B M S COLLEGE OF ENGINEERING

(Autonomous College under VTU)

Department of Computer Applications

LABORATORY CERTIFICATE

This is to certify that **Ganesha S** has satisfactorily completed the course of practical in "**Programming using Java"** Laboratory (18MCA3PCJP) prescribed by BMS College of Engineering (Autonomous college under VTU) 3rdSemester MCA course in this college during the year 2019- 2020.

Signature of Batch in charge **Signature of HOD**

Student Name: Ganesha S **Examiner 1:**

USN:1BF18MCA06 **Examiner 2:**

CONTENTS

PROGRAM 1: A factory gives the following rates of commission for monthly sales of its products

Sales	Commission
Below Rs 10000/-	No Commission
10001 – 15000	5%
15001 – 20000	7.5%
Above 20000	10%

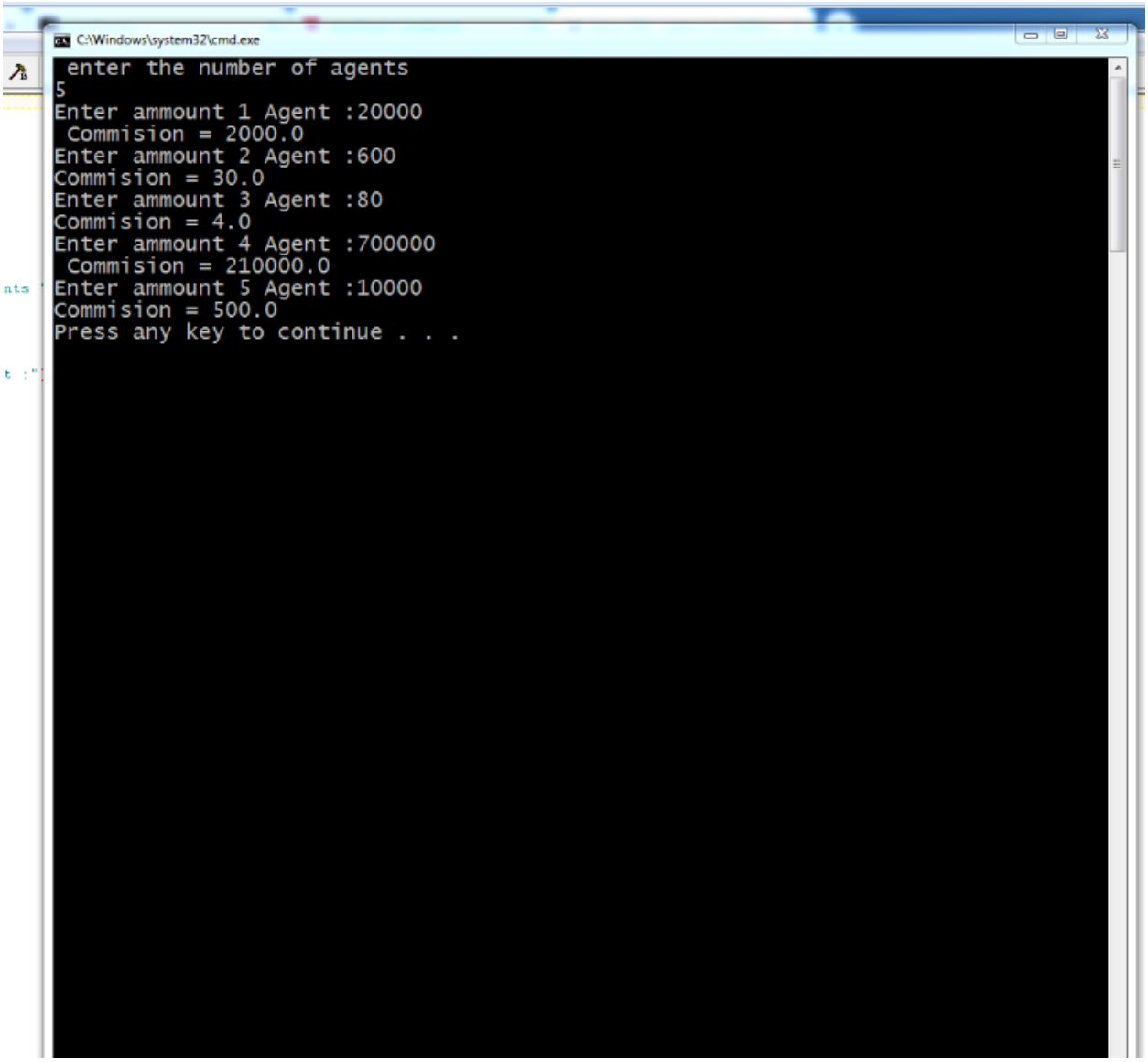
Write a java program to create 5 sample salesmen and print the commission.

SOURCE CODE:

```
importjava.util.Scanner;
class Commission
{
	doubleamt;
	doublecomm;
	int agents;
voidcalc()
{
	Scanner SC=new Scanner(System.in);
	System.out.println(" enter the number of agents ");
	agents=SC.nextInt();
	for (int i = 1 ; i<=agents ; i++)
		{
	System.out.print("Enter ammount " +i+ " Agent :");
	amt=SC.nextDouble();

	if(amt>0 &&amt<=10000)
		{
			comm = amt * 0.05 ;
			System.out.println("Commision = " +comm);
			  }
	else if(amt>10001 &&amt<=15000)
		{
	comm = amt * 0.08;
	System.out.println(" Commision = " +comm );
		}
```

```
        else if(amt>15001 &&amt<=20000)
                {
            comm = amt * 0.10;
            System.out.println(" Commision = " +comm );
                }
        else if(amt>20001 &&amt<=30000)
                {
            comm = amt * 0.25;
            System.out.println(" Commision = " +comm );
                }
        else if(amt>30000)
                {
            comm = amt * 0.30;
            System.out.println(" Commision = " +comm );
                }
            }
    }
}

public class AgentCom
{
public static void main(String args[])
    {
      Commission C = new Commission();
C.calc();
    }
 }
```

OUTPUT:

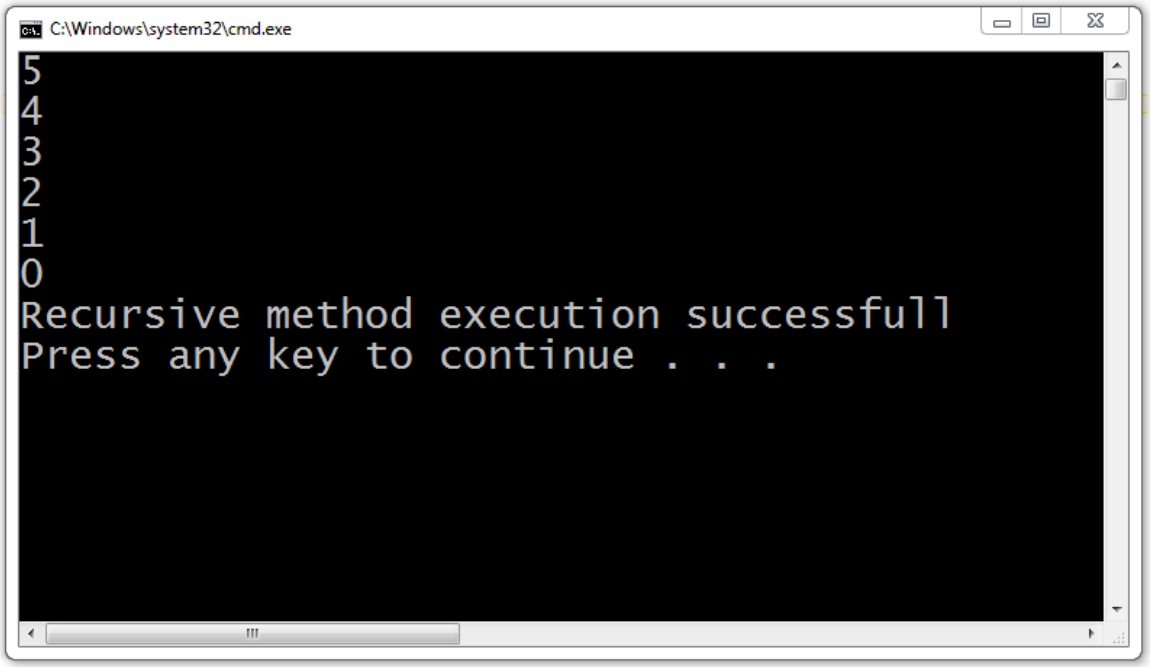

C:\Windows\system32\cmd.exe
enter the number of agents
5
Enter ammount 1 Agent :20000
Commision = 2000.0
Enter ammount 2 Agent :600
Commision = 30.0
Enter ammount 3 Agent :80
Commision = 4.0
Enter ammount 4 Agent :700000
Commision = 210000.0
Enter ammount 5 Agent :10000
Commision = 500.0
Press any key to continue . . .

