority of the main thread
b) Change the name and priority of the main thread and print the same.
c) Print the thread group info of both the child threads
c) Use isAlive method to check the status of the childThread.

Program

```
class Prime extends Thread{
boolean isPrime = true;
@Override
public void run() {
for(int num=2;num<100;num++){
for(int div=2;div<=num/2;div++){
if(num%div==0){
isPrime=false;
}
}
if(isPrime){
try{
System.out.println("Prime :"+ num);
Thread.sleep(250);
}catch(InterruptedException ie){
}
}
isPrime=true;
}
}
}

class Fibonacci implements Runnable{
int n1 = 0, n2=1;
final int MAX = 20;
@Override
public void run() {
System.out.println("Fib: "+n1);
System.out.println("Fib: "+n2);
for(int i=2;i<MAX;i++){
int n3 = n1+n2;
try{
```

```
System.out.println("Fib: "+ n3);
Thread.sleep(500);
}catch(InterruptedException ie){
}
n1=n2;
n2=n3;
}
}
}
public class PrimeandFibThread {
/**
* @param args the command line arguments
*/
public static void main(String[] args) {
//get the name and Id of the main thread
Thread mainThread = Thread.currentThread();
System.out.println("Id of Main Thread: "+ mainThread.getId());
System.out.println("Name of main Thread: "+ mainThread.getName());
System.out.println("Default Priority of Main Thread: "+ mainThread.getPriority());
//Change the name and priority of the main thread and print the same.
mainThread.setName("ThreadMainClass");
mainThread.setPriority(Thread.MIN_PRIORITY);
System.out.println("Name of main Thread after changing: "+mainThread.getName());
System.out.println("Priority of main thread after changing:"+mainThread.getPriority());
        //creating a child thread "primeThread"
        Prime prime= new Prime(); //prime is thread object
        prime.setName("primeThread");
        //creating a child thread "fiboThread"
        Fibonacci fibonacci = new Fibonacci();//fibonacci is object of class Fibonacci
        Thread fib = new Thread(fibonacci,"fiboThread");
        prime.start();
        fib.start();
        //Thread group of the child-Thread
        System.out.println("child-Thread1: "+ prime);
        System.out.println("child-Thread2: "+fib);
        //checking with isAlive() method
        System.out.println("primeThread is Alive??: "+ prime.isAlive());
        }
        }
```

Output

```
Id of Main Thread: 1
Name of main Thread: main
Default Priority of Main Thread: 5
Name of main Thread after changing: ThreadMainClass
Priority of main thread after changing:        1
Fib: 0
Prime :2
```

```
Fib: 1
Fib: 1
child-Thread1: Thread[primeThread,1,main]
child-Thread2: Thread[fiboThread,1,main]
primeThread is Alive??: true
Prime :3
Fib: 2
Prime :5
Prime :7
Fib: 3
Prime :11
Prime :13
Fib: 5
Prime :17
Prime :19
Fib: 8
Prime :23
Prime :29
Fib: 13
Prime :31
Prime :37
Fib: 21
Prime :41
Prime :43
Fib: 34
Prime :47
Prime :53
Fib: 55
Prime :59
Prime :61
Fib: 89
Prime :67
Prime :71
Fib: 144
Prime :73
Prime :79
Fib: 233
Prime :83
Prime :89
Fib: 377
Prime :97
Fib: 610
Fib: 987
Fib: 1597
Fib: 2584
Fib: 4181
```

Question 2 : Presidency University is organising an event. While each participant arrives, get the name of the participant and assign participant IDs to each participant starting from 101, and should be following with sequence of ID for the further participants. Presidency has created a String array that has the following info. {“Hi”, “name of the participant”, “ID of the participant”, “Welcome Message”}. The registration team register the Participant details. You can create a class “Participant” with participantId, participantName as members. Create a setter method to set the participant details. Once the participants are registered, the welcome message is printed by separate Threads on Participant instance. Write a java program to handle
this scenario.

