Bharathidasan University
Centre for Differently Abled Persons
Tiruchirappalli - 620024

Programme Name	Bachelor of Computer Applications
Course Code	20UCA3CP4
Course Title	Programming in JAVA Lab
Semester	II
Unit	I
Compiled by	Dr.M.Prabavathy & Ms.S.Patric Matharasi

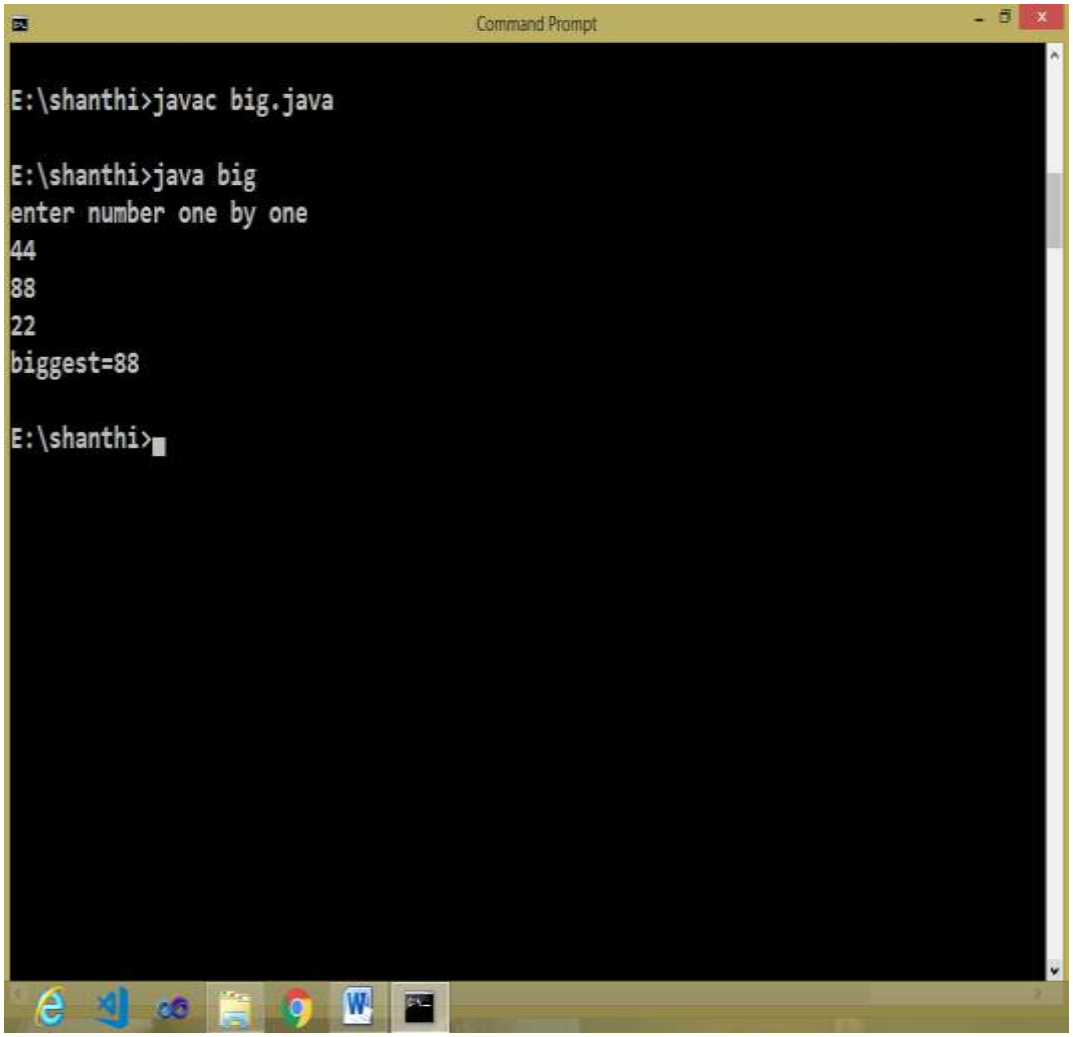
Ex.: 1

BIGGEST OF THREE NUMBERS

Program:

```
import java.io.*;
public class big
{
        public static void main(String[] args)throws IOException
        {
                BufferedReader in=new BufferedReader(new
                                        InputStreamReader(System.in));
                System.out.println("enter number one by one");
                int a=Integer.parseInt(in.readLine());
                int b=Integer.parseInt(in.readLine());
                int c=Integer.parseInt(in.readLine());
                int big=a;
                if(b>big)
                        big=b;
                if(c>big)
                        big=c;
                System.out.println("biggest="+big);
        }
}
```

Output:


Command Prompt
E:\shanthi>javac big.java
E:\shanthi>java big
enter number one by one
44
88
22
biggest=88
E:\shanthi>

