

VEMU INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

LAB MANUAL



15A05405- JAVA PROGRAMMING LABORATORY

Regulation – R15

Year / Semester: II / II

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B. Tech II - II sem (C.S.E)

L C
4 2

(15A05405) JAVA PROGRAMMING LABORATORY

Course Objectives:

- Learn to use object orientation to solve problems and use java language to implement them.
- To experiment with the syntax and semantics of java language and gain experience with java programming

Course Outcomes:

- Ability to write portable programs which work in all environments
- Ability to create user friendly interfaces
- Ability to solve the problem using object oriented approach and design solutions which are robust

1) Write Java program(s) on use of inheritance, preventing inheritance using final, abstract classes.

Inheritance:

```
class Parent
{
public void p1()
    {
System.out.println("Parent method");
    }
}
public class Child extends Parent {
public void c1()
    {
System.out.println("Child method");
    }
public static void main(String[] args)
    {
        Child cobj = new Child();
cobj.c1();    //method of Child class
cobj.p1();    //method of Parent class
    }
}
```

Output :

Child method

Parent methodParent method

preventing inheritance using final:

```
final class B{}
```

```
class H extends B{
```

```
void run(){System.out.println("running safely with 100kmph");}
```

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

```
    H h= new H();
```

```
h.run();
```

```
    }
```

```
}
```

Output:Compile Time Error

Abstract Classes:

```
abstract class A
```

```
{
```

```
    abstract void callme();
```

```
}
```

```
class B extends A
```

```
{
```

```
void callme()
```

```
{
```

```
System.out.println("this is callme.");
```

```
}
```

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

```
{
```

```
    B b = new B();
```

```
b.callme();
```

```
}
```

```
}
```

Output :

this is callme

2) Write Java program(s) on dynamic binding, differentiating method overloading and overriding.

Dynamic Binding:

```
class A{
void samp(){System.out.println("hai...");}
}

class D extends A{
void samp(){System.out.println("hello...");}

public static void main(String args[]){
    A a=new D();
a.samp();
}
}
```

Output:

Hello

Overload

```
class Overload
{
void demo (int a)
{
System.out.println ("a: " + a);
}
void demo (int a, int b)
{
```

```
System.out.println ("a and b: " + a + "," + b);  
    }
```

```
double demo(double a) {  
System.out.println("double a: " + a);  
return a*a;  
    }
```

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

```
    {  
        Overload Obj = new Overload();  
  
double result;  
Obj .demo(10);  
Obj .demo(10, 20);  
result = Obj .demo(5.5);  
System.out.println("O/P : " + result);  
    }
```

```
    }
```

Overriding:

```
public class BaseClass  
{  
public void methodToOverride() //Base class method  
{  
System.out.println ("I'm the method of BaseClass");
```

```

    }
}
public class DerivedClass extends BaseClass
{
public void methodToOverride() //Derived Class method
    {
System.out.println ("I'm the method of DerivedClass");
    }
}

```

```

public class TestMethod
{
public static void main (String args []) {
    // BaseClass reference and object
BaseClass obj1 = new BaseClass();
    // BaseClass reference but DerivedClass object
BaseClass obj2 = new DerivedClass();
    // Calls the method from BaseClass class
obj1.methodToOverride();
    //Calls the method from DerivedClass class
obj2.methodToOverride();
    }
}

```

4)Write Java program(s) on ways of implementing interface.

```
interface MyInterface
{
    public void method1();
    public void method2();
}

class XYZ implements MyInterface
{
    public void method1()
    {
        System.out.println("implementation of method1");
    }

    public void method2()
    {
        System.out.println("implementation of method2");
    }

    public static void main(String arg[])
    {
        MyInterface obj = new XYZ();
        obj.method1();
    }
}
```

5) Write a program for the following

- **Develop an applet that displays a simple message.**
- **Develop an applet for waving a Flag using Applets and Threads**

Develop an applet that displays a simple message.

```

import java.awt.*;
import java.applet.*;
/*
<applet code="sim" width=300 height=300>
</applet>
*/
public class sim extends Applet
{
    String msg=" ";
    public void init()
    {
        msg+="init()--->";
        setBackground(Color.orange);
    }
    public void start()
    {
        msg+="start()--->";
        setForeground(Color.blue);
    }
    public void paint(Graphics g)
    {
        msg+="paint()--->";
        g.drawString(msg,200,50);
    }
}