Program

```
import java.util.Scanner;
class Presidency {
static synchronized void display(int id, String name)
{
String welcomeMessgae[] ={"Hi",name, "yourId "+ id+"", "welcome to Presidency University"};
for(int i=0; i<welcomeMessgae.length; i++)
{
try
{
Thread.sleep(1000);
System.out.println(welcomeMessgae[i]);
} catch(Exception e) { }
}
}
}
class Participant extends Thread
{
static int temp = 100;
int participantId;
String participantName;
Scanner scan = new Scanner(System.in);
public void setDetails()
{
System.out.println("Enter the name");
participantName = scan.nextLine();
participantId = ++temp;
}
public void run()
{
Presidency.display(participantId, participantName);
}
}
public class TestThread
{
public static void main(String[] args) {
Participant p1 = new Participant();
Participant p2 = new Participant();
Participant p3 = new Participant();
p1.setDetails();
p2.setDetails();
p3.setDetails();
p1.start();
p2.start();
p3.start();
}
}
```

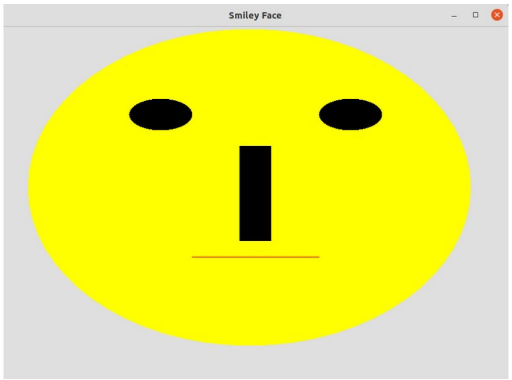
Output

```
Enter the name
tom
Enter the name
tim
Enter the name
mike
Hi
tom
yourId 101
welcome to Presidency University
Hi
mike
yourId 103
welcome to Presidency University
Hi
tim
yourId 102
welcome to Presidency University
```

Labsheet 5

Module 5

Question 1: Mr. Sham wants to create smiley pattern face. Write a java program using AWT graphics to create a smiley face design.



Program

```
import java.awt.*;
import java.awt.event.*;
public class SmileyCreate extends Frame{
        SmileyCreate()
        {
                setSize(800, 600);
                setTitle("Smiley Face");
                setVisible(true);
                addWindowListener(new WindowAdapter()
                {
                        public void windowClosing(WindowEvent we)
                        {
                                System.exit(0);
                        }
                }
                );
        }
        public void paint(Graphics g)
        {
                        g.setColor(Color.YELLOW);
                        g.fillOval(40,40,700,500);
                        g.setColor(Color.BLACK);
                        g.fillOval(200,150,100,50);
                        g.fillOval(500,150,100,50);
                        g.setColor(Color.BLACK);
                        g.fillRect(375, 225, 50,150);
```

```
                    g.setColor(Color.RED);
                    g.drawLine(300, 400, 500, 400);
        }
        public static void main(String[] args)
        {
                SmileyCreate s = new SmileyCreate();
        }
}
```

Question 2:

Design an interactive form using Java AWT which shows the count of number of words and characters in the entered string.

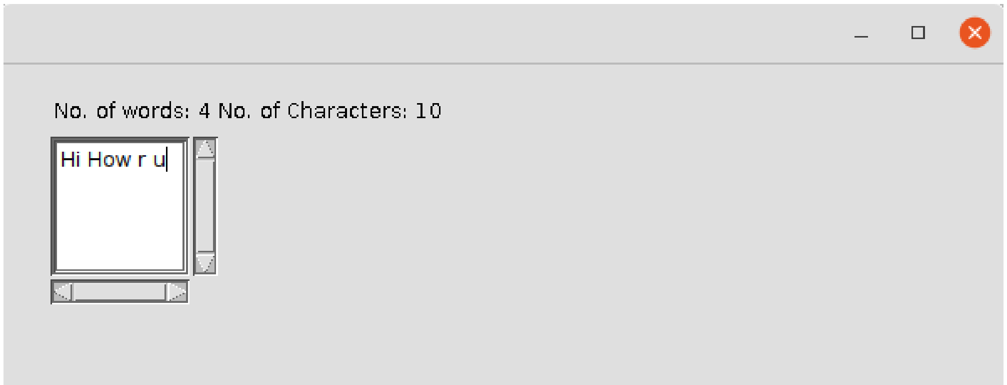
Program

```
import java.awt.Frame;
import java.awt.Label;
import java.awt.TextArea;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
public class JavaKeyListenerExample extends Frame implements KeyListener{
//label
Label lbl;
//text area
TextArea a;

JavaKeyListenerExample()
{
//create a label
lbl=new Label();
//set the bounds for label
lbl.setBounds(30,50,300,30);
//create a text area
a=new TextArea();
//set the bounds for text area
a.setBounds(30 , 80 , 100 , 100);
a.addKeyListener(this); // registering the textarea component for key event
//add label
add(lbl);
//add text area
add(a);
//set the size
setSize(600,600);
//set layout
setLayout(null);
//set the visibility as true
setVisible(true);
}

public void keyPressed( KeyEvent ev )
{
        System.out.println("The key is pressed");
}
//on releasing the key
public void keyReleased( KeyEvent ev )
{
//get the text typed in text area
String t=a.getText();
//split the text
String w[]=t.split("\\s");
//set the text as label
lbl.setText("No. of words: "+ w.length +" No. of Characters: "+ t.length() );
}
public void keyTyped( KeyEvent ev ) {
}
//main method
public static void main(String[] args) {
JavaKeyListenerExample obj = new JavaKeyListenerExample();
}  }
```