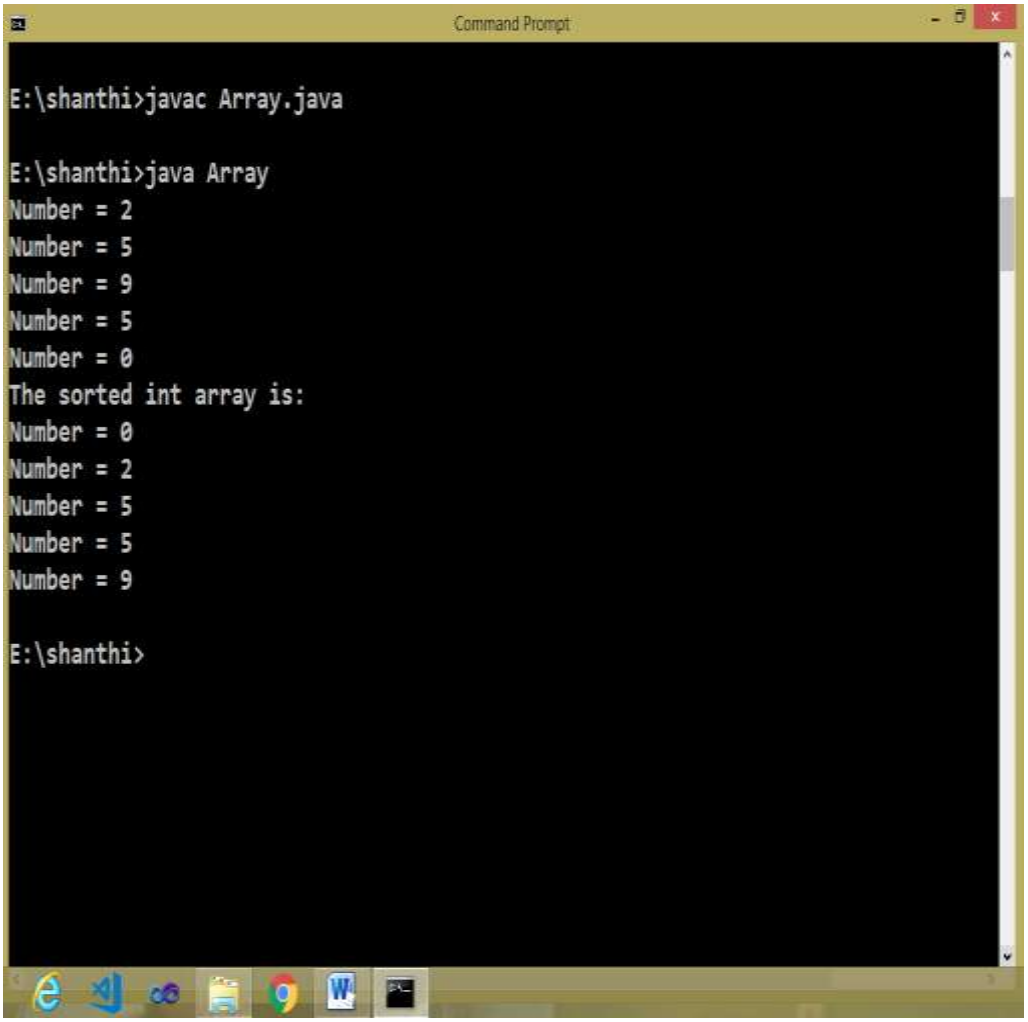
Ex.: 2

SORTING OF NUMBERS

Program:

```
importjava.util.*;
importjava.util.Arrays;
public class Array
{
        public static void main(String[ ] args)
        {
                int iArr[] = {2,5,9,5,0};
                for ( int number : iArr)
                {
                        System.out.println("Number = " +number);
                }
                Arrays.sort(iArr);
                System.out.println("The sorted int array is:");
                for ( int number: iArr)
                {
                        System.out.println("Number = " +number);
                }
        }
}
```

Output:


Command Prompt
E:\shanthi>javac Array.java
E:\shanthi>java Array
Number = 2
Number = 5
Number = 9
Number = 5
Number = 0
The sorted int array is:
Number = 0
Number = 2
Number = 5
Number = 5
Number = 9
E:\shanthi>