PROGRAM 2: Write a recursive method CountDown( ) that takes integer n as its parameter.it prints the integers from n down to 0, one per line, and then in prints "Recursive Method Execution Successful".

SOURCE CODE:

```
public class recursive {
  static void countDown(int d) {
        if(d>=0)
         {
                System.out.println(d);
                countDown(d-1);

                 }
        }

   public static void main(String[] args)
{
      countDown(5);
      System.out.println("Recursive method execution successfull");
   }
}
```

OUTPUT:


C:\Windows\system32\cmd.exe
5
4
3
2
1
0
Recursive method execution successfull
Press any key to continue . . .

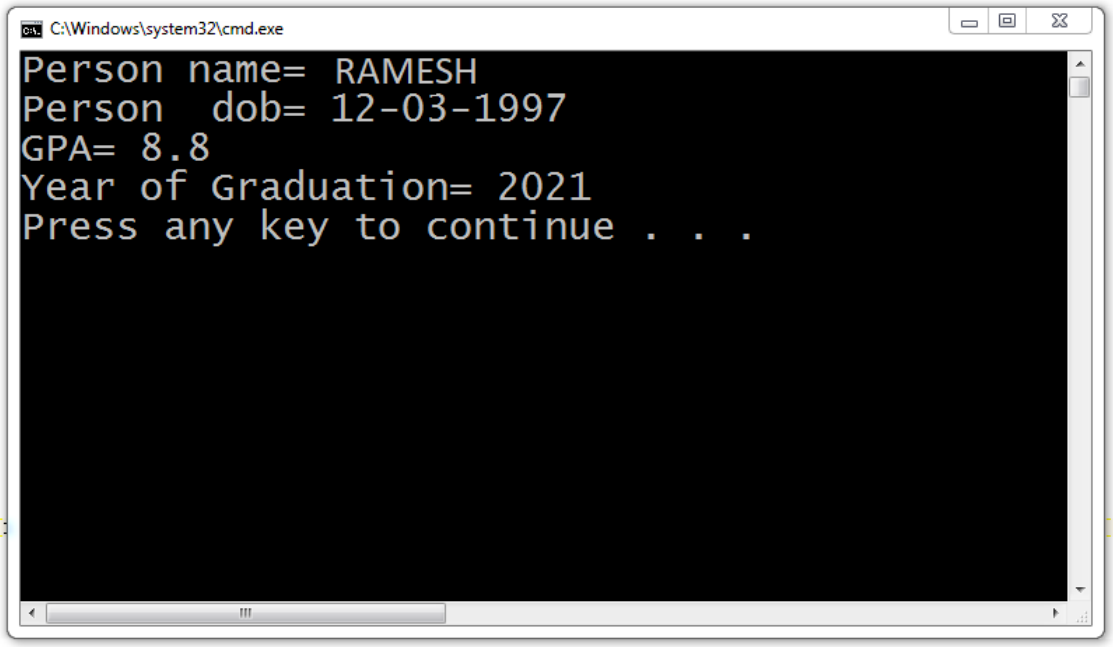
PROGRAM 3: create a person class with private instance variables for the person's name and birth date. Add appropriate accessor methods for these variables. Then create a subclass CollegeGraduate with private instance variables for the Student's GPA and year of Graduation and appropriate accessors for these variables. Don't forget to include appropriate constructors for your classes. Then create a class with a main( ) method that demonstrates your classes.

SOURCE CODE:

```
class Person
{
   private String name;
   privateint DOB;
   //constructor
   public Person(String name,int DOB)
   {
      this.name=name;
      this.DOB=DOB;
   }
   //method
   public String toString()
   {
       return "name="+name+"\nDOB="+DOB;
   }
}
classCollegeGraduate extends Person
{
   privateint GPA;
   privateint YOG;
```

```
    publicCollegeGraduate(String name,intDOB,intGPA,int YOG)
    {
        super(name,DOB);
        this.GPA=GPA;
        this.YOG=YOG;
    }
    public String toString()
    {
         return "\n\nDetails of student is \n"+super.toString()+"\nGPA="+GPA+"\nYOG="+YOG;
    }
}
public class studentnew
{
    public static void main(String args[])
      {
                    studentnewob = new studentnew();
        Person a=new Person("Maya",1998);
        CollegeGraduate cg=new CollegeGraduate("Maya",1998,82,2018);
        System.out.println(a);
        System.out.println(cg);
      }
}
```

OUTPUT:


C:\Windows\system32\cmd.exe
Person name= RAMESH
Person dob= 12-03-1997
GPA= 8.8
Year of Graduation= 2021
Press any key to continue . . .

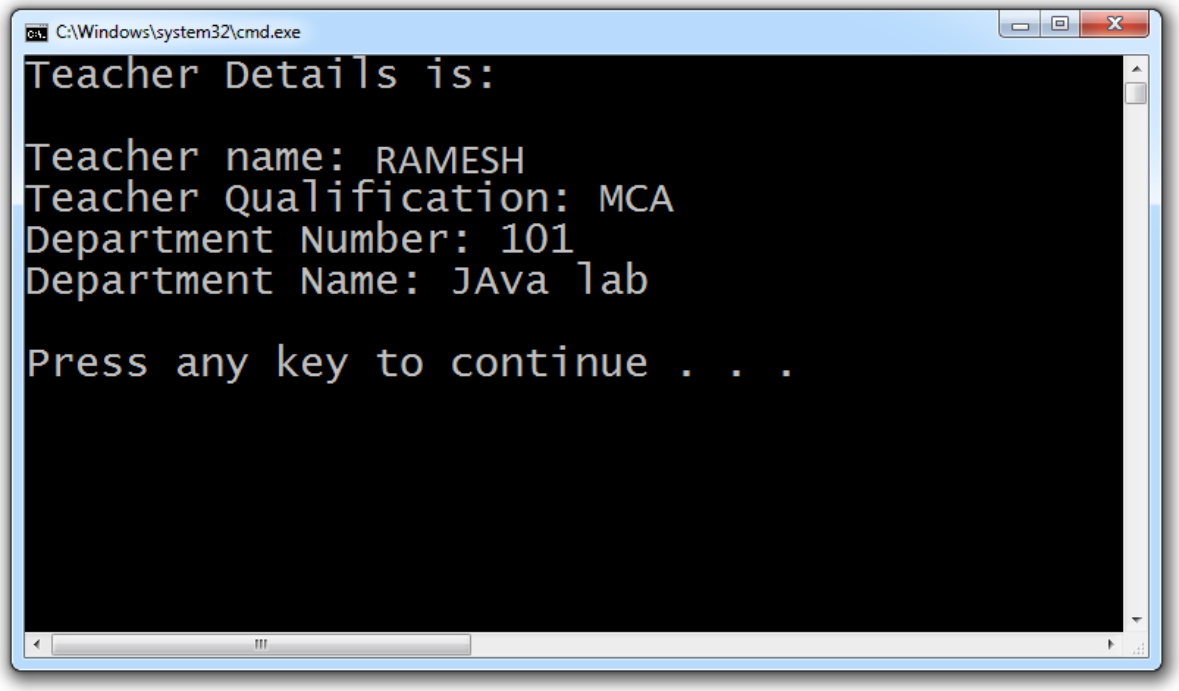
PROGRAM 4: Develop a class Teacher contains two fields, Name and Qualification. Extended the class to Department, it contains Dept.No and Dept.Name. An interface named as College contains one field Name of the College. Using above classes and interface get the appropriate information and display it.

SOURCE CODE:

```
interface College{
	String College_name="p";
	public void display();
}
class Teacher{
	String Name;
	String Qualification;
	Teacher(String tn,Stringtq)
	{

		Name=tn;
		Qualification=tq;
	}
}
class Department extends Teacher{
	intDept_No;
	String Dept_Name;
	Department(String tn,Stringtq,intd,Stringdn)
	{
		super(tn,tq);
		Dept_No=d;
		Dept_Name=dn;
	}
	void show()
	{
		System.out.println("Teacher name: "+Name+"\nTeacher Qualification:
"+Qualification+"\nDepartment Number: "+Dept_No+"\nDepartment Name:
"+Dept_Name+"\n");
	}
}
```

```
public class collegeMain implements College
{
        public void display()
        {
                System.out.println("Teacher Details is:\n");
        }
        public static void main(String args[])
                {
                Department ob=new Department("Chatur ","MCA",101,"JAva lab");
                College i=new collegeMain();
                i.display();
                ob.show();

                }
}
```

OUTPUT:


C:\Windows\system32\cmd.exe
Teacher Details is:
Teacher name: RAMESH
Teacher Qualification: MCA
Department Number: 101
Department Name: JAva lab
Press any key to continue . . .

