**Lab Record on**

**Object Oriented Programming with JAVA Lab**

Submitted in partial fulfillment of the requirements for the

Object Oriented Programming with JAVA LAB in

I Semester

MASTER OF COMPUTER APPLICATIONS

Bangalore North University

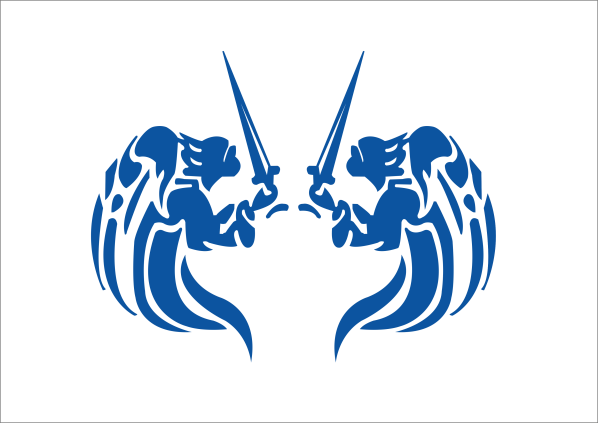
Submitted by

**(STUDENT NAME)**

**(REGISTER NO)**

Lab Guide

**Dr. Bobby Lukose**



**Department of MCA**

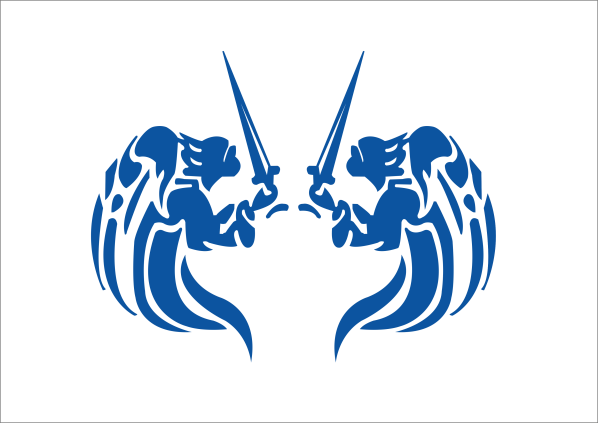
**Krupanidhi Group of Institutions**

**Affiliated To Bangalore North University**

ChikkaBellandur, Carmelaram Post, VarthurHobli

Bangalore-560035

**Krupanidhi Group of Institutions**



**certificate**

This is to certify that the report entitled **“Object Oriented Programming with JAVA LAB** “ embodies the original work by **STUDENT NAME (RESISTER NO)** in the partial fulfilment of the requirements for **Object Oriented Programming with JAVA LAB** for **MCA, I Semester** course during the academic semester from March 2021 to June 2021 as prescribed by Bangalore North University.

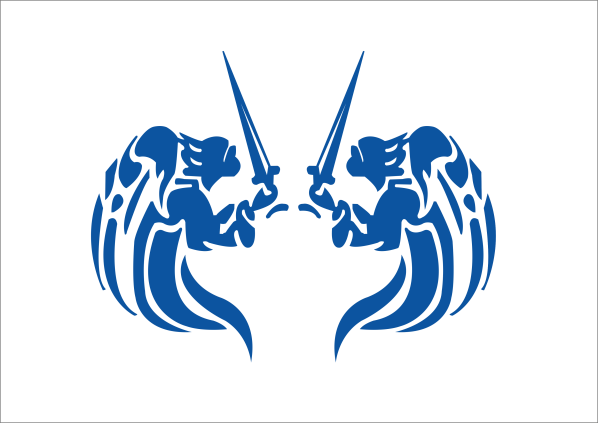
**Lab Guide Head of the Department**

**Dr. Bobby Lukose**

**Place:**

**Date:**

**Krupanidhi Group of Institutions**



**certificate**

This is to certify that the report entitled “**Object Oriented Programming with JAVA LAB**” embodies the original work by **STUDENT NAME (REGISTER NO)** in the partial fulfilment of the requirements for **Object Oriented Programming with JAVA LAB** for **MCA, I Semester** course during the academic semester from March 2021 to June 2021 as prescribed by Bangalore North University.