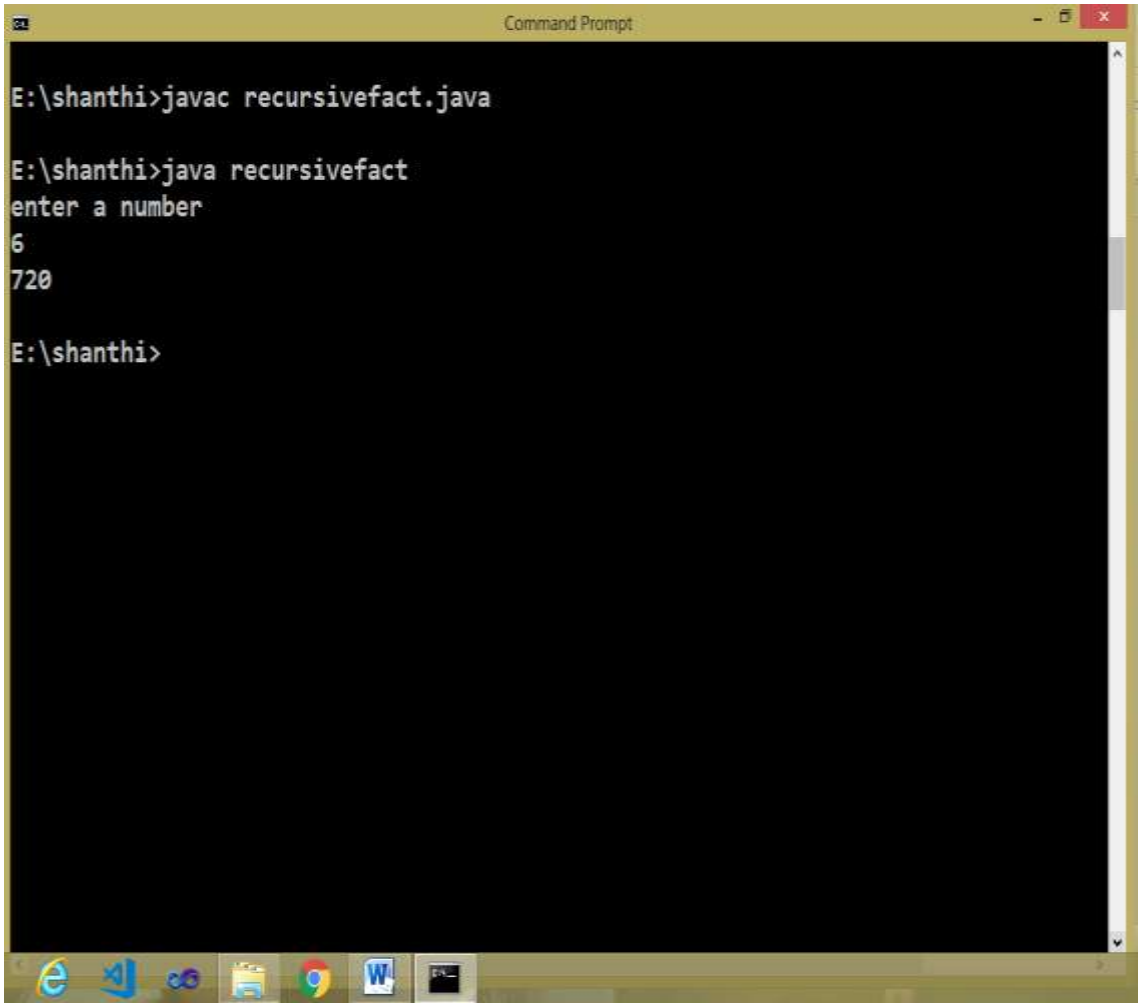
Ex.: 3

FACTORIAL USING RECURSIVE FUNCTION

Program:

```
import java.io.*;
importjava.util.*;
public class recursivefact
{
        public static long fact(long n)
        {
                if(n<0)
                        throw new RuntimeException("Exception:"+n+"is a
                                                        negative number");
                if(n==0)
                        return 1;
                else
                        return n*fact(n-1);
        }
        public static void main(String[ ] args)
        {
                System.out.println("enter a number");
                Scanner sc=new Scanner(System.in);
                long n=sc.nextLong();
                System.out.println(fact(n));
        }
}
```

Output:


Command Prompt
E:\shanthi>javac recursivefact.java
E:\shanthi>java recursivefact
enter a number
6
720
E:\shanthi>

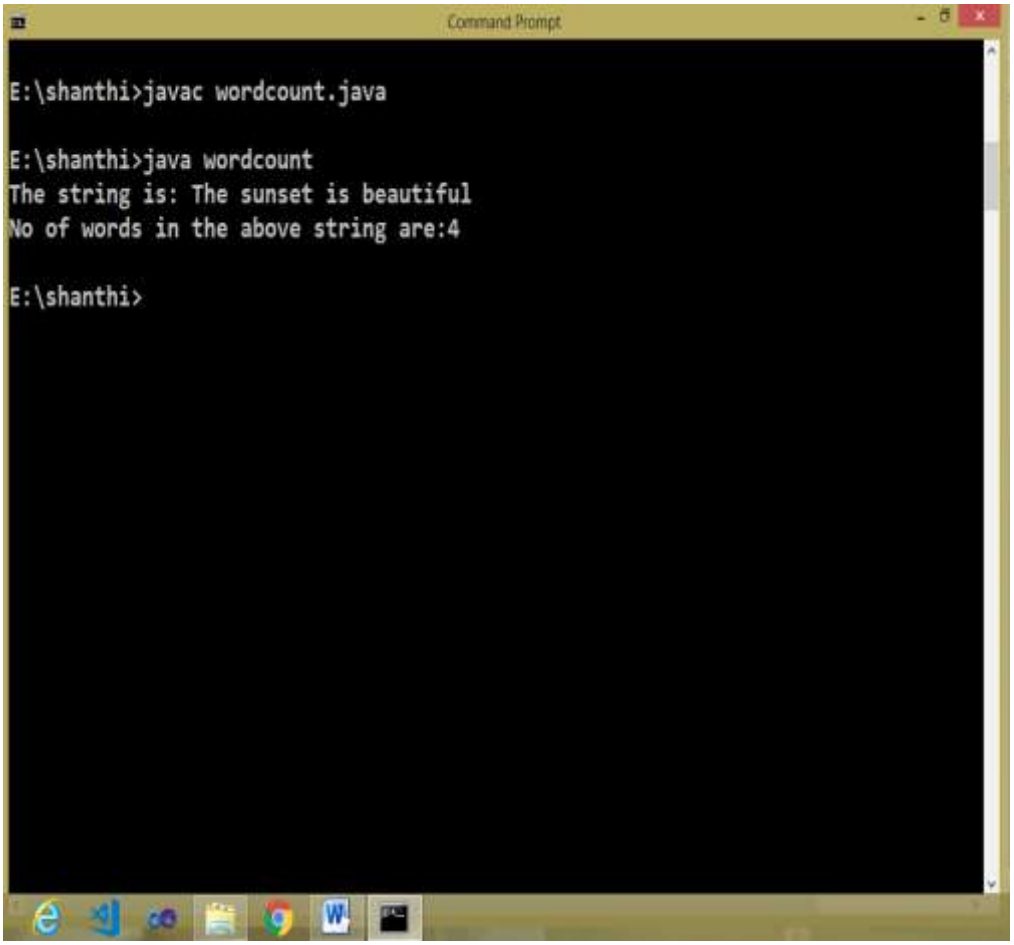
Ex.: 4

COUNTING THE WORDS IN A SENTENCE

Program:

```
import  java.io.*;
public class wordcount
{
        public static void main(String args[ ])
        {
                int flag=0;
                int count=0;
                int i=0;
                String str="The sunset is beautiful";
                while(i<str.length( ))
                {
                        if(str.charAt(i) == ' ' || str.charAt(i) == '\n' || str.charAt(i) == '\t')
                        {
                                flag=0;
                        }
                        else if (flag == 0)
                        {
                                flag= 1;
                                count++;
                        }
                        i++;
                }
                System.out.println("The string is: " +str);
                System.out.println("No of words in the above string are:"  +count);
        }
}
```

Output:


Command Prompt
E:\shanthi>javac wordcount.java
E:\shanthi>java wordcount
The string is: The sunset is beautiful
No of words in the above string are:4
E:\shanthi>