PROGRAM 5: 5. Write a program to implement package program.

SOURCE CODE:

```
//Package Creation
package opr;
public class Mathop{
public int add(int a,int b)
{
return a+b;
}
public int sub(int a,int b)
{
return a-b;
}
}


//Importing Package
import opr.*;
public class Math{
public static void main(String args[])
{
Mathop m =new Mathop();
System.out.println("Adding two numbers: "+m.add(5,6));
System.out.println("SUbtracting two numbers:  "+m.sub(6,2));
}
}
```

Output:

C:\WINDOWS\system32\cmd.exe

```
Adding two numbers: 11
SUbtracting two numbers:  4
Press any key to continue . . .
```

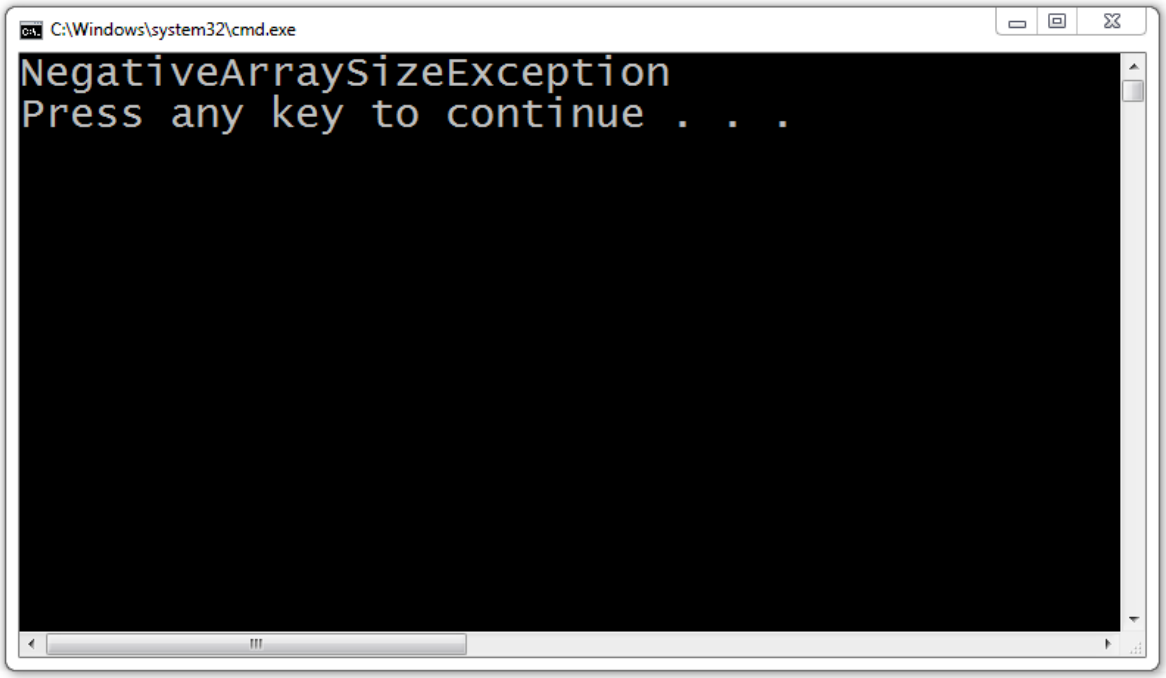
PROGRAM 6: Write a java program to throw the following exception,

A) Negative Array Size

SOURCE CODE:

```
importjava.util.*;
import java.io.*;
public class Collegetest
{
   public static void main(String args[])throws IOException
     {

         Scanner sc=new Scanner(System.in);
           try{
                      int[] arr=new int[-2];
                    }
     catch(NegativeArraySizeException e)
        {
                      System.out.println("NegativeArraySizeException");
        }
     }
}
```

OUTPUT:

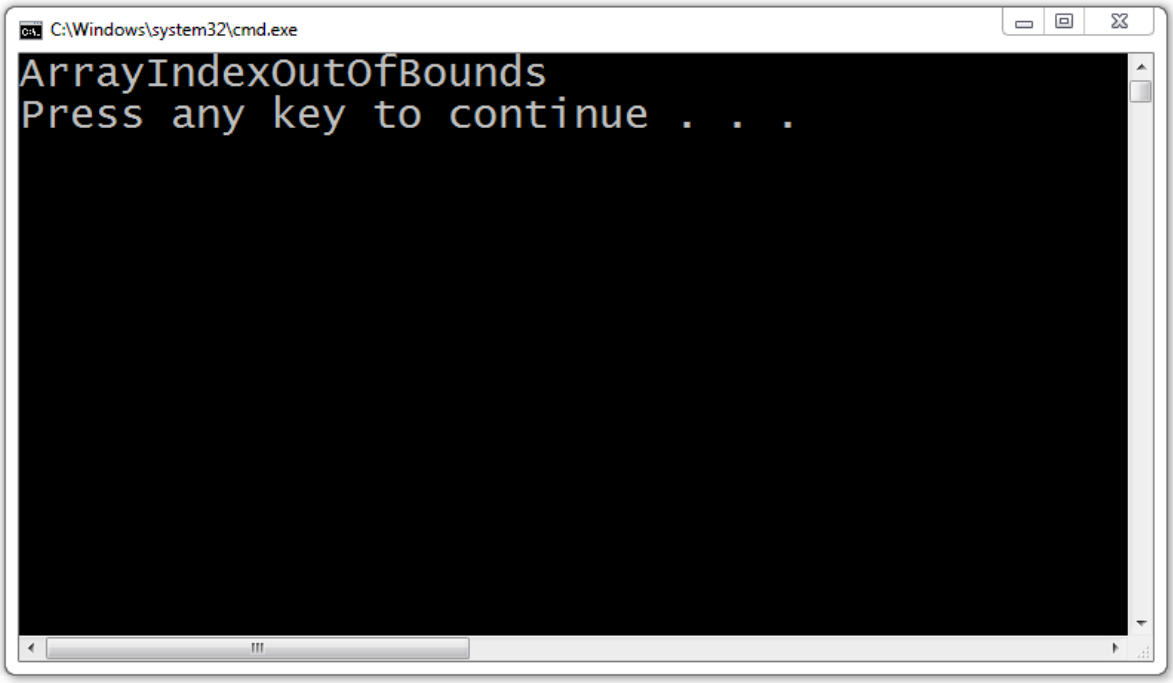

C:\Windows\system32\cmd.exe
NegativeArraySizeException
Press any key to continue . . .

PROGRAM 6: B) Array Index out of Bounds

SOURCE CODE:

```
import java.io.*;
classCollegetest
{
  public static void main(String args[])throws IOException
  {
    try{
      int a[]=new int[10];
      //Array has only 10 elements
      a[11] = 9;
    }
    catch(ArrayIndexOutOfBoundsException e){
       System.out.println ("ArrayIndexOutOfBounds");
    }
  }
}
```

OUTPUT:


C:\Windows\system32\cmd.exe
ArrayIndexOutOfBounds
Press any key to continue . . .

PROGRAM 7: Write a program using Synchronized Threads, which demonstrates Producer Consumer concept.

SOURCE CODE:

```
public class ProducerConsumer {
  public static void main(String[] args) {
    CubbyHole c = new CubbyHole();
    Producer p1 = new Producer(c, 1);
    Consumer c1 = new Consumer(c, 1);
    p1.start();
    c1.start();
  }
}
classCubbyHole {
  privateint contents;
  privateboolean available = false;

  public synchronized int get() {
    while (available == false) {
      try {
        wait();
      } catch (InterruptedException e) {}
    }
    available = false;
    notifyAll();
    return contents;
  }
  public synchronized void put(int value) {
    while (available == true) {
      try {
        wait();
      } catch (InterruptedException e) { }
    }
    contents = value;
    available = true;
    notifyAll();
  }
}
class Consumer extends Thread {
  privateCubbyHole cubbyhole;
  privateint number;

  public Consumer(CubbyHole c, int number) {
    cubbyhole = c;
    this.number = number;
```

```
  }
  public void run() {
    int value = 0;
    for (int i = 0; i < 5; i++) {
      value = cubbyhole.get();
      System.out.println("Consumer no:" + this.number + " got: " + value);
    }
  }
}
class Producer extends Thread {
  privateCubbyHole cubbyhole;
  privateint number;
  public Producer(CubbyHole c, int number) {
    cubbyhole = c;
    this.number = number;
  }
  public void run() {
    for (int i = 0; i < 5; i++) {
      cubbyhole.put(i);
      System.out.println("Producer no:" + this.number + " put: " + i);
      try {
        sleep((int)(Math.random() * 100));
      } catch (InterruptedException e) { }
    }
  }
}
```