**External Examiner**

**1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**2.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Place:**

**Date:**

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Date** | **Program Title** | **Page No** |
| 1 | 04-01-2022 | Develop a JAVA program to demonstrate the precedence and associativity among arithmetic operators. The program should also demonstrate how the default precedence can be overridden. | 6 |
| 2 | 08-01-2022 | Write a JAVA program to validate a date. The program should accept day, month and year and it should report whether they form a valid date or not. | 7 |
| 3 | 11-01-2022 | Write a JAVA program to display the following pattern.  1  2 2  3 3 3  4 4 4 4  5 5 5 5 5 | 8 |
| 4 | 13-01-2022 | Write a JAVA program to print the first n members of Fibonacci series. | 9 |
| 5 | 18-01-2022 | Write a program to generate the multiplication tables of a range of numbers between m and n inclusive and m < n. | 10 |
| 6 | 20-01-2022 | Write a JAVA program to define a class, define instance methods for setting and retrieving values of instance variables and instantiate its object | 11 |
| 7 | 22-01-2022 | Write a JAVA program to demonstrate static member data and static member methods | 12 |
| 8 | 25-01-2022 | Write a JAVA Program to demonstrate nested classes | 13 |
| 9 | 27-01-2022 | Write a JAVA program to demonstrate dynamic method dispatch. | 14 |
| 10 | 01-02-2022 | Write a JAVA program to implement inheritance and demonstrate use of method overriding. | 15 |
| 11 | 03-03-2022 | Write a JAVA program to implement the concept of importing classes from user defined package and creating packages. | 16 |
| 12 | 19-03-2022 | Write a program to demonstrate abstract class and abstract methods | 17 |
| 13 | 24-03-2022 | Write a JAVA Program to implement an array of objects of a class. | 18 |
| 14 | 04-05-2022 | Write a JAVA program to demonstrate String class and its methods. | 19 |
| 15 | 07-05-2022 | Write a JAVA program to implement the concept of exception handling by creating user defined exceptions. | 20 |
| 16 | 13-05-2022 | Write a JAVA program using synchronized threads, which demonstrates producer consumer concept. | 21 |
| 17 | 14-05-2022 | Write a JAVA program that creates three threads. First thread displays “Good Morning” every one second, second thread displays “Hello” every two seconds and the third thread displays “Welcome” every three seconds. | 24 |
| 18 | 21-05-2022 | Write a JAVA program which uses FileInputStream / FileOutPutStream Classes. | 26 |
| 19 | 25-05-2022 | Write a JAVA program to list all the files in a directory including the files present in all its subdirectories. | 27 |
| 20 | 27-05-2022 | Write a JAVA program to demonstrate the life cycle of applet | 28 |

**Program No:1**

**AIM:**

Develop a JAVA program to demonstrate the precedence and associativity among arithmetic

operators. The program should also demonstrate how the default precedence can be

overridden.

**SOURCE CODE:**

public class lab1 {

public static void main(String[] args) {

int a=10,b=5,c=20;

int x =++a - b + c++ + a \* b + c \* a- --b +c %b\*a;

System.out.println("before overriding the operator precedence");

System.out.println(x);

System.out.println("updated a b c values are:");

System.out.println(a);

System.out.println(b);

System.out.println(c);

int y=(++a -b)\* c++ + a \* b +c \* (a- --b) + c % b\*a;

System.out.println("the value of"+y);

System.out.println(a);

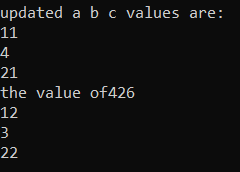
System.out.println(b);

System.out.println(c);

}

}

**OUTPUT:**



**Program No:2**

**AIM:**

Write a JAVA program to validate a date. The program should accept day, month and year and it should report whether they form a valid date or not.

**SOURCE CODE:**

import java.util.Scanner;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class lab2 {

public static boolean isValidDate(String d)

{

String regex="^(1[0-2]|[1-9])/(3[01]"

+"|[12][0-9]|0[1-9])/[0-9]{4}$";

Pattern pattern = Pattern.compile(regex);

Matcher matcher = pattern.matcher((CharSequence)d);

return matcher.matches();

}

public static void main(String[] args) {

Scanner sc=new Scanner (System.in);

System.out.println("Enter Month:-");

int month=sc.nextInt();

System.out.println("Enter Date:-");

int date=sc.nextInt();

System.out.println("Enter Year:-");

int year=sc.nextInt();

String x=month+"/"+date+"/"+year;

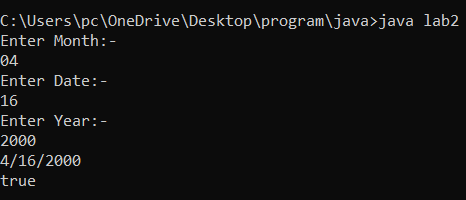
System.out.println(x);

System.out.println(isValidDate(x));

}

}

**OUTPUT:**



**Program No:3**

**AIM:**

Write a JAVA program to display the following pattern.