Output

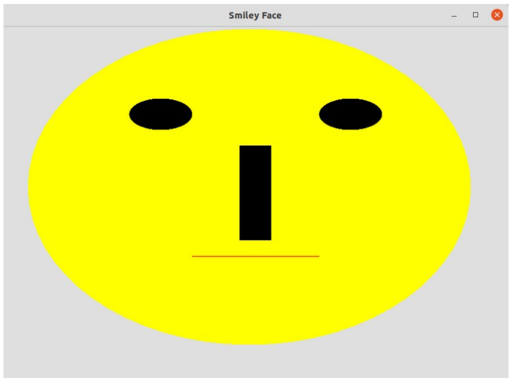


```
No. of words: 4 No. of Characters: 10
Hi How r u
```

Labsheet 5

Module 5

Question 1: Mr. Sham wants to create smiley pattern face. Write a java program using AWT graphics to create a smiley face design.



Program

```
import java.awt.*;
import java.awt.event.*;
public class SmileyCreate extends Frame{
        SmileyCreate()
        {
                setSize(800, 600);
                setTitle("Smiley Face");
                setVisible(true);
                addWindowListener(new WindowAdapter()
                {
                        public void windowClosing(WindowEvent we)
                        {
                                System.exit(0);
                        }
                }
                );
        }
        public void paint(Graphics g)
        {
                        g.setColor(Color.YELLOW);
                        g.fillOval(40,40,700,500);
                        g.setColor(Color.BLACK);
                        g.fillOval(200,150,100,50);
                        g.fillOval(500,150,100,50);
                        g.setColor(Color.BLACK);
                        g.fillRect(375, 225, 50,150);
```

```
                    g.setColor(Color.RED);
                    g.drawLine(300, 400, 500, 400);
        }
        public static void main(String[] args)
        {
                SmileyCreate s = new SmileyCreate();
        }
}
```

Question 2:

Design an interactive form using Java AWT which shows the count of number of words and characters in the entered string.

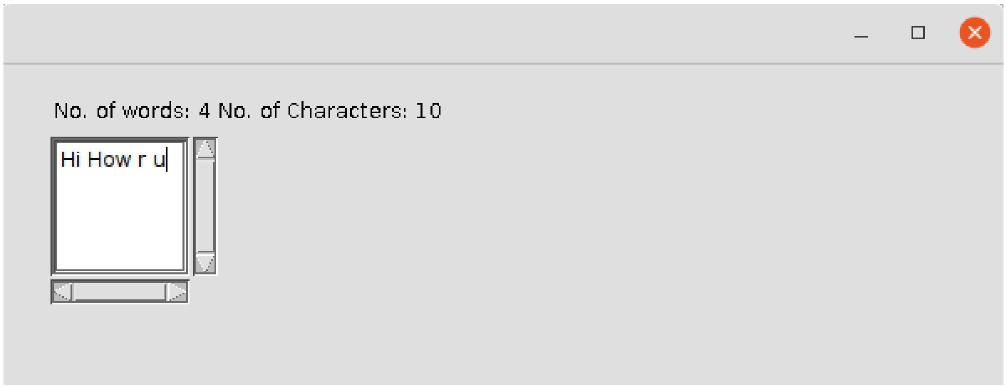
Program

```
import java.awt.Frame;
import java.awt.Label;
import java.awt.TextArea;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
public class JavaKeyListenerExample extends Frame implements KeyListener{
//label
Label lbl;
//text area
TextArea a;

JavaKeyListenerExample()
{
//create a label
lbl=new Label();
//set the bounds for label
lbl.setBounds(30,50,300,30);
//create a text area
a=new TextArea();
//set the bounds for text area
a.setBounds(30 , 80 , 100 , 100);
a.addKeyListener(this); // registering the textarea component for key event
//add label
add(lbl);
//add text area
add(a);
//set the size
setSize(600,600);
//set layout
setLayout(null);
//set the visibility as true
setVisible(true);
}

public void keyPressed( KeyEvent ev )
{
        System.out.println("The key is pressed");
}
//on releasing the key
public void keyReleased( KeyEvent ev )
{
//get the text typed in text area
String t=a.getText();
//split the text
String w[]=t.split("\\s");
//set the text as label
lbl.setText("No. of words: "+ w.length +" No. of Characters: "+ t.length() );
}
public void keyTyped( KeyEvent ev ) {
}
//main method
public static void main(String[] args) {
JavaKeyListenerExample obj = new JavaKeyListenerExample();
}  }
```