OUTPUT:

```
C:\Windows\system32\cmd.exe
Produced value 1 put: 0
Consumed value 1 got: 0
Produced value 1 put: 1
Consumed value 1 got: 1
Produced value 1 put: 2
Consumed value 1 got: 2
Produced value 1 put: 3
Consumed value 1 got: 3
Produced value 1 put: 4
Consumed value 1 got: 4
Produced value 1 put: 5
Consumed value 1 got: 5
Produced value 1 put: 6
Consumed value 1 got: 6
Produced value 1 put: 7
Consumed value 1 got: 7
Produced value 1 put: 8
Consumed value 1 got: 8
Produced value 1 put: 9
Consumed value 1 got: 9
Press any key to continue . . .
```

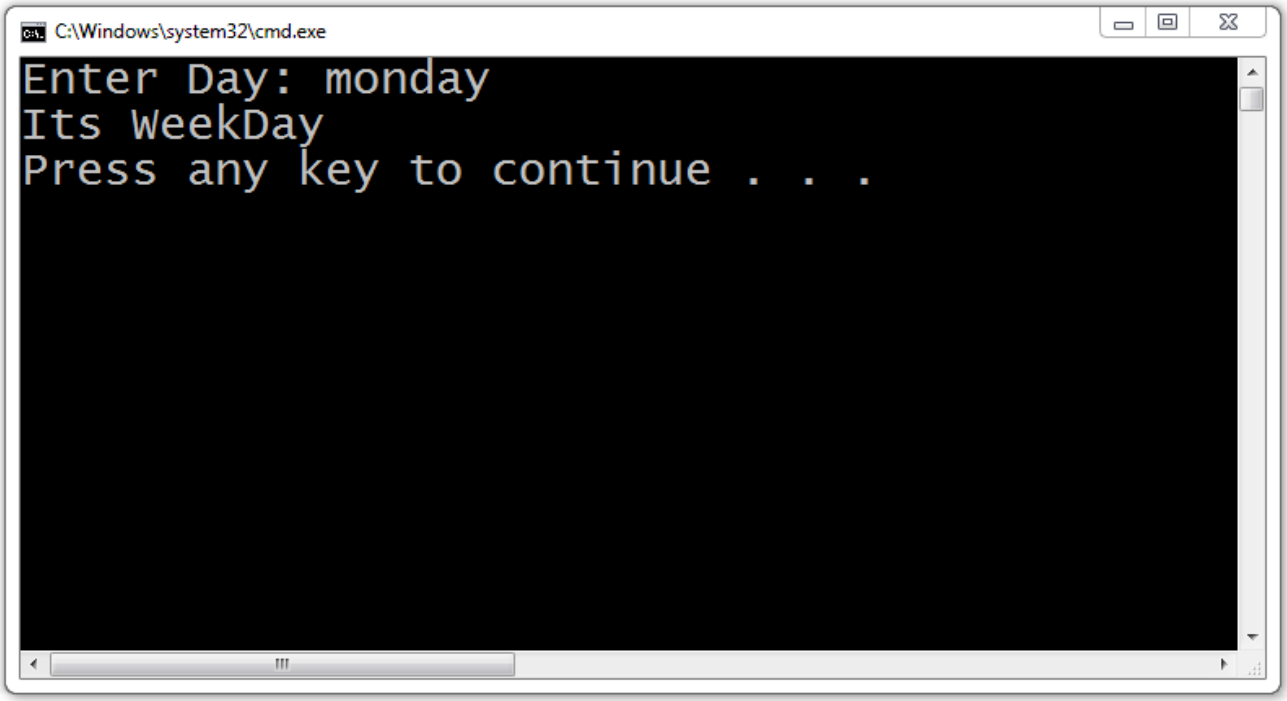
PROGRAM 8: A) Create an enumeration DayofWeek, with seven values Sunday through Saturday. Add a method is-workday( ) to the DayofWeek class that returns true if the value on which it is called Monday through Friday. For example, the DayofWeek.Sunday.IsWorkDay( ) returns False.

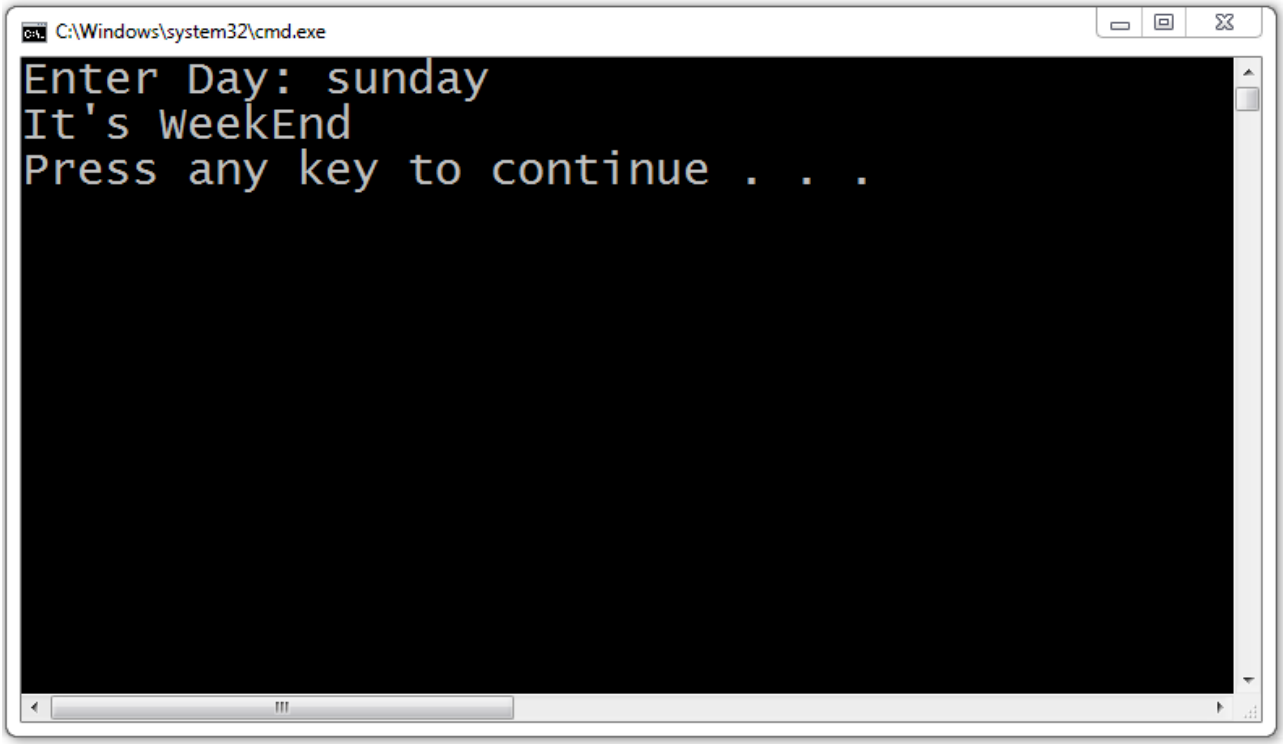
SOURCE CODE:

```
importjava.util.Scanner;

// An Enum class
enum Day
{
   SUNDAY, MONDAY, TUESDAY, WEDNESDAY,
   THURSDAY, FRIDAY, SATURDAY;
}

// main()

public class Enum
{
   Day day;

   // Constructor
publicEnum(Day day)
   {
this.day = day;
   }

publicbooleandayIsLike()
   {
switch (day)
       {
case SATURDAY:
case SUNDAY:
return false;

default:
return true;

       }
   }

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

```
booleanst;
String str;

 Scanner SC=new Scanner(System.in);

                    System.out.print("Enter Day: ");
                    str=SC.nextLine();

str = str.toUpperCase();