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

**SOURCE CODE:**

public class lab3 {

public static void main(String[] args) {

int i,j,row=5;

for(i=0;i<row;i++)

{

for(j=row-i;j>1;j--)

{

System.out.print(" ");

}

for(j=0;j<=i;j++)

{

System.out.print(i+1+" ");

}

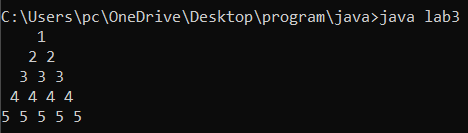
System.out.println();

}

}

}

**OUTPUT:**



**Program No:4**

**AIM:**

Write a JAVA program to print the first n members of Fibonacci series

**SOURCE CODE:**

import java.util.Scanner;

public class lab4

{

public static void main(String[] args) {

int n,a=0,b=0,c=1;

Scanner s=new Scanner (System.in);

System.out.print("Enter value of n:");

n=s.nextInt();

System.out.print("Fibonacci series:");

for(int i = 1; i <= n; i++)

{

a = b;

b = c;

c = a + b;

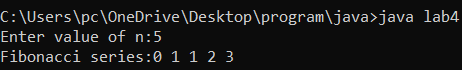
System.out.print(a+" ");

}

}

}

**OUTPUT:**

****

**Program No:5**

**AIM:**

Write a program to generate the multiplication tables of a range of numbers between m and n inclusive and m < n.

**SOURCE CODE:**

import java.util.Scanner;

public class lab5{

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

System.out.print("Enter the value of m:- ");

int m=s.nextInt();

System.out.print("Enter the value of n:- ");

int n=s.nextInt();

if(m<n)

{

for(int i=1;i<=10;i++)

{

for(int j=m;j<=n;j++)

{

System.out.printf(j+"\*"+i+"="+j\*i);

}

System.out.println(" ");

}

}

else

{

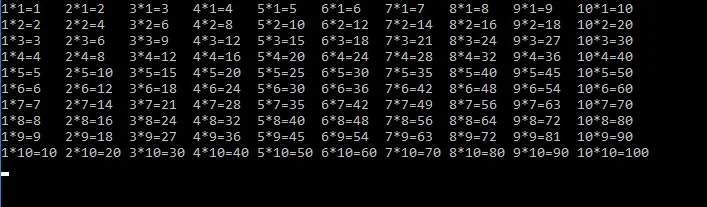
System.out.print("m should be lesser than n");

}

}

}

**OUTPUT:**

****

**Program No:6**

**AIM:**

Write a JAVA program to define a class, define instance methods for setting and retrieving

values of instance variables and instantiate its object.

**SOURCE CODE:**

public class lab6 {

int rollno;

String Name;

String Class;

String Section;

void insertRecord(int R,String N,String C,String S)

{

rollno=R;

Name=N;

Class=C;

Section=S;

}

void display()

{

System.out.println("The Student Details are");

System.out.println(rollno+" "+Name+" "+Class+" "+Section);

System.out.println();

}

public static void main(String[] args) {

lab6 s1=new lab6();

lab6 s2=new lab6();

s1.insertRecord(110,"Stud1","1 SEM MCA","A");

s1.display();

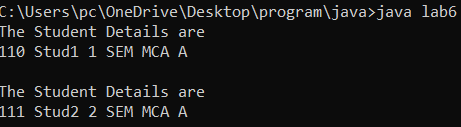
s2.insertRecord(111,"Stud2","2 SEM MCA","A");

s2.display();

}

}

**OUTPUT:**



**Program No:7**

**AIM:**

Write a JAVA program to demonstrate static member data and static member methods

**SOURCE CODE:**

class Student{

int rollno;

String name;

static String college = "KDC";

static void change(){

college = "KGI";

}

Student(int r, String n){

rollno = r;

name = n;

}

void display(){System.out.println(rollno+" "+name+" "+college);}

}

public class lab7{

public static void main(String args[]){

Student.change();

Student s1 = new Student(111,"STUD1");

Student s2 = new Student(222,"STUD2");

Student s3 = new Student(333,"STUD3");

s1.display();

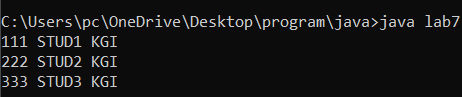
s2.display();

s3.display();

}

}

**OUTPUT:**

****

**Program No:8**

**AIM:**

Write a JAVA Program to demonstrate nested classes

**SOURCE CODE:**

class OuterClass {

int x = 10;

class InnerClass {

int y = 5;

}

}

public class lab8 {

public static void main(String[] args) {

OuterClass myOuter = new OuterClass();

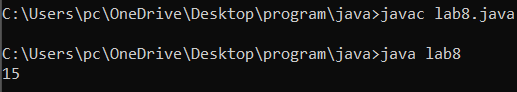
OuterClass.InnerClass myInner = myOuter.new InnerClass();

System.out.println(myInner.y + myOuter.x);

}

}

**OUTPUT:**



**Program No:9**

**AIM:**

Write a JAVA program to demonstrate dynamic method dispatch.