Ex.: 5

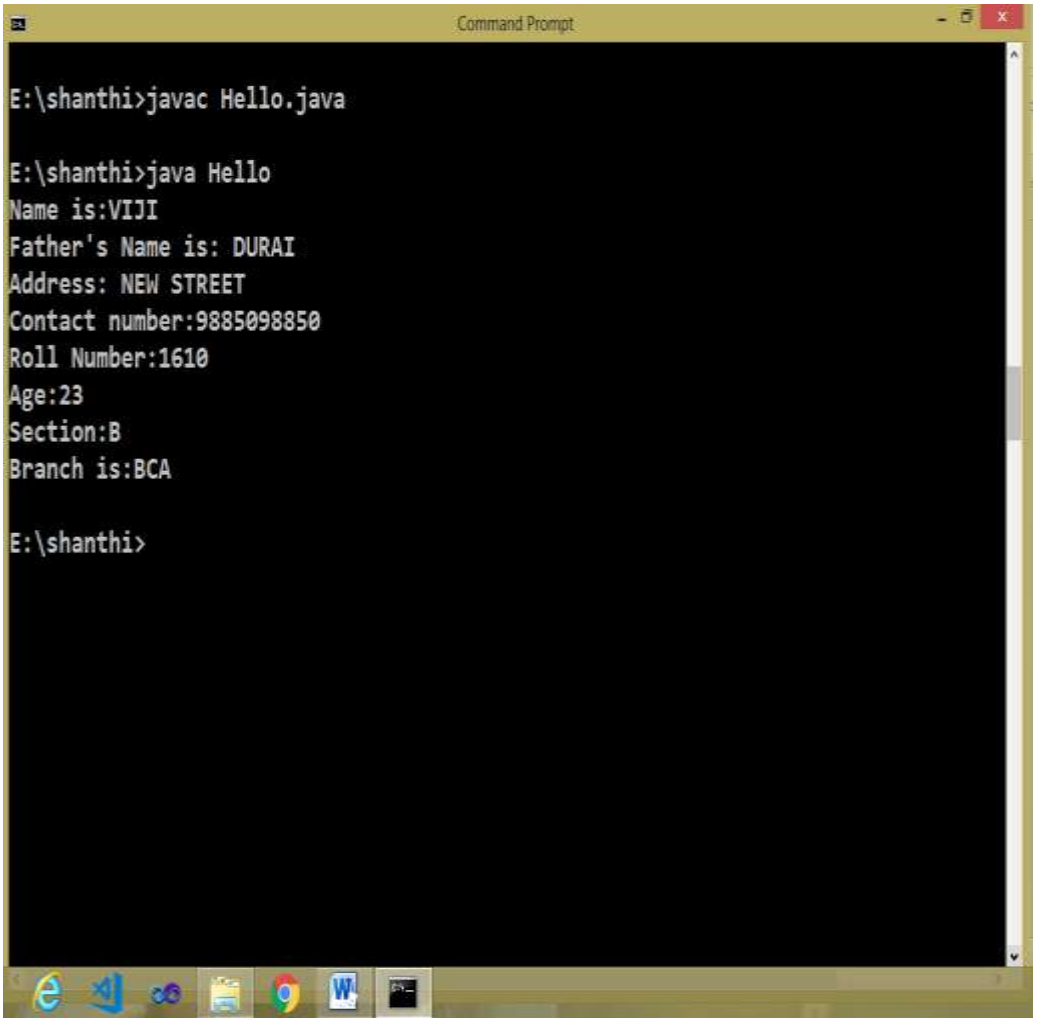
DISPLAYING STUDENT DETAILS USING INHERITANCE

Program:

```
import java.io.*;
class Personal
{
        String name=new String();
        String fname=new String();
        String add=new String();
        String phno=new String();
        Personal(String a,String b,String c,String d)
        {
                name=a;
                fname=b;
                add=c;
                phno=d;
        }
        void display()
        {
                System.out.println("Name is: "+name);
                System.out.println("Address: "+add);
                System.out.println("Father's Name is: "+fname);
                System.out.println("Contact number: "+phno);
        }
}
class Education extends Personal
{
        Int roll,age;
        char section;
```

```
        String branch=new String();
        Education(String a, String b, String c, String d, int e, int f, char  g, String h)
        {
                super(a,b,c,d);
                roll=e;
                age=f;
                section=g;
                branch=h;
        }
        void display2()
        {
                super.display();
                System.out.println("Roll Number:"+roll);
                System.out.println("Age:"+age);
                System.out.println("Section:"+section);
                System.out.println("Branch is: "+branch);
        }
}
class Hello
{
        public static void main(String args[])
        {
                Education e=new Education("VIJI","DURAI","NEW STREET",
                                                "9885098850",1610,23,'B',"BCA");
                e.display2();
        }
}
```

Output:


Command Prompt
E:\shanthi>javac Hello.java
E:\shanthi>java Hello
Name is:VIJI
Father's Name is: DURAI
Address: NEW STREET
Contact number:9885098850
Roll Number:1610
Age:23
Section:B
Branch is:BCA
E:\shanthi>