Enum t1 = new Enum(Day.valueOf(str));
st=t1.dayIsLike();
if(st==true){
System.out.println("Its WeekDay");
}
else
{
System.out.println("It's WeekEnd");}
    }
}
```

OUTPUT:


C:\Windows\system32\cmd.exe
Enter Day: monday
Its WeekDay
Press any key to continue . . .

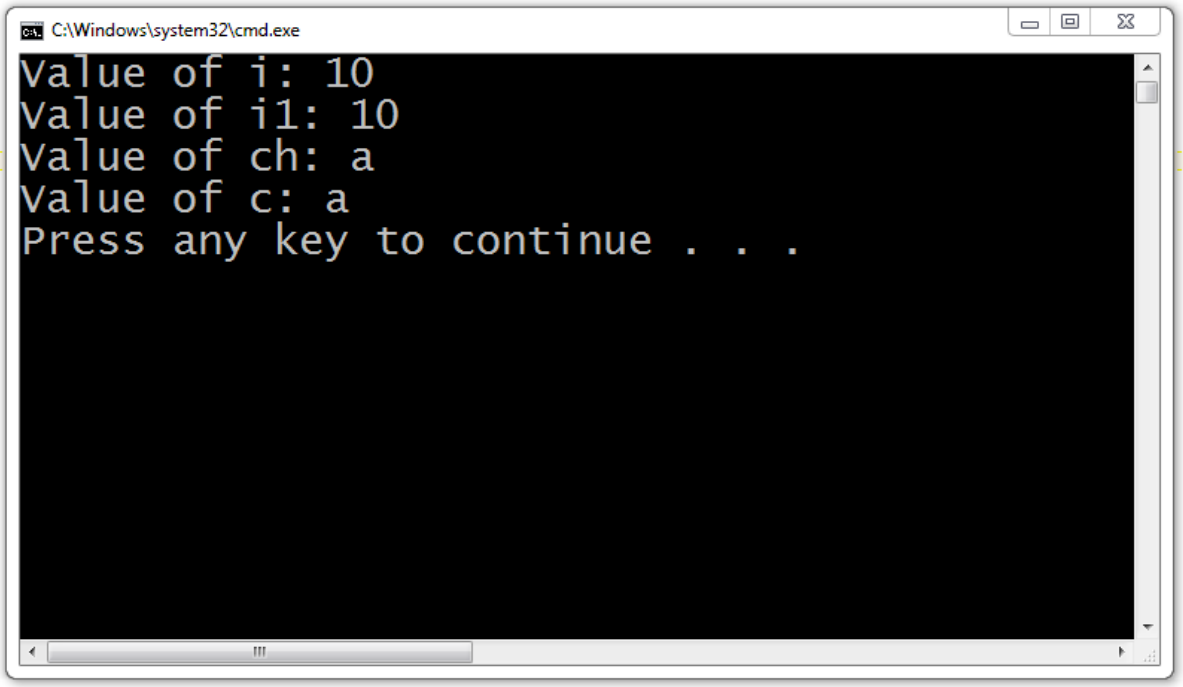

C:\Windows\system32\cmd.exe
Enter Day: sunday
It's WeekEnd
Press any key to continue . . .

PROGRAM 8: B) Write a java program demonstrate Boxing and Unboxing concept.

SOURCE CODE:

```
public class unboxing {

public static void main (String args[]){

     Integer ob = new Integer("2526");//unboxing


int i = ob.intValue();//boxing


System.out.println("Converting in boxing "+i);

System.out.println("\nConverting in unboxing "+ob);

   }

}
```

OUTPUT:

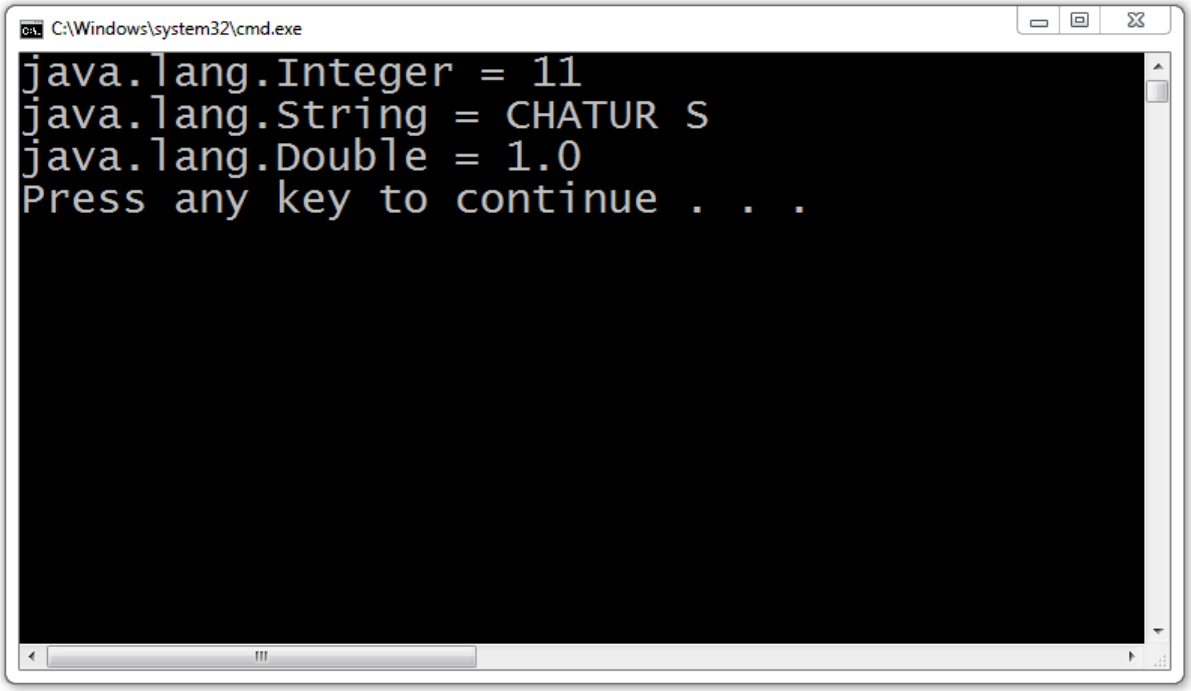

C:\Windows\system32\cmd.exe
Value of i: 10
Value of i1: 10
Value of ch: a
Value of c: a
Press any key to continue . . .

PROGRAM 9: Write a java program demonstrating java Generic.

SOURCE CODE:

```
// A Simple Java program to show working of user defined
// Generic functions

class Generic
{
   // A Generic method example
static<T> void genericDisplay (T element)
   {
System.out.println(element.getClass().getName() +
                        " = " + element);
   }

   // Driver method
public static void main(String[] args)
   {
       // Calling generic method with Integer argument
genericDisplay(11);

      // Calling generic method with String argument
genericDisplay("CHATUR S ");

      // Calling generic method with double argument
genericDisplay(1.0);
   }
}
```

OUTPUT:

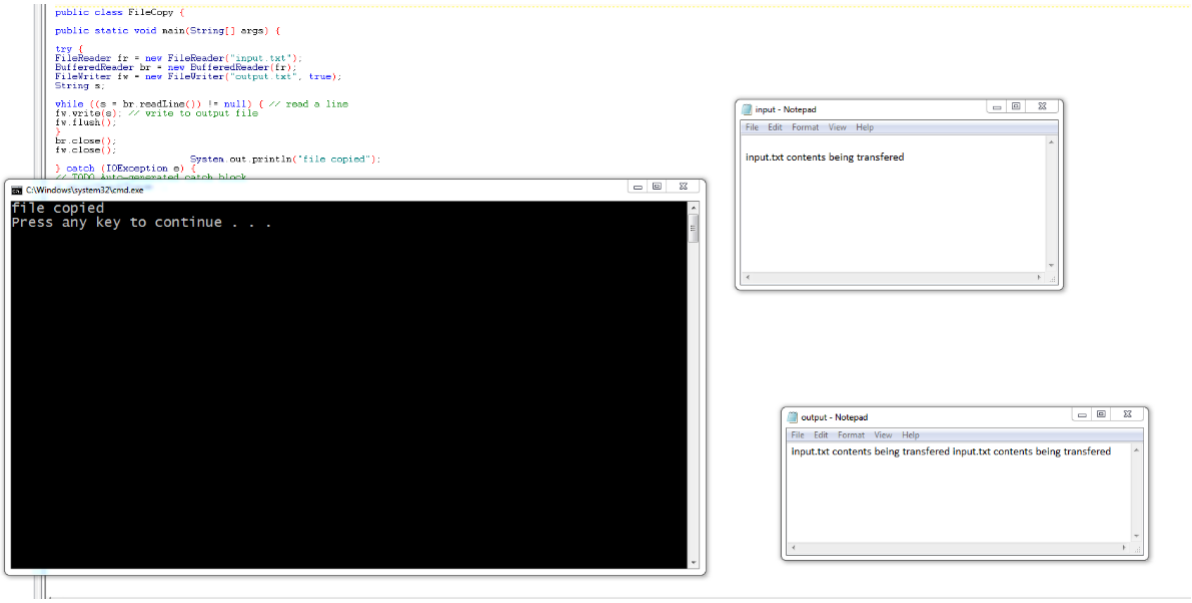

C:\Windows\system32\cmd.exe
java.lang.Integer = 11
java.lang.String = CHATUR S
java.lang.Double = 1.0
Press any key to continue . . .

PROGRAM 10: Write a java program to copy the content from input.txt to output.txt using file input stream and file output stream.

SOURCE CODE:

```
importjava.io.BufferedReader;
importjava.io.FileReader;
importjava.io.FileWriter;
importjava.io.IOException;

public class FileCopy {

public static void main(String[] args) {

try {
FileReaderfr = new FileReader("input.txt");
BufferedReaderbr = new BufferedReader(fr);
FileWriterfw = new FileWriter("output.txt", true);
String s;

while ((s = br.readLine()) != null) { // read a line
fw.write(s); // write to output file
fw.flush();
}
br.close();
fw.close();
System.out.println("file copied");
} catch (IOException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
}
}
```

OUTPUT:

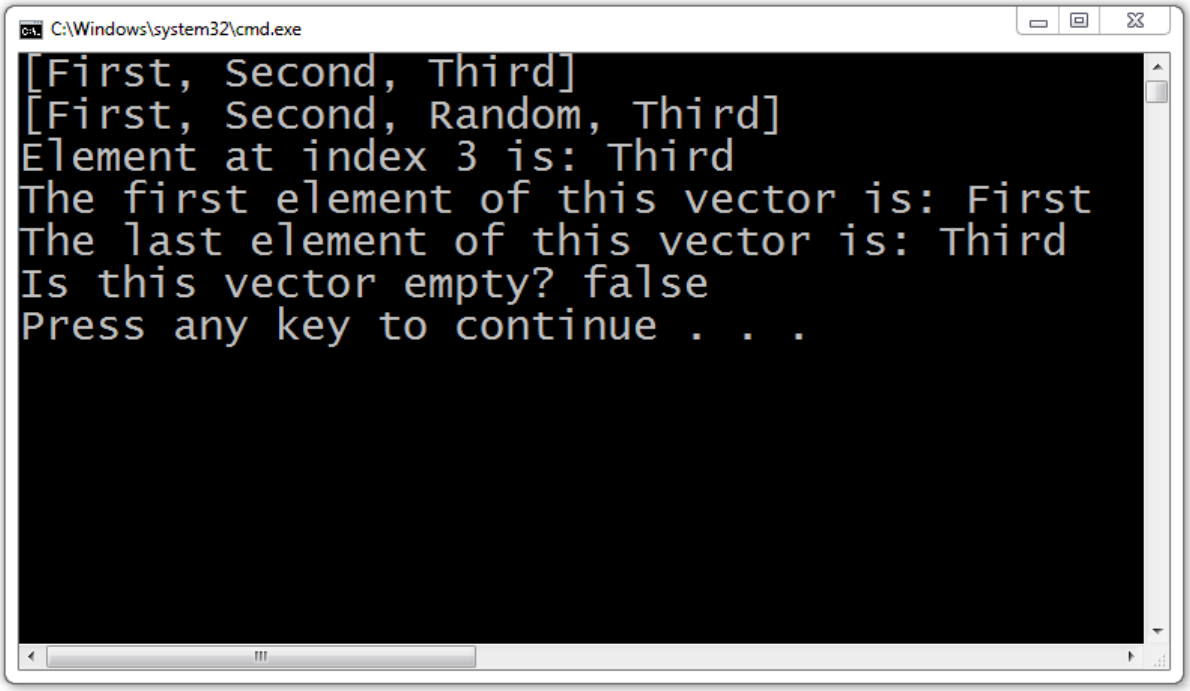

public class FileCopy {
public static void main(String[] args) {
try {
FileReader fr = new FileReader("input.txt");
BufferedReader br = new BufferedReader(fr);
FileWriter fw = new FileWriter("output.txt", true);
String s;
while ((s = br.readLine()) != null) { // read a line
fw.write(s); // write to output file
fw.flush();
}
br.close();
fw.close();
System.out.println("file copied");
} catch (IOException e) {
C:\Windows\system32\cmd.exe
file copied
Press any key to continue . . .
input - Notepad
File Edit Format View Help
input.txt contents being transfered
output - Notepad
File Edit Format View Help
input.txt contents being transfered input.txt contents being transfered

PROGRAM 11: Write a program to demonstrate the features of Vector class.

SOURCE CODE:

```
importjava.util.Vector;

public class VectorOperations {

public static void main(String a[]){
Vector<String>vct = new Vector<String>();

//adding elements to the end

vct.add("First");
vct.add("Second");
vct.add("Third");
System.out.println(vct);

//adding element at specified index

vct.add(2,"Random");
System.out.println(vct);

//getting elements by index

System.out.println("Element at index 3 is: "+vct.get(3));

//getting first element

System.out.println("The first element of this vector is: "+vct.firstElement());

//getting last element

System.out.println("The last element of this vector is: "+vct.lastElement());

//how to check vector is empty or not

System.out.println("Is this vector empty? "+vct.isEmpty());
}
}
```

OUTPUT:


C:\Windows\system32\cmd.exe
[First, Second, Third]
[First, Second, Random, Third]
Element at index 3 is: Third
The first element of this vector is: First
The last element of this vector is: Third
Is this vector empty? false
Press any key to continue . . .

PROGRAM 12: Develop java program for Client and Server setup where a client connects, sends messages to server and the server shows them using socket connection.

SOURCE CODE:

Server.java

```
// File Name Server.java
import java.net.*;
import java.io.*;

public class Server extends Thread
{
privateServerSocketserverSocket;

public Server(int port) throws IOException
  {
serverSocket = new ServerSocket(port);
serverSocket.setSoTimeout(10000);
  }

public void run()
  {
while(true)
    {
try
      {
System.out.println("Waiting for client on port " +
serverSocket.getLocalPort() + "...");
          Socket server = serverSocket.accept();

System.out.println("Just connected to " + server.getRemoteSocketAddress());
DataInputStream in = new DataInputStream(server.getInputStream());

System.out.println(in.readUTF());
DataOutputStream out = new DataOutputStream(server.getOutputStream());
out.writeUTF("Thank you for connecting to " + server.getLocalSocketAddress()
            + "\nGoodbye!");
server.close();

      }
```

```
catch (SocketTimeoutException s) {
System.out.println("Socket timed out!");
break;
        }
catch (IOException e)
        {
e.printStackTrace();
break;
        }
      }
   }

public static void main(String [] args) {
int port = Integer.parseInt(args[0]);
try {
        Thread t = new Server(port);
t.start();
      } catch (IOException e) {
e.printStackTrace();
      }
   }
}
```

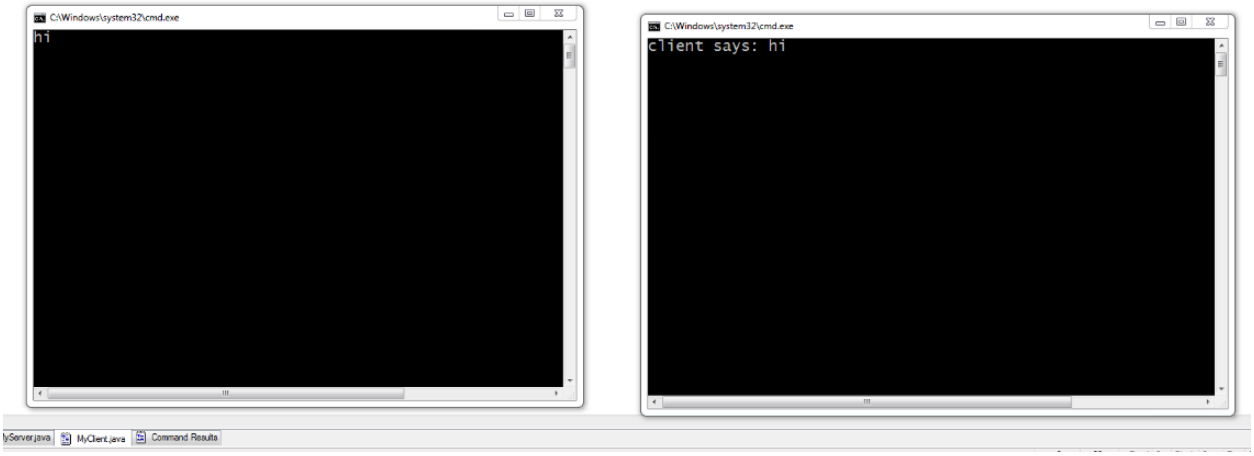
Client.java

```
import java.net.*;
import java.io.*;

public class Client
{
public static void main(String [] args)
   {
      String serverName = args[0];
int port = Integer.parseInt(args[1]);
try
      {
System.out.println("Connecting to " + serverName + " on port " + port);
        Socket client = new Socket(serverName, port);