Output



```
No. of words: 4 No. of Characters: 10
```

```
Hi How r u
```

Question 3: Mr Ram is developing a music player where each user has his own song playlist. Design a java application that enables the user to add or delete any song from the playlist. The user can add or delete the song at first or last location or anywhere.

Program

```
import java.util.*;
public class PlayList {
```

```
public static void main(String args[])
{
    boolean contplay=true;
    LinkedList<String> playlist = new LinkedList<String>();
    int choice;
    while(contplay)
    {
    System.out.println("Enter your choice 1. Add Song 2. Delete song 3. Display 4. Exit ");
    Scanner s =  new Scanner(System.in);
    choice=s.nextInt();

    if(choice == 1){
        System.out.println("Enter song name");
        String sname = s.next();
        int addchoice;
        System.out.println("Select option 1. Add First 2. Add Last 3. Add anywhere");
        addchoice=s.nextInt();
        switch(addchoice)
        {
            case 1:
                playlist.addFirst(sname);
                System.out.println("Song added at first");
                break;
            case 2:
                playlist.addLast(sname);
                System.out.println("Song added at last");
                break;
            case 3:
                playlist.add(sname);
                System.out.println("Song added");
        }

    }
    else if(choice==2)
    {
        int deletechoice;
        System.out.println("Select option 1. Delete First 2. Delete Last 3. Delete Specific song");
        deletechoice=s.nextInt();
        switch(deletechoice)
        {
            case 1:
                playlist.removeFirst();
                System.out.println("First song removed");
                break;
            case 2:
                playlist.removeLast();
                System.out.println("Last song removed");
                break;
            case 3:
```

```
          System.out.println("Enter song name");
          String sname = s.next();
          playlist.remove(sname);
          System.out.println("Song Removed");
          break;
      }
    }
    else if(choice ==3)
    {
      System.out.println("Song Playlist : " + playlist );
    }
    else
    {
      System.out.println("Exit Done");
      contplay= false;
    }
    }
  }
}
```

Output

```
Enter your choice 1. Add Song 2. Delete song 3. Display 4. Exit
1
Enter song name
hey
Select option 1. Add First 2. Add Last 3. Add anywhere
1
Song added at first
Enter your choice 1. Add Song 2. Delete song 3. Display 4. Exit
1
Enter song name
rain
Select option 1. Add First 2. Add Last 3. Add anywhere
2
Song added at last
Enter your choice 1. Add Song 2. Delete song 3. Display 4. Exit
1
Enter song name
colors
Select option 1. Add First 2. Add Last 3. Add anywhere
3
Song added
Enter your choice 1. Add Song 2. Delete song 3. Display 4. Exit
3
Song Playlist : [hey, rain, colors]
Enter your choice 1. Add Song 2. Delete song 3. Display 4. Exit
2
Select option 1. Delete First 2. Delete Last 3. Delete Specific song
```

```
1
First song removed
Enter your choice 1. Add Song 2. Delete song 3. Display 4. Exit
3
Song Playlist : [rain, colors]
Enter your choice 1. Add Song 2. Delete song 3. Display 4. Exit
2
Select option 1. Delete First 2. Delete Last 3. Delete Specific song
2
Last song removed
Enter your choice 1. Add Song 2. Delete song 3. Display 4. Exit
3
Song Playlist : [rain]
Enter your choice 1. Add Song 2. Delete song 3. Display 4. Exit
4
Exit Done
```