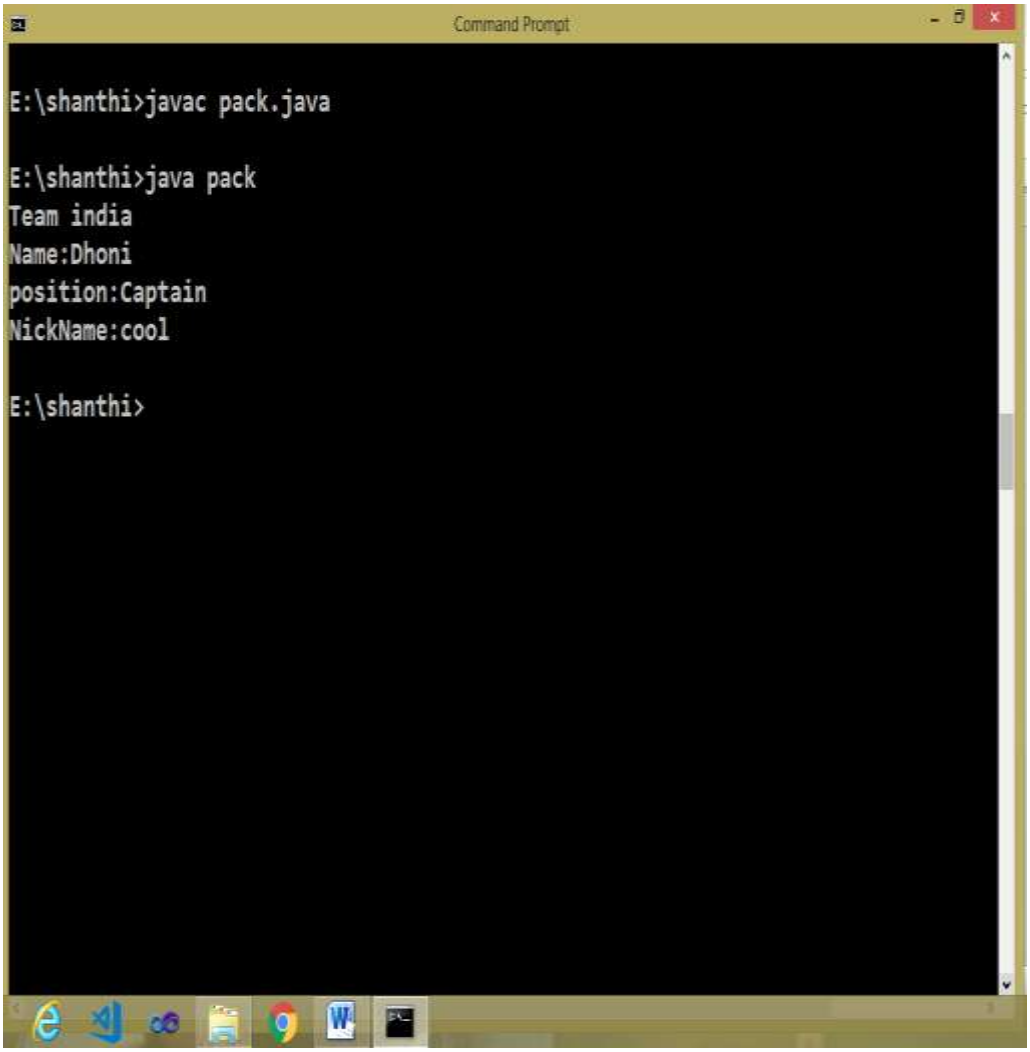
Ex.: 6

DISPLAY A TEXT USING USER DEFINED PACKAGE

Program:

```
Import india.team;
class pack
{
        public static void main(String args[])
        {
                team c=new team("Dhoni","Captain","cool");
                c.display();
        }
}

Package india;
public class team
{
        String name,position,nick;
        public team(String n,String p,String ni)
        {
                name=n;
                position=p;
                nick=ni;
        }
        public void display()
        {
                System.out.println("Team india");
                System.out.println("Name:" +name);
                System.out.println("position:" +position);
                System.out.println("NickName:" +nick);}}
```

Output:


Command Prompt
E:\shanthi>javac pack.java
E:\shanthi>java pack
Team india
Name:Dhoni
position:Captain
NickName:cool
E:\shanthi>