```

Develop an applet for waving a Flag using Applets and Threads

```

//<applet code="Flag1.class" height=300 width=300></applet>
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class Flag1 extends Applet implements ActionListener {
    Button but = new Button ("click");
    double add = 0;
    public void init () {
        setBackground (Color.white);
        add (but);
        but.addActionListener (this);
    }
    public void paint (Graphics g) {
        Dimension d = getSize();
        int height = d.height, width = d.width;
        int x[ ] = new int[2*width], y[ ] = new int[2*width];
        for (int color=0; color<3; color++) {
            switch(color) {

```



```

case 0: g.setColor (Color.orange); break;
case 1: g.setColor (Color.white); break;
case 2: g.setColor (Color.green);
}
// The following 2 lines are actually one:
int size = fillArrays
(x, y, width/8, 3*width/4, (1+color)*height/5, height/5, 5, add);
g.fillPolygon (x, y, size);
}
}
// The following 2 lines are actually one:
intfillArrays
(int h[ ], int v[ ], intxStart, intwd, intyStart, intht, int step, double extra) {
int count = 0;
// top line of the colour band:
for (inti=xStart; (i<=xStart+wd); i += step) {
h[count] = i;
v[count++] = yStart+(int)(ht*Math.sin(i/(0.2*wd)+extra));
}
int n = count -1;
// The values of the bottom line are not calculated.
// They're equal to to the values of the top line + ht
for (inti=h[count-1]; i>=xStart; i-=step) {
h[count] = i;
v[count++] = v[n--]+ht;
}
return count;
}
public void actionPerformed (ActionEventvt) {
add += 0.7;
repaint();
}
}
}

```

6)Write Java program(s) which uses the exception handling features of the language, creates exceptions and handles them properly, uses the predefined exceptions, and create own exceptions

Example of User defined exception in Java

```

classMyException extends Exception{

String str1;

MyException(String str2) {

str1=str2;

```

```
    }  
    public String toString(){  
    return ("Output String = "+str1) ;  
    }  
}
```

```
classCustomException{  
    public static void main(String args[]){  
    try{  
    throw new MyException("Custom");  
        // I'm throwing user defined custom exception above  
    }  
    catch(MyExceptionexp){  
    System.out.println("Hi this is my catch block") ;  
    System.out.println(exp) ;  
    }  
    }  
}
```

uses the predefined exceptions

```
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");
    }
}

```

7)Write java program that inputs 5 numbers, each between 10 and 100 inclusive. As each number is read display it only if it's not a duplicate of any number already read. Display the complete set of unique values input after the user enters each new value.

```

import java.lang.*;
import java.io.*;
import java.util.*;
class Number
{
public static void main(String args[])throws IOException
{
Scanner input =new Scanner(System.in);
int sid[]=new int[5];
int count=0;
int x=0;
int num=0;
while(x<sid.length)
{
System.out.println("enter number");
num=input.nextInt();
if((num>=10)&&(num<=100))
{
boolean digit=false;
x++;
for(int i=0;i<sid.length;i++)
{
if(sid[i]==num)
digit=true;
}
if(!digit)
{

```

```

sid[count]=num;
count++;
}
else
System.out.println("number was entered before\n");
}
else
{
System.out.println("the number must be between 10 &100");
}
for(inti=0;i<x;i++)
{
System.out.println(sid[i]+" ");
}
System.out.println();
}
}
}

```

8)Write Java program(s) on creating multiple threads, assigning priority to threads, synchronizing threads, suspend and resume threads

```

classNewThread implements Runnable
{
    String name;          //name of thread
    Thread thr;
    booleansuspendFlag;

    NewThread(String threadname)
    {
        name = threadname;
        thr = new Thread(this, name);
        System.out.println("New thread : " + thr);
        suspendFlag = false;
        thr.start();      // start the thread
    }

    /* this is the entry point for thread */
    public void run()
    {
        try
        {
            for(inti=12; i>0; i--)
            {
                System.out.println(name + " : " + i);
                Thread.sleep(200);
                synchronized(this)
                {

