S.Y B.Sc Information Technology

Semester IV

Core JAVA

Program :-01

Aim :-

```
importjava.util.Scanner;

public class prog1 {

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

System.out.print("Input a number: ");
int num1 = in.nextInt();

for (int i=0; i< 10; i++)
{
System.out.println(num1 + " x " + (i+1) + " = " + (num1 * (i+1)));
  }
 }
}
```

Output :-

Program :-02

Aim :-

```
public class prog3 {

private static final double radius = 7.5;

public static void main(String[] args) {

double perimeter = 2 * Math.PI * radius;
double area = Math.PI * radius * radius;

System.out.println("Perimeter is = " + perimeter);
System.out.println("Area is = " + area);
    }
}
```

Output :-

Program :-03

Aim :-

```
class Star
```

```
{
public static void main(String[] args)
{
inti,j;
for(i=1; i<=6; i++)
{
for(j=1; j<i; j++)
{
System.out.print("*");
}
System.out.println();
}
}
}
```

Output :-

Program :-04

Aim :-

```
public class PrintPattern
{
public static void main(String args[])
   {
int space=0;
for(int i=5; i>=1; i--)
      {
```

```
for(int j=1; j<=space;j++)
System.out.print(" ");
for(int j=1; j<=i; j++)
System.out.print("*");
System.out.println();
space++;
        }
    }
}
```

Output :-

Program :-05

Aim :-

```
importjava.util.*;
class binary
{
          public static void main (String[] args)
          {
          Scanner sc = new Scanner(System.in);
                    int[] arr=new int[8];
                    int i=0,m=0,n=0,sum=0;
          System.out.println ("enter 1st binary number");
                    int n1 = sc.nextInt();
          System.out.println ("enter 2nd binary number");
```

```
        int n2 = sc.nextInt();
        for(i=arr.length-1;i>=0;i--)
        {
            m=n1%10;
            n=n2%10;
            n1=n1/10;
            n2=n2/10;
            sum=m+n+arr[i];

            if(sum==1)
            {
                arr[i]=sum;
            }

            else if(sum==2)
            {
                arr[i]=0;
                arr[i-1]=1;
            }

            else if(sum==3)
            {
                arr[i]=1;
                arr[i-1]=1;
            }
            else
            {

                arr[i]=0;
            }
        }
        for(i=0;i<arr.length;i++)
        {
            if(i==4)
            System.out.print(" ");

            System.out.print(arr[i]);
        }
    }
```

```
}
```

Output :-

Program :-06

Aim :-

```
importjava.util.*;
classDtoB
{
public static void main(String arg[])
{
   Scanner sc=new Scanner(System.in);
System.out.println("Enter a decimal number");
int n=sc.nextInt();
System.out.print("Binary number is : ");
binary(n);
}
static void  binary(intnum)
{
int i = 0;
int  bin[]=new int[100];
bin[0]=0;
while(num>0)
    {
```

```
bin[i++] = num%2;
num = num/2;
    }
for(int j =i-1;j >= 0;j--)
    {
System.out.print(bin[j]);
    }
}
}
```

Output :-

Program :-07

Aim :-

```
importjava.util.*;
classReverseString
{
public static void main(String args[])
   {
      String original, reverse = "";
      Scanner in = new Scanner(System.in);

System.out.println("Enter a string to reverse");
original = in.nextLine();
```

```
int length = original.length();

for ( int i = length - 1 ; i >= 0 ; i-- )
reverse = reverse + original.charAt(i);

System.out.println("Reverse of entered string is: "+reverse);
  }
}
```

Output :-

Program :-08

Aim :-

```
public class LargestSmallest
{
	public static void main(String[] args)
	{
		int a[] = new int[] { 23, 34, 13, 64, 72, 90, 10, 15, 9, 27 };

		int min = a[0]; //  assume first elements as smallest number
		int max = a[0]; //  assume first elements as largest number

		for (int i = 1; i <a.length; i++)  // iterate for loop from arrays 1st
index (second element)
		{
```

```
                        if (a[i] > max)
                        {
                                max = a[i];
                        }
                        if (a[i] < min)
                        {
                                min = a[i];
                        }
                }

                System.out.println("Largest Number in a given array is : " + max);
                System.out.println("Smallest Number in a given array is : " + min);
        }
}
```