Ex.: 7

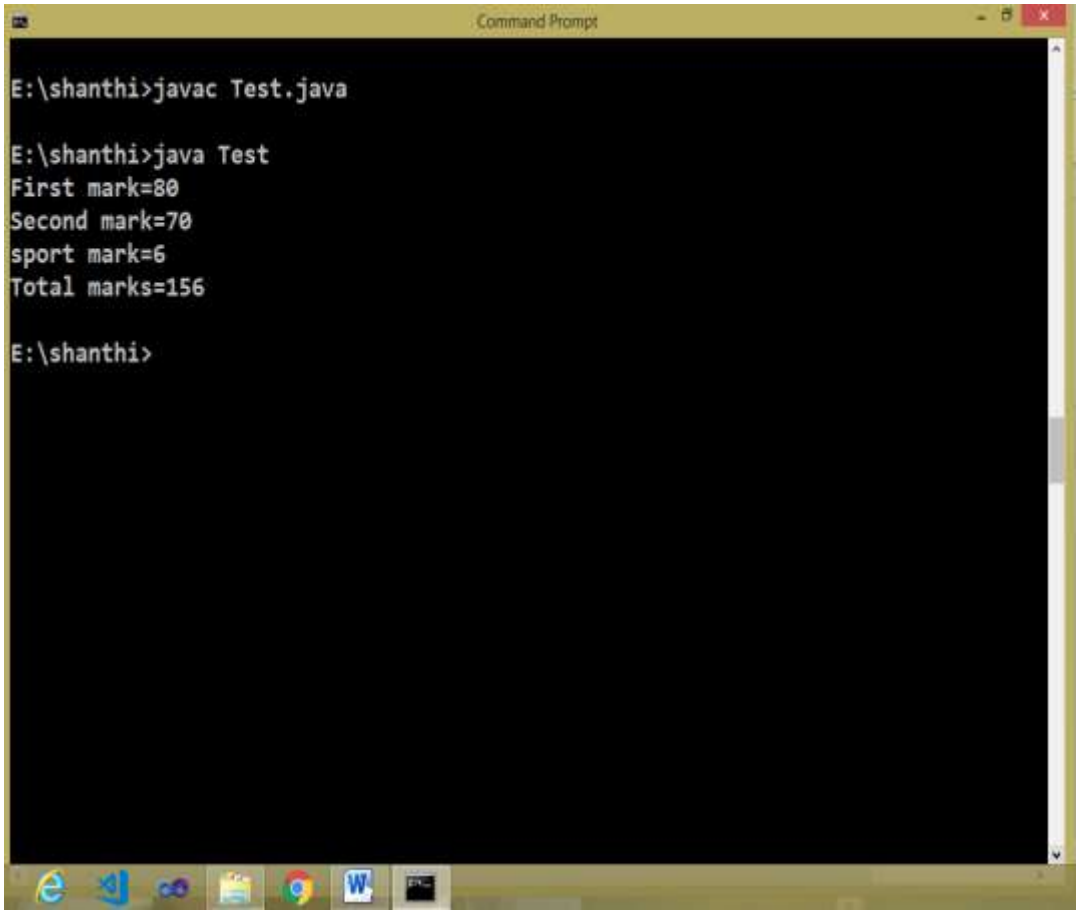
MARKS CALCULATION USING INTERFACE

Program:

```
Import java.util.*;
class Student
{
        int m1,m2;
        void getmarks(int x,int y)
        {
                m1=x;
                m2=y;
        }
        void putmarks()
        {
                System.out.println("First mark=" +m1);
                System.out.println("Second mark=" +m2);
        }
}

interface Sport
{
        int sp=6;
        void spmarks();
}

class Result extends Student implements Sport
{
        public void spmarks()
        {
```

```
            System.out.println("sport mark=" +sp);
        }
        void disp()
        {
            putmarks();
            spmarks();
            int total=m1+m2+sp;
            System.out.println("Total marks=" +total);
        }
}

class Test
{
        public static void main(String args[])
        {
            Result obj=new Result();
            obj.getmarks(80,70);
            obj.disp();
        }
}
```

Output:


Command Prompt
E:\shanthi>javac Test.java
E:\shanthi>java Test
First mark=80
Second mark=70
sport mark=6
Total marks=156
E:\shanthi>