```

```

while(suspendFlag)
    {
wait();
    }
    }
}
catch(InterruptedException e)
{
System.out.println(name + " interrupted");
}

System.out.println(name + " exiting...");
}

synchronized void mysuspend()
{
suspendFlag = true;
}

synchronized void myresume()
{
suspendFlag = false;
notify();
}
}

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

NewThread obj1 = new NewThread("One");
NewThread obj2 = new NewThread("two");

try
{
Thread.sleep(1000);
obj1.mysuspend();
System.out.println("Suspending thread One...");
Thread.sleep(1000);
obj1.myresume();
System.out.println("Resuming thread One...");

obj2.mysuspend();
System.out.println("Suspending thread Two...");
Thread.sleep(1000);
obj2.myresume();
System.out.println("Resuming thread Two...");
}
catch(InterruptedException e)
{
System.out.println("Main thread Interrupted..!!");
}
}

```

```

        /* wait for threads to finish */
try
    {
System.out.println("Waiting for threads to finish...");
obj1.thr.join();
obj2.thr.join();
    }
catch (InterruptedException e)
    {
System.out.println("Main thread Interrupted..!!");
    }

System.out.println("Main thread exiting...");

    }
}

```

10) Write a java program to split a given text file into n parts. Name each part as the name of the original file followed by .part<n> where n is the sequence number of the part file

```

import java.io.*;

public class SplitFile1 {

    private static String FILE_NAME = "f:/TextFile.txt";

    private static byte PART_SIZE = 5;

    public static void main(String[] args) {

        File inputFile = new File(FILE_NAME);

        FileInputStream inputStream;

        String newFileName;

        FileOutputStream filePart;

        int fileSize = (int) inputFile.length();

        int nChunks = 0, read = 0, readLength = PART_SIZE;

        byte[] byteChunkPart;

        try {

            inputStream = new FileInputStream(inputFile);

```

```

while (fileSize > 0) {
    if (fileSize <= 5) {
        readLength = fileSize;
        }
    byteChunkPart = new byte[readLength];
    read = inputStream.read(byteChunkPart, 0, readLength);
    fileSize -= read;
    assert (read == byteChunkPart.length);
    nChunks++;
    newFileName = FILE_NAME + ".part"
                + Integer.toString(nChunks - 1);
    filePart = new FileOutputStream(new File(newFileName));
    filePart.write(byteChunkPart);
    filePart.flush();
    filePart.close();
    byteChunkPart = null;
    filePart = null;
    }
    inputStream.close();
    } catch (IOException exception) {
        exception.printStackTrace();
    }
}

```

11. Write a java program to create a super class called Figure that receives the dimensions of two dimensional objects. It also defines a method called area that computes the area of an object. The program derives two subclasses from Figure. The first is Rectangle and second is Triangle. Each of the sub classes override area() so that it returns the area of a rectangle and triangle respectively.

```
import java.lang.*;
```

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

```
abstract class Figure
```

```
{
```

```
    int dim1,dim2;
```

```
    void getd()throws IOException
```

```
    {
```

```
        BufferedReader br = new BufferedReader (new InputStreamReader(System.in));
```

```
        System.out.println ("Enter Value of 1st Dimension");
```

```
        dim1=Integer.parseInt(br.readLine());
```

```
        System.out.println ("Enter Value of 2nd Dimension");
```

```
        dim2=Integer.parseInt(br.readLine());
```

```
    }
```

```
    abstract void area();
```

```
}
```

```
class Rectangle extends Figure
```

```
{
```

```
    void getd() throws IOException
```

```
    {
```

```
        super.getd();
```

```
    }
```

```
    void area()
```



```

    {
        int a=dim1*dim2;
        System.out.println ("Area of Rectangle = "+a);
    }
}
class Triangle extends Figure
{
    void getd() throws IOException
    {
        super.getd();
    }
    void area()
    {
        double b=(1*dim1*dim2)/2;
        System.out.println ("Area of Triangle = "+b);
    }
}
class Methodover
{
    public static void main(String args[]) throws IOException
    {
        Rectangle R = new Rectangle();
        R.getd();
        R.area();
        Triangle T = new Triangle();
    }
}

```

```

        T.getd();

        T.area();

    }

}

```

12) Write a Java program that creates three threads. First thread displays “Good Morning” every one second, the second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds

```

class Frst implements Runnable
{
    Thread t;
    Frst()
    {
        t=new Thread(this);
        System.out.println("Good Morning");
        t.start();
    }
    public void run()
    {
        for(int i=0;i<10;i++)
        {
            System.out.println("Good Morning"+i);
            try{
                t.sleep(1000);
            }
            catch(Exception e)
            {
                System.out.println(e);
            }
        }
    }
}

class sec implements Runnable
{
    Thread t;
    sec()
    {
        t=new Thread(this);