Output :-

Program :-09

Aim :-

____

____

____

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

```
            int a[] = new int[] { 23, 34, 13, 64, 72, 90, 10, 15, 9, 27 };

            int min = a[0]; //  assume first elements as smallest number
            int max = a[0]; //  assume first elements as largest number

            for (int i = 1; i <a.length; i++)  // iterate for loop from arrays 1st index (second element)
            {
                if (a[i] > max)
                {
                    max = a[i];
                }
                if (a[i] < min)
                {
                    min = a[i];
                }
            }
            System.out.println("Largest Number in a given array is : " + max);
            System.out.println("Smallest Number in a given array is : " + min);
        }
}
```

Output :-

Program :-10

Aim :-

```
importjava.util.Scanner;
public class Exercise38 {

public static void main(String[] args) {
		String test = "core java";
		count(test);

	}
	public static void count(String x){
		char[] ch = x.toCharArray();
		int letter = 0;
		int space = 0;
		intnum = 0;
		int other = 0;
		for(int i = 0; i <x.length(); i++){
			if(Character.isLetter(ch[i])){
				letter ++ ;
			}
			else if(Character.isDigit(ch[i])){
				num ++ ;
			}
			else if(Character.isSpaceChar(ch[i])){
				space ++ ;
			}
			else{
				other ++;
			}
		}
		System.out.println("The string is : core java");
		System.out.println("letter: " + letter);
		System.out.println("space: " + space);
		System.out.println("number: " + num);
		System.out.println("other: " + other);
			}
}
```

Output :-

Program :-11

Aim :-

____

____

____

```
importjava.util.Scanner;
public class SumOfDigits{ public static void main(String args[])
{
 Scanner sc = new Scanner(System.in);
System.out.println("Please enter a number to calculate sum of digits");
int number = sc.nextInt();
int sum = 0;
int input = number;
while (input != 0)
{ intlastdigit = input % 10;
sum += lastdigit;
input /= 10;
  }
System.out.printf("Sum of digits of number %d is %d", number, sum);
sc.close();
} }
```

Output :-

Program :-12

Aim :-

___

___

___

```
abstract class calc
{
abstractintsqr(int n1);
abstractint cube(int n1);

void show()
{
System.out.println("Hello");
 }
}
class pract4 extends calc
{
intsqr(int n1)
 {
return n1*n1;
 }
int cube(int n1)
{
return n1*n1*n1;
}

public static void main(String args[])
{
pract4 p1=new pract4();
System.out.println(p1.sqr(333));
System.out.println(p1.cube(444));
p1.show();
}
}
```

Output :-

Program :-13

Aim :-

```
class cons
{
cons()
{
System.out.println("Hello");
}
public static void main(String args[])
{
cons c1=new cons();
 c1=null;
System.gc();
}
public void finalize()
{
System.out.println("Destroyed");
}
}
```

Output :-

Program :-14

Aim :-

____

____

____

```
importjava.util.Arrays;
importjava.util.Collections;
public class SortData {

voidasec()
{
        // sample int[] array
Integer[] intArray = {1975, 2003, 1979, 1992, 1983, 1999, 1987};
        // before sorting
System.out.println("Integer[] Array - before sorting : ");
for(Integer iValue : intArray) {
System.out.println(iValue);
        }
        // sorting int[] array in ascending order
Arrays.sort(intArray);
        // after sorting
System.out.println("\nInteger[] Array - after sorting in ascending order : ");
for(Integer iValue : intArray) {
System.out.println(iValue);
        }
}
voiddesc()
{
// sample int[] array
Integer[] intArray = {1975, 2003, 1979, 1992, 1983, 1999, 1987};
        // before sorting
System.out.println("Integer[] Array - before sorting : ");
for(Integer iValue : intArray) {
System.out.println(iValue);
        }
        // sorting int[] array in descending order
Arrays.sort(intArray, Collections.reverseOrder());
```

```
        // after sorting
System.out.println("\nInteger[] Array - after sorting in descending order : ");
for(Integer iValue : intArray) {
System.out.println(iValue);
        }
}
public static void main(String[] args) {
SortDataai=new SortData();
SortData a2=new SortData();
ai.asec();
a2.desc();
}
}
```