System.out.println("Just connected to " + client.getRemoteSocketAddress());
```

```
OutputStreamoutToServer = client.getOutputStream();
DataOutputStream out = new DataOutputStream(outToServer);

out.writeUTF("Hello from " + client.getLocalSocketAddress());
InputStreaminFromServer = client.getInputStream();
DataInputStream in = new DataInputStream(inFromServer);

System.out.println("Server says " + in.readUTF());
client.close();
      }
catch (IOException e)
      {
e.printStackTrace();
      }
   }
}
```

OUTPUT:

```
ut.writeUTF(str);
ut.flush();
r2=din.readUTF();
stem.out.println("Server says: "+str2);

dout.close();
close();
```

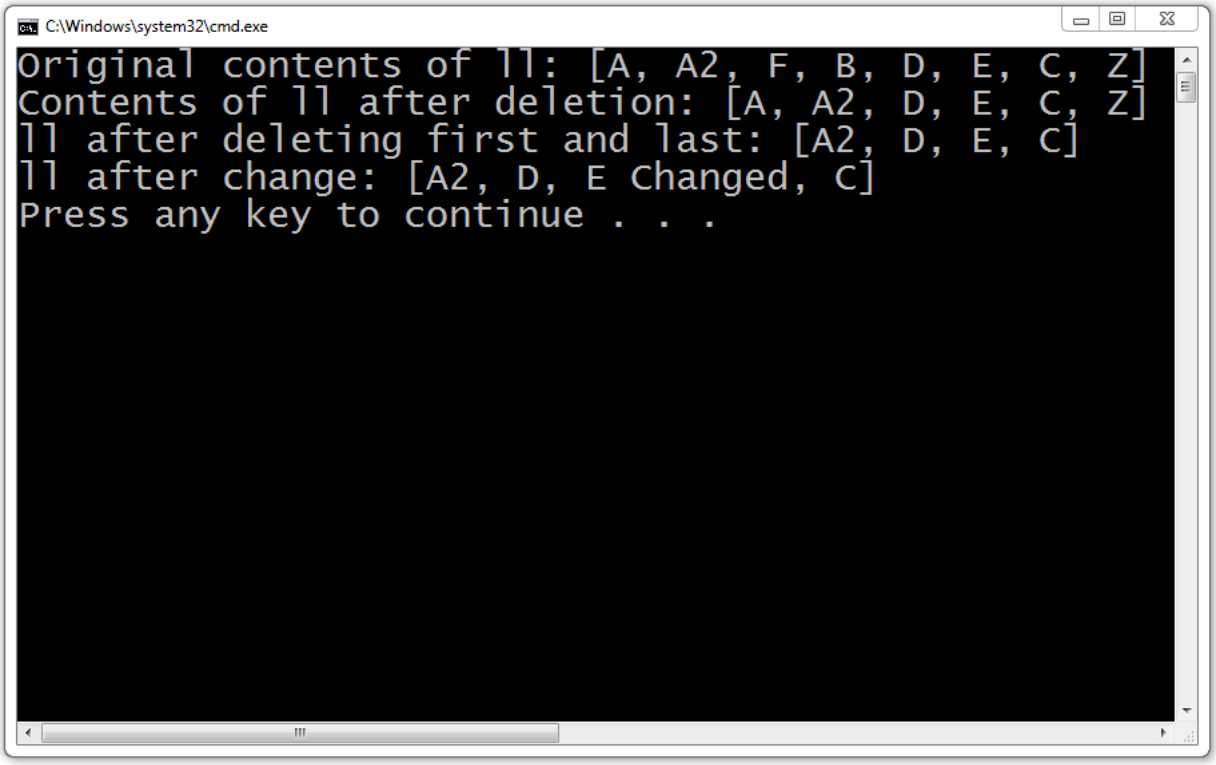


PROGRAM 13: Program based on collection framework.

SOURCE CODE:

```
importjava.util.*;
classLinkedListDemo {
public static void main(String args[]) {
// Create a linked list.
LinkedList<String>ll = new LinkedList<String>();
// Add elements to the linked list.
ll.add("F");
ll.add("B");
ll.add("D");
ll.add("E");
ll.add("C");
ll.addLast("Z");
ll.addFirst("A");
ll.add(1, "A2");
System.out.println("Original contents of ll: " + ll);
// Remove elements from the linked list.
ll.remove("F");
ll.remove(2);
System.out.println("Contents of ll after deletion: " + ll);
// Remove first and last elements.
ll.removeFirst();
ll.removeLast();
System.out.println("ll after deleting first and last: " + ll);
// Get and set a value.
String val = ll.get(2);
ll.set(2, val + " Changed");
System.out.println("ll after change: " + ll);
}
}
```

OUTPUT:


C:\Windows\system32\cmd.exe
Original contents of ll: [A, A2, F, B, D, E, C, Z]
Contents of ll after deletion: [A, A2, D, E, C, Z]
ll after deleting first and last: [A2, D, E, C]
ll after change: [A2, D, E Changed, C]
Press any key to continue . . .

PROGRAM 14: Write a java program to create a file menu with options New, Save and Close, Edit menu with option cut, copy and paste.

SOURCE CODE:

```
import java.io.*;

importjavax.swing.*;

importjava.awt.event.*;

importjava.util.*;

public class TextEdit extends JFrame implements ActionListener {

privateJTextAreatextArea = new JTextArea();

privateJMenufileMenu = new JMenu("File");

privateJMenuBarmenuBar = new JMenuBar();

privateJMenuItemnewItem = new JMenuItem("New");

privateJMenuItemsaveItem = new JMenuItem("Save");

privateJMenuItemexitItem = new JMenuItem("Close");

privateJMenueditMenu = new JMenu("Edit");

privateJMenuItem Cut = new JMenuItem("Cut");

privateJMenuItem Copy  = new JMenuItem("Copy");

privateJMenuItem Paste  = new JMenuItem("Paste");

private String filename = null;  // set by "Open" or "Save As"


public static void main(String args[]) {

newTextEdit();

  }


  // Constructor: create a text editor with a menu
```

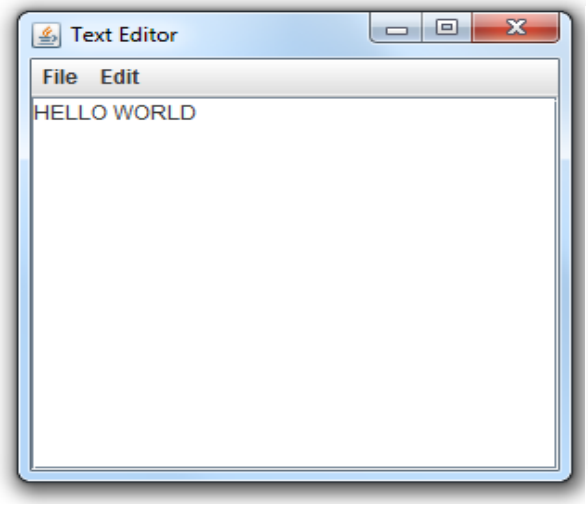
```
publicTextEdit() {
super("Text Editor");

    // Create menu and add listeners
menuBar.add(fileMenu);
setJMenuBar(menuBar);
fileMenu.add(newItem);
fileMenu.add(saveItem);
fileMenu.add(exitItem);
newItem.addActionListener(this);
saveItem.addActionListener(this);
exitItem.addActionListener(this);
menuBar.add(editMenu);
setJMenuBar(menuBar);
editMenu.add(Cut);
editMenu.add(Copy);
editMenu.add(Paste);
Cut.addActionListener(this);
Copy.addActionListener(this);
Paste.addActionListener(this);
menuBar.add(editMenu);
setJMenuBar(menuBar);

    // Create and display rest of GUI
add(new JScrollPane(textArea));
```

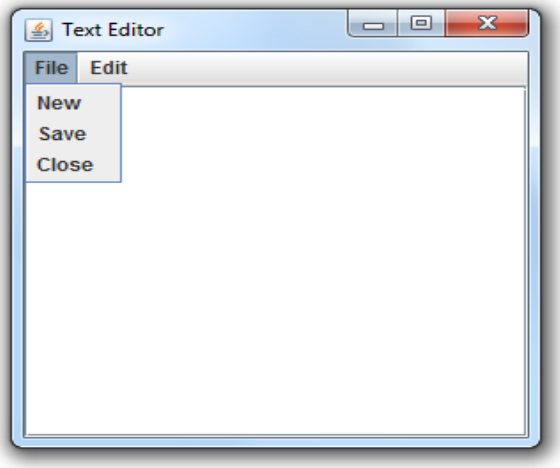
```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(300, 300);
setVisible(true);
  }