**SOURCE CODE:**

class Animal {

public void move() {

System.out.println("Animals can move");

}

}

class Dog extends Animal {

public void move() {

System.out.println("Dogs can walk and run");

}

}

public class lab9 {

public static void main(String args[]) {

Animal a = new Animal();

Animal b = new Dog();

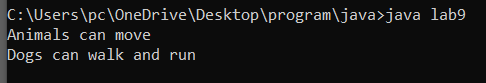
a.move();

b.move();

}

}

**OUTPUT:**

****

**Program No:10**

**AIM:**

Write a JAVA program to implement inheritance and demonstrate use of method overriding.

**SOURCE CODE:**

class Bike {

public void move() {

System.out.println("Bike has stopped running");

}

}

class univ extends Bike {

public void move() {

System.out.println("Bike is running");

}

}

public class lab10{

public static void main(String args[]) {

Bike a = new Bike();

Bike b = new univ();

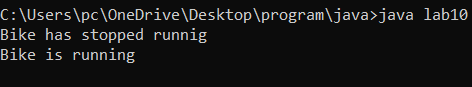
a.move();

b.move();

}

}

**OUTPUT:**

****

**Program No:11**

**AIM:**

Write a JAVA program to implement the concept of importing classes from user defined package and creating packages.

**SOURCE CODE:**

package pack;

public class A{

public void msg(int X,int Z)

{

int c=X+Z;

int d=X-Z;

int e=X/Z;

int f=X\*Z;

System.out.println("IMPLEMENTATION OF PACKAGE for Arithmetic Funtions for values "+X+" & "+Z);

System.out.println("Addition:"+c);

System.out.println("Subraction:"+d);

System.out.println("Division:"+e);

System.out.println("Multiplication:"+f);

}

}

---------------------------------------------------------------------------------------------------------------------

import pack.A;

class B{

public static void main(String args [])

{

int a=14,b=6;

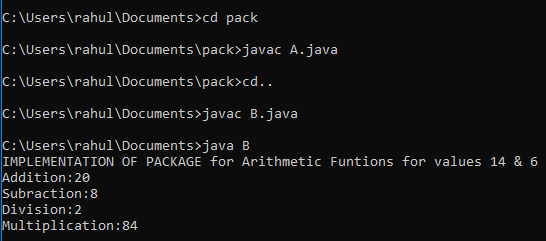
A obj=new A();

obj.msg(a,b);

}

}

**OUTPUT:**



**Program No:12**

**AIM:**

Write a program to demonstrate abstract class and abstract methods.

**SOURCE CODE:**

import java.io.\*;

import java.util.\*;

abstract class A {

int a=9,b=10,c;

abstract void mS();

public void B() {

c=a+b;

System.out.println("SUM of "+a+" & "+b+" is: "+c);

}

}

class D extends A {

public void mS() {

c=a-b;

System.out.println("SUBRACTION of "+a+" & "+b+" is: "+c);

}

}

class lab12 {

public static void main(String[] args) {

D d1 = new D();

d1.mS();

d1.B();

}

}

**OUTPUT:**

****

**Program No:13**

**AIM:**

Write a JAVA Program to implement an array of objects of a class

**SOURCE CODE:**

class lab13 {

public static void main(String args[])

{

Student[] arr;

arr = new Student[2];

arr[0] = new Student(100, "Student1");

arr[1] = new Student(101, "Student2");

System.out.println("Student data in student arr 0: ");

arr[0].display();

System.out.println( "Student data in student arr 1: ");

arr[1].display();

}

}

class Student {

public int id;

public String name;

Student(int id, String name)

{

this.id = id;

this.name = name;

}

public void display()

{

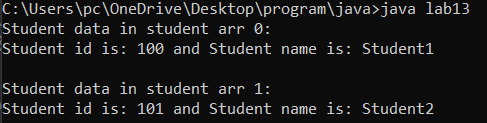
System.out.println("Student id is: " + id + " "+ "and Student name is: "+ name);

System.out.println();

}

}

**OUTPUT:**



**Program No:14**

**AIM:**

Write a JAVA program to demonstrate String class and its methods.