Ex.: 8

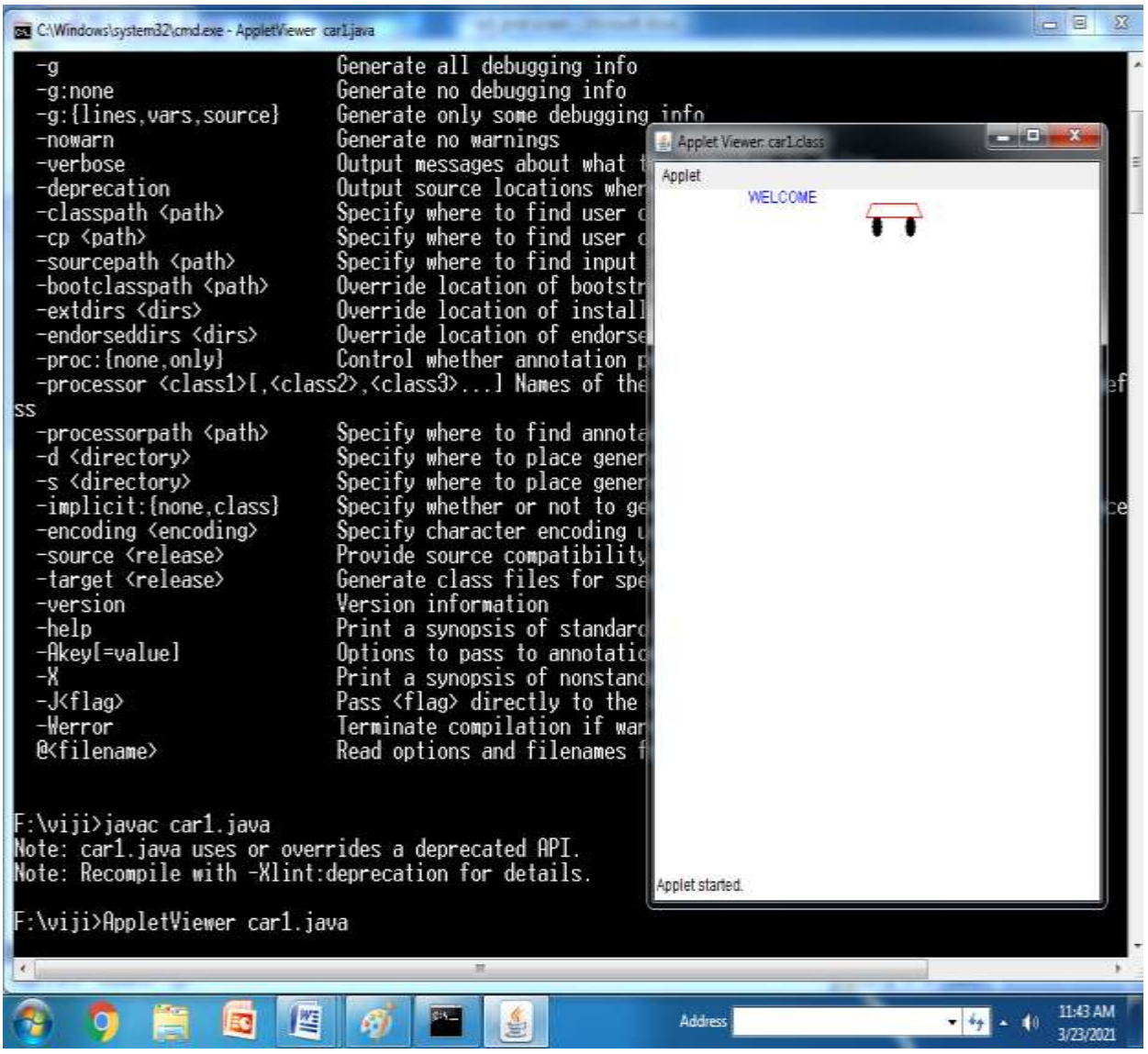
MOVING A CAR USING THREAD

Program:

```
import java.awt.*;
import java.applet.*;
public class car1 extends Applet implements Runnable
{
        Thread m=null;
        int p;
        public void start()
        {
                m=new Thread(this);
                m.start();
        }
        public void run()
        {
                while(true)
                {
                        for(p=30;p<getSize().width;p+=5)
                        {
                                repaint();
                                try
                                {
                                        m.sleep(100);
                                }
                                catch(InterruptedException e)
                                {}
                        }
                }}
```

```
        public void stop()
        {
                m.stop();
                m=null;
        }
        public void paint(Graphics g)
        {
                g.setColor(Color.blue);
                g.drawString("WELCOME",85,10);
                g.setColor(Color.red);
                g.drawLine(p,10,p+40,10);
                g.drawLine(p-5,20,p+45,20);
                g.drawLine(p,10,p-5,20);
                g.drawLine(p+40,10,p+45,20);
                g.setColor(Color.black);
                g.fillOval(p,20,10,15);
                g.fillOval(p+30,20,10,15);
        }
}
//<applet code=car1.class width=400 height=500></applet>
```

Output:

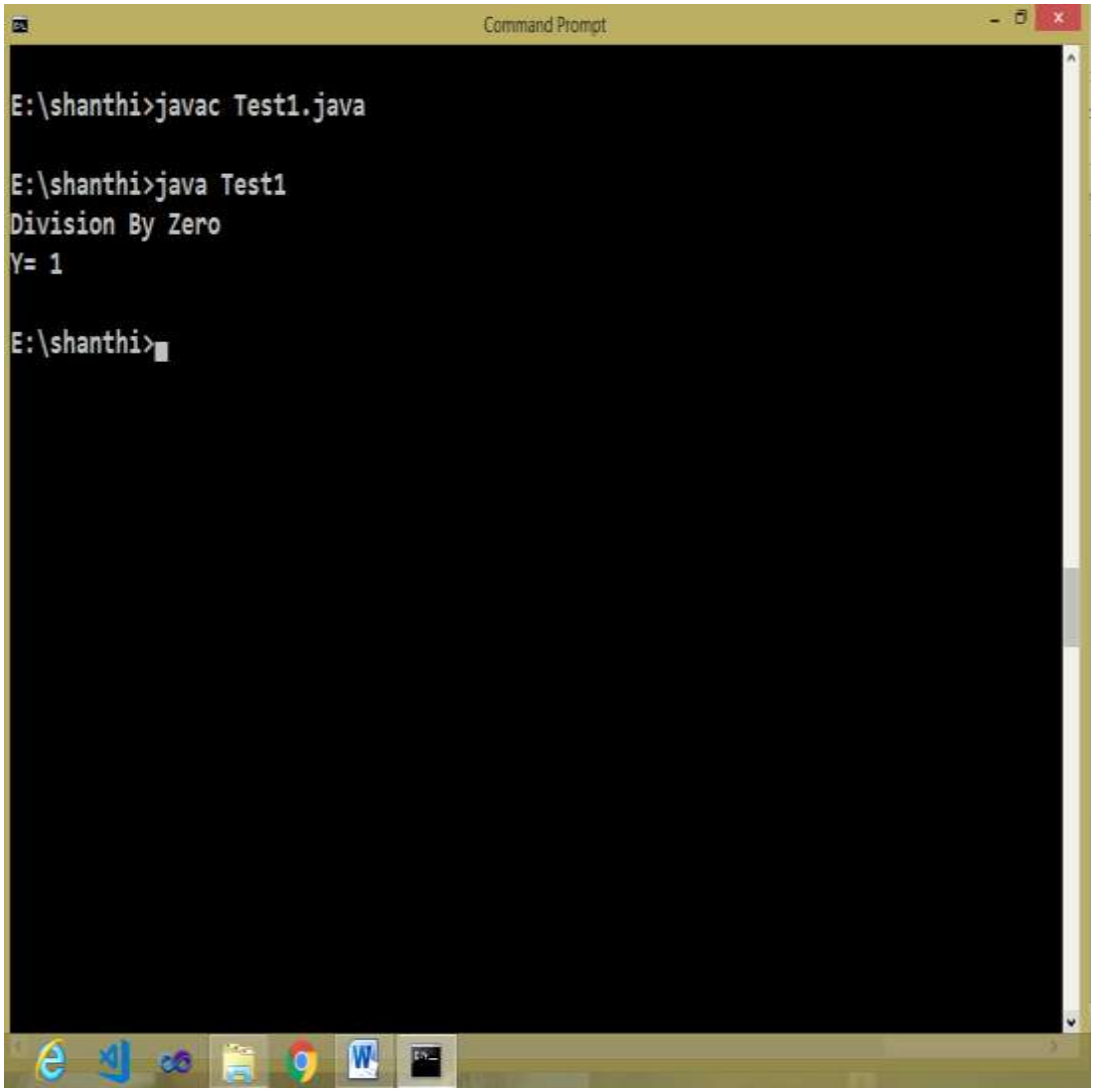

C:\Windows\system32\cmd.exe - AppletViewer car1.java
-g Generate all debugging info
-g:none Generate no debugging info
-g:{lines,vars,source} Generate only some debugging info
-nowarn Generate no warnings
-verbose Output messages about what
-deprecation Output source locations wher
-classpath <path> Specify where to find user
-cp <path> Specify where to find user
-sourcepath <path> Specify where to find input
-bootclasspath <path> Override location of bootstr
-extdirs <dirs> Override location of install
-endorseddirs <dirs> Override location of endorse
-proc:{none,only} Control whether annotation
-processor <class1>[,<class2>,<class3>...] Names of the
ss
-processorpath <path> Specify where to find annota
-d <directory> Specify where to place gener
-s <directory> Specify where to place gener
-implicit:{none,class} Specify whether or not to ge
-encoding <encoding> Specify character encoding
-source <release> Provide source compatibility
-target <release> Generate class files for spe
-version Version information
-help Print a synopsis of standard
-Akey[=value] Options to pass to annotatio
-X Print a synopsis of nonstand
-J<flag> Pass <flag> directly to the
-Werror Terminate compilation if war
@<filename> Read options and filenames
F:\viji>javac car1.java
Note: car1.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
F:\viji>AppletViewer car1.java
Applet Viewer: car1.class
Applet
WELCOME
Applet started.
Address
11:43 AM
3/23/2021