  // Handle menu events
public void actionPerformed(ActionEvent e) {
if (e.getSource() ==Cut)
textArea.cut();
else if (e.getSource() ==Copy)
textArea.copy();
else if (e.getSource() ==Paste)
textArea.paste();
else if (e.getSource() == newItem)
textArea.setText("");
else if (e.getSource() == saveItem)
saveFile(filename);
else if (e.getSource() == exitItem)
System.exit(0);
  }

  // Prompt user to enter filename and load file.  Allow user to cancel.
  // If file is not found, pop up an error and leave screen contents
  // and filename unchanged.
private void loadFile() {
```

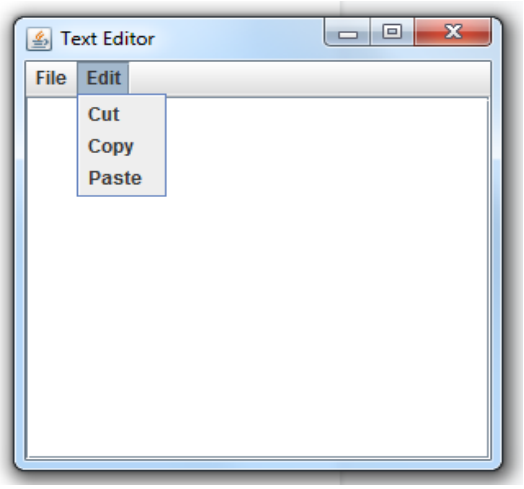
```
JFileChooser fc = new JFileChooser();
   String name = null;
if (fc.showOpenDialog(null) != JFileChooser.CANCEL_OPTION)
name = fc.getSelectedFile().getAbsolutePath();
else
return;  // user cancelled
try {
     Scanner in = new Scanner(new File(name));  // might fail
filename = name;
textArea.setText("");
while (in.hasNext())
textArea.append(in.nextLine() + "\n");
in.close();
   }
catch (FileNotFoundException e) {
JOptionPane.showMessageDialog(null, "File not found: " + name,
       "Error", JOptionPane.ERROR_MESSAGE);
   }
  }


  // Save named file.  If name is null, prompt user and assign to filename.
  // Allow user to cancel, leaving filename null.  Tell user if save is
  // successful.
private void saveFile(String name) {
if (name == null) {  // get filename from user
```

```
JFileChooser fc = new JFileChooser();
if (fc.showSaveDialog(null) != JFileChooser.CANCEL_OPTION)
name = fc.getSelectedFile().getAbsolutePath();
    }
if (name != null) {  // else user cancelled
try {
        Formatter out = new Formatter(new File(name));  // might fail
filename = name;
out.format("%s", textArea.getText());
out.close();
JOptionPane.showMessageDialog(null, "Saved to " + filename,
          "Save File", JOptionPane.PLAIN_MESSAGE);
      }
catch (FileNotFoundException e) {
JOptionPane.showMessageDialog(null, "Cannot write to file: " + name,
          "Error", JOptionPane.ERROR_MESSAGE);
      }
    }
  }   }
```

OUTPUT:


Text Editor
File
Edit
HELLO WORLD


Text Editor
File
Edit
New
Save
Close


Text Editor
File
Edit
Cut
Copy
Paste

PROGRAM 15: Write a java program to Design a calculator to perform only addition and division. It must contains three Buttons with labels +, / and =, and a TextFeild to get input and display the result.

SOURCE CODE:

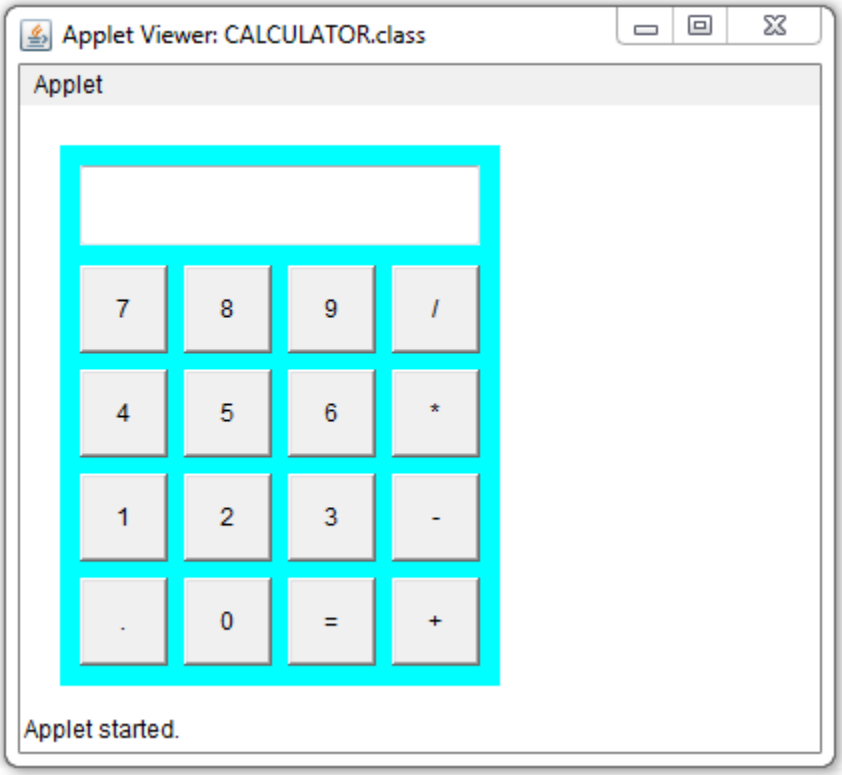
```
importjava.awt.*;
importjava.awt.event.*;
importjava.applet.*;

//<applet code = "CALCULATOR.class" width = 260 height = 310></applet>

public class CALCULATOR extends Applet implements ActionListener
{
TextField t1;
 Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b0;
 Button add,sub,mul,div, eql, dot;
 String msg="",tmp;
int a, b;
public void init()
 {
setLayout(null);
  t1=new TextField(20);
  b1=new Button("1");
  b2=new Button("2");
  b3=new Button("3");
  b4=new Button("4");
  b5=new Button("5");
  b6=new Button("6");
  b7=new Button("7");
  b8=new Button("8");
  b9=new Button("9");
  b0=new Button("0");
add=new Button("+");
sub=new Button("-");
div=new Button("/");
mul=new Button("*");
dot=new Button(".");
eql=new Button("=");

add(t1);
```

```
add(b7);
add(b8);
add(b9);
add(div);

add(b4);
add(b5);
add(b6);
add(mul);

add(b1);
add(b2);
add(b3);
add(sub);

add(dot);
add(b0);
add(eql);
add(add);

t1.setBounds(30,30,200,40);
b7.setBounds(30,80,44,44);
b8.setBounds(82,80,44,44);
b9.setBounds(134,80,44,44);
b4.setBounds(30,132,44,44);
b5.setBounds(82,132,44,44);
b6.setBounds(134,132,44,44);
b1.setBounds(30,184,44,44);
b2.setBounds(82,184,44,44);
b3.setBounds(134,184,44,44);
dot.setBounds(30,236,44,44);
b0.setBounds(82,236,44,44);
eql.setBounds(134,236,44,44);
add.setBounds(186,236,44,44);
sub.setBounds(186,184,44,44);
mul.setBounds(186,132,44,44);
div.setBounds(186,80,44,44);

b0.addActionListener(this);
b1.addActionListener(this);
```

```
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);
b6.addActionListener(this);
b7.addActionListener(this);
b8.addActionListener(this);
b9.addActionListener(this);
  //b0.addActionListener(this);
  //b0.addActionListener(this);
div.addActionListener(this);
mul.addActionListener(this);
add.addActionListener(this);
sub.addActionListener(this);
eql.addActionListener(this);
 }
public void actionPerformed(ActionEventae)
 {
  String str = ae.getActionCommand();
if (str.equals("+")||str.equals("-")||str.equals("*")||str.equals("/"))
  {
   String str1 = t1.getText();
tmp=str;
   a = Integer.parseInt(str1);
msg="";
  }
else if(str.equals("="))
  {
   String str2 = t1.getText();
   b = Integer.parseInt(str2);
int sum=0;
if(tmp=="+")
sum=a+b;
else if(tmp=="-")
sum=a-b;
else if(tmp=="*")
sum=a*b;
else if(tmp=="/")
sum=a/b;
   String str1=String.valueOf(sum);
```

```
t1.setText(""+str1);
msg="";
  }
else
  {
   //String ae.getActionCommand();
   //str += ae.getActionCommand();
msg+=str;
t1.setText(""+msg);
  }
 }
public void paint(Graphics g)
 {
g.setColor(Color.cyan);
g.fillRect(20,20,220,270);
 }
}
```

OUTPUT:


Applet Viewer: CALCULATOR.class
Applet
7
8
9
/
4
5
6
*
1
2
3
-
.
0
=
+
Applet started.