```

```

System.out.println("hello");
t.start();
}
public void run()
{
for(inti=0;i<10;i++)
{
System.out.println("hello"+i);
try{
t.sleep(2000);
}
catch(Exception e)
{
System.out.println(e);
}
}
}
}
class third implements Runnable
{
Thread t;
third()
{
t=new Thread(this);
System.out.println("welcome");
t.start();
}
public void run()
{
for(inti=0;i<10;i++)
{
System.out.println("welcome"+i);
try{
t.sleep(3000);
}
catch(Exception e)
{
System.out.println(e);
}
}
}
}
public class Multithread
{
public static void main(String arg[])
{

```

```
newFrst();
new sec();
new third();
}
}
```

14) Write a java program to handle mouse events

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*
<applet code="Mouse" width=500 height=500>
</applet>
*/
public class Mouse extends Applet
implements MouseListener, MouseMotionListener
{
    int X=0, Y=20;
    String msg="MouseEvents";
    public void init()
    {
        addMouseListener(this);
        addMouseMotionListener(this);
        setBackground(Color.black);
        setForeground(Color.red);
    }
    public void mouseEntered(MouseEvent m)
    {
        setBackground(Color.magenta);
        showStatus("Mouse Entered");
        repaint();
    }
    public void mouseExited(MouseEvent m)
    {
        setBackground(Color.black);
        showStatus("Mouse Exited");
        repaint();
    }
    public void mousePressed(MouseEvent m)
    {
        X=10;
        Y=20;
        msg="NEC";
        setBackground(Color.green);
        repaint();
    }
}
```

```

    }
    public void mouseReleased(MouseEvent m)
    {
        X=10;
        Y=20;
        msg="Engineering";
        setBackground(Color.blue);
        repaint();
    }
    public void mouseMoved(MouseEvent m)
    {
        X=m.getX();
        Y=m.getY();
        msg="College";
        setBackground(Color.white);
        showStatus("Mouse Moved");
        repaint();
    }
    public void mouseDragged(MouseEvent m)
    {
        msg="CSE";
        setBackground(Color.yellow);
        showStatus("Mouse Moved"+m.getX()+" "+m.getY());
        repaint();
    }
    public void mouseClicked(MouseEvent m)
    {
        msg="Students";
        setBackground(Color.pink);
        showStatus("Mouse Clicked");
        repaint();
    }
    public void paint(Graphics g)
    {
        g.drawString(msg,X,Y);
    }
}

```

15) Write a java program to handle keyboard events

```

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*
<applet code="Key" width=300 height=400>
</applet>
*/

```

```

public class Key extends Applet
implementsKeyListener
{
    int X=20,Y=30;
    String msg="KeyEvents--->";
    public void init()
    {
        addKeyListener(this);
        requestFocus();
        setBackground(Color.green);
        setForeground(Color.blue);
    }
    public void keyPressed(KeyEvent k)
    {
        showStatus("KeyDown");
        int key=k.getKeyCode();
        switch(key)
        {
            caseKeyEvent.VK_UP:
                showStatus("Move to Up");
                break;
            caseKeyEvent.VK_DOWN:
                showStatus("Move to Down");
                break;
            caseKeyEvent.VK_LEFT:
                showStatus("Move to Left");
                break;
            caseKeyEvent.VK_RIGHT:
                showStatus("Move to Right");
                break;
        }
        repaint();
    }
    public void keyReleased(KeyEvent k)
    {
        showStatus("Key Up");
    }
    public void keyTyped(KeyEvent k)
    {
        msg+=k.getKeyChar();
        repaint();
    }
    public void paint(Graphics g)
    {
        g.drawString(msg,X,Y);
    }
}

```

16) Write a java program that allows conduction of object type examination containing multiple choice questions, and true/false questions. At the end of the examination when the user clicks a button the total marks have to be displayed in the form of the message.

17) Write a java program that creates menu which appears similar to the menu of notepad application of the Microsoft windows or any editor of your choice.