Ex.: 9

Program:

```
import java.io.*;
importjava.util.*;
class Test1
{
	public static void main(String args[])
	{
		int a=20, b=10, c=10;
		try
		{
			int x= a/(b-c);
			System.out.println("X= "+x);
		}
		catch(ArithmeticException e)
		{
			System.out.println("Division By Zero");
		}
		int y=a/(b+c);
		System.out.println("Y= "+y);
	}
}
```

Output:


Command Prompt
E:\shanthi>javac Test1.java
E:\shanthi>java Test1
Division By Zero
Y= 1
E:\shanthi>

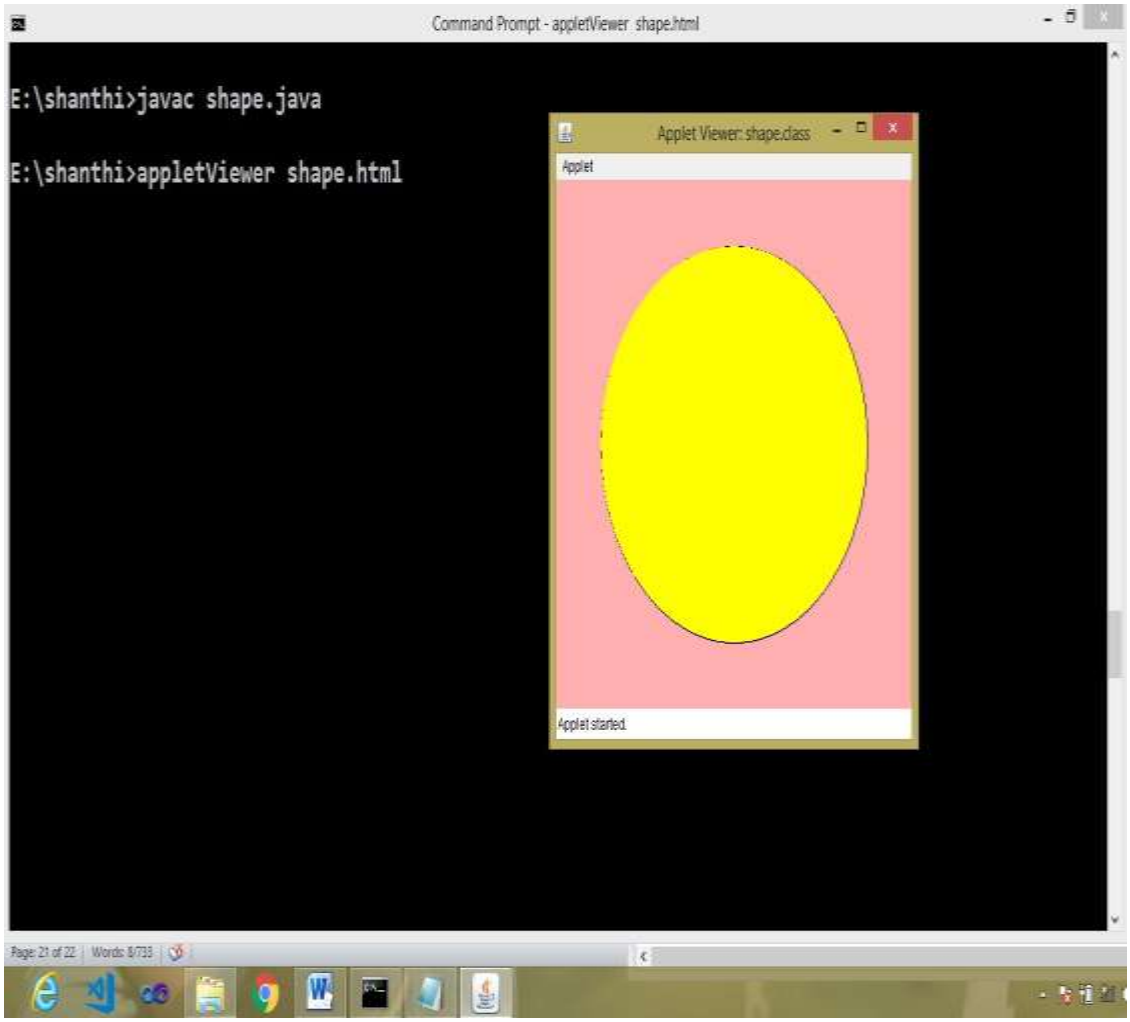
Ex.: 10

Program:

```
Import java.applet.*;
Import java.awt.*;
public class shape extends Applet
{
        public void paint(Graphics x)
        {
                setBackground(Color.pink);
                x.drawOval(50,50,300,300);
                x.setColor(Color.yellow);
                x.fillOval(50,50,300,300);
        }
}
```

```
<html>
        <title>applet</title>
        <applet code="shape.class" width=400 height=400>
        </applet>
</html>
```

Output:


Command Prompt - appletViewer shape.html
E:\shanthi>javac shape.java
E:\shanthi>appletViewer shape.html
Applet Viewer: shape.class
Applet
Applet started.