Output :-

Program :-15

Aim :-

___

___

___

```
class Human{
public void eat()
   {
System.out.println("Human is eating");
   }
}
class Boy1 extends Human{
```

```
public void eat(){
System.out.println("Boy is eating");
  }
public static void main( String args[]) {
    Boy1 obj = new Boy1();
obj.eat();
  }
}
```

Output :-

Program :-16

Aim :-

```
classSingleInheritance{
        staticint num1=10;
        staticint num2=5;
}
classMainInheritance extends SingleInheritance{
        public static void main(String[] args){
        int num3=2;
        int result=num1+num2+num3;
        System.out.println("Result of child class is "+result);
        }
}
```

Output :-

Program :-17

Aim :-

```
interface Car
{
int  speed=60;
public void distanceTravelled();
}
interface Bus
{
int distance=100;
public void speed();
}
public class Vehicle  implements Car,Bus
{
intdistanceTravelled;
intaverageSpeed;
public void distanceTravelled()
    {
distanceTravelled=speed*distance;
System.out.println("Total Distance Travelled is : "+distanceTravelled);
    }
public void speed()
    {
intaverageSpeed=distanceTravelled/speed;
```

```
System.out.println("Average Speed maintained is : "+averageSpeed);
    }
public static void main(String args[])
    {
        Vehicle v1=new Vehicle();
v1.distanceTravelled();
v1.speed();
    }
}
```

Output :-

Program :-18

Aim :-
___
___
___

```
importjava.util.Scanner;
class matrix
{
public static void main(String args[])
   {
int m, n, p, q, sum = 0, c, d, k;

      Scanner in = new Scanner(System.in);
System.out.println("Enter the number of rows and columns of first matrix");
      m = in.nextInt();
      n = in.nextInt();
```

```
int first[][] = new int[m][n];
System.out.println("Enter the elements of first matrix");
for ( c = 0 ; c < m ; c++ )
for ( d = 0 ; d < n ; d++ )
first[c][d] = in.nextInt();
System.out.println("Enter the number of rows and columns of second matrix");
    p = in.nextInt();
    q = in.nextInt();
if ( n != p )
System.out.println("Matrices with entered orders can't be multiplied with each
other.");
else
    {
int second[][] = new int[p][q];
int multiply[][] = new int[m][q];
System.out.println("Enter the elements of second matrix");
for ( c = 0 ; c < p ; c++ )
for ( d = 0 ; d < q ; d++ )
second[c][d] = in.nextInt();
for ( c = 0 ; c < m ; c++ )
      {
for ( d = 0 ; d < q ; d++ )
        {
for ( k = 0 ; k < p ; k++ )
          {
sum = sum + first[c][k]*second[k][d];
          }
multiply[c][d] = sum;
sum = 0;
        }
      }
System.out.println("Product of entered matrices:-");
for ( c = 0 ; c < m ; c++ )
      {
for ( d = 0 ; d < q ; d++ )
System.out.print(multiply[c][d]+"\t");
System.out.print("\n");
      }
    }
```

```
   }
}
```