```
import javax.swing.*;
import java.awt.event.*;

public class Notepad implements ActionListener{
    JFrame f;
    JMenuBar mb;
    JMenu file, edit, help;
    JMenuItem cut, copy, paste, selectAll;
    JTextArea ta;

    Notepad(){
        f=new JFrame();

        cut=new JMenuItem("cut");
        copy=new JMenuItem("copy");
        paste=new JMenuItem("paste");
        selectAll=new JMenuItem("selectAll");

        cut.addActionListener(this);
```

```
copy.addActionListener(this);
paste.addActionListener(this);
selectAll.addActionListener(this);

mb=new JMenuBar();
mb.setBounds(5,5,400,40);

file=new JMenu("File");
edit=new JMenu("Edit");
help=new JMenu("Help");

edit.add(cut);edit.add(copy);edit.add(paste);edit.add(selectAll);

mb.add(file);mb.add(edit);mb.add(help);

ta=new JTextArea();
ta.setBounds(5,30,460,460);

f.add(mb);f.add(ta);

f.setLayout(null);
f.setSize(500,500);
f.setVisible(true);
}
```



```

public void actionPerformed(ActionEvent e) {
    if(e.getSource()==cut)
        ta.cut();
    if(e.getSource()==paste)
        ta.paste();
    if(e.getSource()==copy)
        ta.copy();
    if(e.getSource()==selectAll)
        ta.selectAll();
}

```

```

public static void main(String[] args) {
    new Notepad();
}
}

```

18) Write a java program that creates dialog box which is similar to the save dialog box of the Microsoft windows or any word processor of your choice

```

import javax.swing.*;
import java.awt.event.*;

public class ShowDialogBox{
    JFrame frame;

    public static void main(String[] args){
        ShowDialogBoxdb = new ShowDialogBox();
    }
}

```

```

public ShowDialogBox(){
    JFrame frame = new JFrame("Show Message Dialog");
    JButton button = new JButton("Click Me");
    button.addActionListener(new MyAction());
    frame.add(button);
    frame.setSize(400, 400);
    frame.setVisible(true);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

```

```

public class MyAction implements ActionListener{
    public void actionPerformed(ActionEvent e){
        JOptionPane.showMessageDialog(frame, "Roseindia.net");
    }
}

```

}19) Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.

```

class Q
{
    int n;
    boolean valueSet=false;
    synchronized int get()
    {
        if(!valueSet)

```

```

try
{
    wait();
}
catch(InterruptedException e)
{
    System.out.println("InterruptedException caught");
}
System.out.println("Got:"+n);
valueSet=false;
notify();
return n;
}
synchronized void put(int n)
{
    if(valueSet)
    try
    {
        wait();
    }
    catch(InterruptedException e)
    {
        System.out.println("InterruptedException caught");
    }
    this.n=n;
    valueSet=true;
    System.out.println("Put:"+n);
}

```

```

        notify();
    }
}

class Producer implements Runnable
{
    Q q;

    Producer(Q q)
    {
        this.q=q;

        new Thread(this,"Producer").start();
    }

    public void run()
    {
        inti=0;

        while(true)
        {
            q.put(i++);
        }
    }
}

class Consumer implements Runnable
{
    Q q;

    Consumer(Q q)
    {
        this.q=q;

        new Thread(this,"Consumer").start();
    }
}

```

```
    }  
    public void run()  
    {  
        while(true)  
        {  
            q.get();  
        }  
    }  
}  
class ProdCons  
{  
    public static void main(String[] args)  
    {  
        Q q=new Q();  
        new Producer(q);  
        new Consumer(q);  
        System.out.println("Press Control-c to stop");  
    }  
}
```

Output:

Put:1

Got:1

Put:2

Got:2

Put:3

Got:3

Put:4

Got:4

Put:5

Got:5

VEMUIT

Put:6
Got:6
Put:7
Got:7
Put:8
Got:8
Put:9
Got:9
Put:10
Got:10
Put:11
Got:11
Put:12
Got:12
Put:13
Got:13
Put:14
Got:14

20) Write a java program to find and replace pattern in a given file.

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

```
public class BTest
```

```
{
```

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

```
{
```

```
try
```

```
{
```

```
File file = new File("file.txt");
```

```
BufferedReader reader = new BufferedReader(new FileReader(file));
```

```
String line = "", oldtext = "";
```

```
while((line = reader.readLine()) != null)
```

```
{
```

```
oldtext += line + "\r\n";
```

```
}
```

```
reader.close();
```

```
// replace a word in a file
```

```
//String newtext = oldtext.replaceAll("drink", "Love");
```

```
//To replace a line in a file
```

```
String newtext = oldtext.replaceAll("This is test string 20000", "blah blah blah");
```

```
FileWriter writer = new FileWriter("file.txt");
```

```
writer.write(newtext);writer.close();
```

```

    }
    catch (IOException ioe)
    {
        ioe.printStackTrace();
    }
}
}
}

```

file.txt

I drink Java

I sleep Java

This is test string 1

This is test string 20000

21) Use inheritance to create an exception super class called ExceptionA and exception sub classes ExceptionB and ExceptionC, where ExceptionB inherits from ExceptionA and ExceptionC inherits from ExceptionB. Write a java program to demonstrate that the catch block for type ExceptionA catches exception of type ExceptionB and ExceptionC.

```

import java.util.*;
import java.io.*;
import java.lang.*;
class ExceptionA extends java.lang.Exception {
}

class ExceptionB extends ExceptionA {
}
class ExceptionC extends ExceptionB {

}

public class ExceptionTest{
    public static void main(String[] args) throws Exception {
        System.out.println("throwing ExceptionA, catching ExceptionB");
        try {
            throw new ExceptionA();
        }
    }
}

```



```

    }
    catch (ExceptionB e) {
    System.out.println("caught! " + e);
    }
    System.out.println("throwing ExceptionA, catching ExceptionC");
    try {
    throw new ExceptionA();
    }
    catch (ExceptionC e) {
    System.out.println("caught! " + e);
    }

    System.out.println("bye");
    }
}

```

Input & Output:

Throwing Exception A, Catching Exception B

Exception in thread "main" Exception A at Exception Test main (ExceptionTest.java)

23. Write a Java program to create aURLConnection and use it to examine the documents properties and content.

```

import java.net.*;

import java.io.*;

import java.util.Date;

classUCDemo

{

public static void main(String args[]) throws Exception {

int c;

URL hp = new URL("http://www.java-samples.com/j2me/");

URLConnectionhpCon = hp.openConnection();

System.out.println("Date: " + new Date(hpCon.getDate()));

System.out.println("Content-Type: " +

```

```

hpCon.getContentType();

System.out.println("Expires: " + hpCon.getExpiration());

System.out.println("Last-Modified: " +
new Date(hpCon.getLastModified()));

intlen = hpCon.getContentLength();

System.out.println("Content-Length: " + len);

if (len> 0) {

System.out.println("=== Content ===");

InputStream input = hpCon.getInputStream();

inti = len;

while (((c = input.read()) != -1) && (--i> 0)) {

System.out.print((char) c);

}

input.close();

} else {

System.out.println("No Content Available");

}

}

}

start
Date: Thu Jan 01 01:00:00 GMT 1970
Content-Type: null
Expires: 0
Last-Modified: Thu Jan 01 01:00:00 GMT 1970
length-1
Content not available

```

24. Write a Java program which uses TCP/IP and Datagrams to communicate client and server.

MyServer.java

VEMUIT

```

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

public class MyServer {

public static void main(String[] args){

try{

ServerSocketss=new ServerSocket(6666);

Socket s=ss.accept();//establishes connection

DataInputStream dis=new DataInputStream(s.getInputStream());

String str=(String)dis.readUTF();

System.out.println("message= "+str);

ss.close();

}catch(Exception e){System.out.println(e);}

}

}

```

File: MyClient.java

```

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

public class MyClient {

public static void main(String[] args) {

try{

Socket s=new Socket("localhost",6666);

DataOutputStreamdout=new DataOutputStream(s.getOutputStream());

dout.writeUTF("Hello Server");

}

}

```

```

dout.flush();
dout.close();
s.close();
}catch(Exception e){System.out.println(e);}
}
}

```

Output:

25.Create an interface for stack with push and pop operations. Implement the stack in two ways: fixed size stack and Dynamic stack (stack size is increased when stack is full).