Output :-

Program :-19

Aim :-

```
importjava.util.Scanner;
class matrix
{
public static void main(String args[])
   {
int m, n, p, q, sum = 0, c, d, k;
      Scanner in = new Scanner(System.in);
System.out.println("Enter the number of rows and columns of first matrix");
      m = in.nextInt();
      n = in.nextInt();

int first[][] = new int[m][n];
System.out.println("Enter the elements of first matrix");
for ( c = 0 ; c < m ; c++ )
for ( d = 0 ; d < n ; d++ )
first[c][d] = in.nextInt();
System.out.println("Enter the number of rows and columns of second matrix");
```

```
    p = in.nextInt();
    q = in.nextInt();
if ( n != p )
System.out.println("Matrices with entered orders can't be multiplied with each
other.");
else
    {
int second[][] = new int[p][q];
int multiply[][] = new int[m][q];
System.out.println("Enter the elements of second matrix");
for ( c = 0 ; c < p ; c++ )
for ( d = 0 ; d < q ; d++ )
second[c][d] = in.nextInt();
for ( c = 0 ; c < m ; c++ )
      {
for ( d = 0 ; d < q ; d++ )
        {
for ( k = 0 ; k < p ; k++ )
          {
sum = sum + first[c][k]*second[k][d];
          }
multiply[c][d] = sum;
sum = 0;
        }
      }
System.out.println("Product of entered matrices:-");
for ( c = 0 ; c < m ; c++ )
      {
for ( d = 0 ; d < q ; d++ )
System.out.print(multiply[c][d]+"\t");
System.out.print("\n");
      }
    }
  }
}
```

Output :-

Program :-20

Aim :-

```
importjava.util.Scanner;
classAddTwoMatrix
{
public static void main(String args[])
   {
int m, n, c, d;
      Scanner in = new Scanner(System.in);
System.out.println("Enter the number of rows and columns of matrix");
      m = in.nextInt();
n  =in.nextInt();
int first[][] = new int[m][n];
int second[][] = new int[m][n];
int sum[][] = new int[m][n];
System.out.println("Enter the elements of first matrix");
for (  c = 0 ; c < m ; c++ )
for ( d = 0 ; d < n ; d++ )
first[c][d] = in.nextInt();
System.out.println("Enter the elements of second matrix");
for ( c = 0 ; c < m ; c++ )
for ( d = 0 ; d < n ; d++ )
second[c][d] = in.nextInt();
for ( c = 0 ; c < m ; c++ )
for ( d = 0 ; d < n ; d++ )
sum[c][d] = first[c][d] + second[c][d];
System.out.println("Sum of entered matrices:-");
```

```
for ( c = 0 ; c < m ; c++ )
      {
for ( d = 0 ; d < n ; d++ )
System.out.print(sum[c][d]+"\t");
System.out.println();
      }
   }
}
```

Output :-

Program :-21

Aim :-

```
//save by A.java
package pack;
public class A{
public void msg(){System.out.println("Hello");}
}

//save by B.java
packagemypack;
importpack.A;
class B{
public static void main(String args[]){
   A obj = new A();
obj.msg();
```

```
  }
}
```

Output :-

Program :-22

Aim :-

```
class A extends Thread {
public void run()
{
System.out.println("Thread A");
System.out.println("i in Thread A ");
for(int i=1;i<=5;i++)
{
System.out.println("i = " + i);
}
System.out.println("Thread A Completed.");
}
}
class B extends Thread {
public void run()
{
System.out.println("Thread B");
System.out.println("i in Thread B ");
```

```
for(int i=1;i<=5;i++)
{
System.out.println("i = " + i);
}
System.out.println("Thread B Completed.");
}
}
public class Main1 {
	public static void main(String[] args) {
		//life cycle of Thread
		// Thread's New State
		A threadA = new A();
		B threadB = new B();
		threadA.start();
		threadB.start();
		System.out.println("Main Thread End");
	}
}
```

Output :-

Program :-23

Aim :-

```
importjava.util.Iterator;
importjava.util.Vector;
public class SimpleVectorExample {
public static void main(String[] args) {
   //create a Vector object
   Vector v = new Vector();
v.add("1");
v.add("2");
v.add("3");
System.out.println("Getting elements of Vector");
System.out.println(v.get(0));
System.out.println(v.get(1));
System.out.println(v.get(2));
  }
}
```