```

interface stackop
{
    void push(int item);
    int pop();
}
class FixedStack implements stackop
{
    private intstk[ ];
    private inttos;
    FixedStack(int size)
    {
        stk=new int[size];
        tos=-1;
    }
    public void push(int item)
    {
        if(tos==stk.length-1)
        {
            System.out.println("Stack Overflows");

```

```

        int t[ ]=new int[stk.length * 2];
        for(int i=0;i<stk.length;i++)
            t[i]=stk[i];
        stk=t;
        stk[++tos]=item;
    }
    else
        stk[++tos]=item;
}
public int pop()
{
    if(tos<0)
    {
        System.out.println("Stack Underflows");
        return 0;
    }
    else
        return stk[tos--];
}
}
class DynStack implements stackop
{
    private int stk[ ];
    private int tos;
    DynStack(int size)
    {
        stk=new int[size];
        tos=-1;
    }
    public void push(int item)
    {
        if(tos==stk.length-1)
        {

```

```

        System.out.println("Stack Overflows.");
        int t[ ]=new int[stk.length * 2];
        for(int i=0;i<stk.length;i++)
            t[i]=stk[i];
        stk=t;
        stk[++tos]=item;
    }
    else
        stk[++tos]=item;
}
public int pop()
{
    if(tos<0)
    {
        System.out.println("Stack Underflows.");
        return 0;
    }
    else
        return stk[tos--];
}
}
class StackTest
{
    public static void main(String args[ ])
    {
        FixedStack fs=new FixedStack(3);
        DynStack ds=new DynStack(5);
        stackop mystk;
        for(int i=0;i<3;i++)
            fs.push(i);
        System.out.println("Fixed length Stack Contents are.");
        for(int i=0;i<3;i++)
            System.out.println(fs.pop());
    }
}

```

```

    for(int i=0;i<7;i++)
        ds.push(i);
    System.out.println("Dynamic length Stack Contents are");
    for(int i=0;i<7;i++)
        System.out.println(ds.pop());
    mystk=fs;
    for(int i=0;i<3;i++)
        mystk.push(i);
    System.out.println("Fixed length Stack Contents using interface
reference");
    for(int i=0;i<3;i++)
        System.out.println(mystk.pop());
    mystk=ds;
    for(int i=0;i<7;i++)
        mystk.push(i);
    System.out.println("Dynamic length Stack Contents using interface
reference");
    for(int i=0;i<7;i++)
        System.out.println(mystk.pop());
    }
}

```

Output:

```

C:\Windows\system32\cmd.exe
Fixed length Stack Contents are.
2
1
0
Stack Overflows.
Dynamic length Stack Contents are
6
5
4
3
2
1
0
Fixed length Stack Contents using interface reference
2
1
0
Dynamic length Stack Contents using interface reference
6
5
4
3
2
1
0
Press any key to continue . . .

```

26. Create multiple threads to access the contents of a stack. Synchronize thread to prevent simultaneous access to push and pop operations.

```

class StackImpl { // (1)
private Object[] stackArray;
private int topOfStack;

public StackImpl(int capacity) {
stackArray = new Object[capacity];
topOfStack = -1;
}

public boolean push(Object element) { // (2a) non-synchronized
// public synchronized boolean push(Object element) { // (2b) synchronized

```



```

if (isFull()) return false;

    ++topOfStack;

try { Thread.sleep(1000); } catch (Exception e) { } // (3) Sleep a little.

stackArray[topOfStack] = element;

return true;

}

public Object pop() {                                // (4a) non-synchronized
//public synchronized Object pop() {                // (4b) synchronized
if (isEmpty()) return null;

    Object obj = stackArray[topOfStack];

stackArray[topOfStack] = null;

try { Thread.sleep(1000); } catch (Exception e) { } // (5) Sleep a little.

topOfStack--;

return obj;

}

public boolean isEmpty() { return topOfStack < 0; }

public boolean isFull() { return topOfStack >= stackArray.length - 1; }

}

public class Mutex {

public static void main(String[] args) {

final StackImpl stack = new StackImpl(20); // (6) Shared by the threads.

    (new Thread("Pusher")) {                        // (7) Thread no. 1

```

```

public void run() {
    for(;;) {
        System.out.println("Pushed: " + stack.push(2008));
    }
}

}).start();

(new Thread("Popper") {           // (8) Thread no. 2
    public void run() {
        for(;;) {
            System.out.println("Popped: " + stack.pop());
        }
    }
}).start();

System.out.println("Exit from main().");
}
}

```

Possible output from the program when run with (2a) and (4a):

Exit from main().

...

Pushed: true

Popped: 2008

Popped: 2008

Popped: null

...

Popped: null

java.lang.ArrayIndexOutOfBoundsException: -1

at StackImpl.push(Mutex.java:15)

at Mutex\$1.run(Mutex.java:41)

Popped: null

Popped: null

...