Output :-

Program :-24

Aim :-

```
importjava.io.File;
importjava.io.FileInputStream;
importjava.io.FileOutputStream;
importjava.io.IOException;
public class CopyExample {
public static void main(String[] args) {
FileInputStreamins = null;
FileOutputStream outs = null;
try {
        File infile = new File("C:\\Users\\TutorialsPoint7\\Desktop\\abc.txt");
        File outfile = new File("C:\\Users\\TutorialsPoint7\\Desktop\\bbc.txt");
ins = new FileInputStream(infile);
outs = new FileOutputStream(outfile);
byte[] buffer = new byte[1024];
int length;
while ((length = ins.read(buffer)) > 0) {
outs.write(buffer, 0, length);
        }
ins.close();
outs.close();
System.out.println("File copied successfully!!");
      } catch(IOExceptionioe) {
ioe.printStackTrace();
      }
   }
}
```

Output :-

Program :-25

Aim :-

____

____

____

```
import java.io.*;

public class Test {
public static void main(String [] args) {
        // The name of the file to open.
  String fileName = "temp.txt";
        // This will reference one line at a time
        String line = null;
try {
            // FileReader reads text files in the default encoding.
FileReaderfileReader =
newFileReader(fileName);
            // Always wrap FileReader in BufferedReader.
BufferedReaderbufferedReader =
newBufferedReader(fileReader);
while((line = bufferedReader.readLine()) != null) {
System.out.println(line);
            }
            // Always close files.
bufferedReader.close();
        }
catch(FileNotFoundException ex) {
System.out.println(
                "Unable to open file '" +
fileName + "'");
        }
catch(IOException ex) {
System.out.println(
                "Error reading file '"
                + fileName + "'");
            // Or we could just do this:
            // ex.printStackTrace();
        }
    }
}
```

Output :-

Program :-26

Aim :-

___

___

___

```
class pract2
{
public static void main(String args[])
{
int a[]={5,10};
int b=5;
try
{int x=a[2]/(b-a[0]);
}
catch(ArithmeticException e)
{
System.out.println("division by zero");
}
catch(ArrayIndexOutOfBoundsException e)
{
System.out.println("Array Error");
}
int y=a[1]/a[0];
System.out.println(y);
}
}
```

Output :-

Program :-27

Aim :-

```
class Excep1
{
public static void main(String[] args)
 {
try
  {
intarr[]={5,0,1,2};
try
   {
int x=arr[3]/arr[1];
   }
catch(ArithmeticExceptionae)
   {
System.out.println("divide by zero");
   }
arr[4]=3;
  }
catch(ArrayIndexOutOfBoundsException e)
  {
```

```
System.out.println("array index out of bound exception");
  }
 }
}
```

Output :-

Program :-28

Aim :-

```
importjava.awt.*;
importjava.awt.event.ActionEvent;
importjava.awt.event.ActionListener;
public class pract1 extends Frame implements ActionListener
{
 Label l1,l2;
TextField t1;
  Button b1;
public pract1()
{
setLayout(new FlowLayout());
t1=new TextField(10);
b1=new Button("Factorial");
l1=new Label("Enter Number");
l2=new Label("Result");
```

```
add(l1);
add(t1);
add(b1);
add(l2);
b1.addActionListener(this);
setSize(200,200);
setVisible(true);
}
public void actionPerformed(ActionEventae)
{
if(ae.getSource()==b1)
{
intnum,fact,i;String result;
fact=1;
num=Integer.parseInt(t1.getText());
for(i=1;i<=num;i++)
{
fact=fact*i;
}
result=String.valueOf(fact);
l2.setText(result);
}
else
{
}
public static void main(String args[])
{
pract1 p1=new pract1();
}
}
```


Output :-

Program :-29

Aim :-

____

____

____

```
classExcp
{
public static void main(String args[])
 {
inta,b,c;
try
  {
   a=0;
   b=10;
   c=b/a;
System.out.println("This line will not be executed");
  }
catch(ArithmeticException e)
  {
System.out.println("Divided by zero");
  }
System.out.println("After exception is handled");
 }
}
```

Output :-