**SOURCE CODE:**

import java.io.\*;

import java.util.\*;

class ary

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

String s1,s2,up,low;

System.out.print("Enter the value for String 1: ");

s1=sc.nextLine();

System.out.print("Enter the value for String 2: ");

s2=sc.nextLine();

up=s1.toUpperCase();

low=s2.toLowerCase();

String rs=s1.replace("a","B");

System.out.println("The Length of String 1:"+s1.length());

System.out.println("The Length of String 2:"+s2.length());

System.out.println("The Upper Case of "+s1+" is:"+up);

System.out.println("The Lower Case of "+s2+" is:"+low);

System.out.println("The 2 concatenated Strings are : "+s1.concat(s2));

System.out.println("The character at position 2 of string "+s1+" is "+s1.charAt(1));

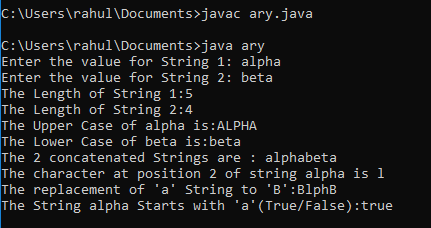
System.out.println("The replacement of 'a' String to 'B':"+rs);

System.out.println("The String "+s1+" Starts with 'a'(True/False):"+s1.startsWith("a"));

}

}

**OUTPUT:**



**Program No:15**

**AIM:**

Write a JAVA program to implement the concept of exception handling by creating user defined exceptions.

**SOURCE CODE:**

class UE{

public static void main(String args[]){

try{

throw new UserException("ERROR HAPPENS");

}

catch(UserException e){

System.out.println(e) ;

}

}

}

class UserException extends Exception{

String s1;

UserException(String s2) {

s1=s2;

}

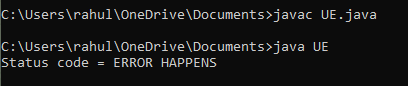
public String toString(){

return ("Status code = "+s1) ;

}

}

**OUTPUT:**



**Program No:16**

**AIM:**

Write a JAVA program using synchronized threads, which demonstrates producer consumer concept.

**SOURCE CODE:**

class Q

{

int n;

boolean valueSet = false;

synchronized int get()

{

while(!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)

{

while(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 ()

{

int i = 1;

while(i<10)

{

q.put(i++);

}

}

}

class Consumer implements Runnable

{

Q q;

Consumer(Q q)

{

this.q = q;

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

}

public void run()

{

int j = 1;

while(j<10)

{

q.get();

}

}

}

class pd

{

public static void main(String args[])

{

System.out.println("Press Control-C to stop.");

Q q = new Q();

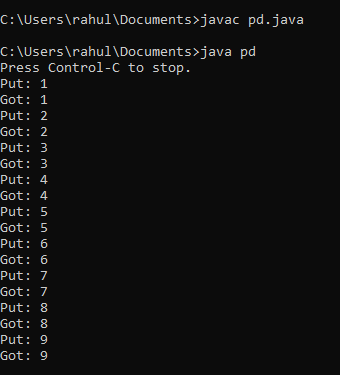
new Producer(q);

new Consumer(q);

}

}

**OUTPUT:**



**Program No:17**

**AIM:**

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

**SOURCE CODE:**

import java.lang.Thread;

import java.io.\*;

class A extends Thread

{

public void run()

{

for(int i=1;i<=5;i++)

{

try {

Thread.sleep(1000);

} catch(InterruptedException ex) {

Thread.currentThread().interrupt();

}

System.out.println("GOOD MORNING");

}

System.out.println("Exit From Thread A");

}

}

class B extends Thread

{

public void run()

{

for(int j=1;j<=5;j++)

{

try {

Thread.sleep(2000);

} catch(InterruptedException ex) {

Thread.currentThread().interrupt();

}

System.out.println("HELLO");

}

System.out.println("Exit From Thread B");

}

}

class C extends Thread

{

public void run()

{

for(int k=1;k<=5;k++)

{

try {

Thread.sleep(3000);

} catch(InterruptedException ex) {

Thread.currentThread().interrupt();

}

System.out.println("WELCOME");

}

System.out.println("Exit From Thread C");

}

}

public class TD

{

public static void main(String args[])

{

A threadA=new A();

B threadB=new B();

C threadC=new C();

System.out.println("Start Thread A");

threadA.start();

System.out.println("Start Thread B");

threadB.start();

System.out.println("Start Thread C");

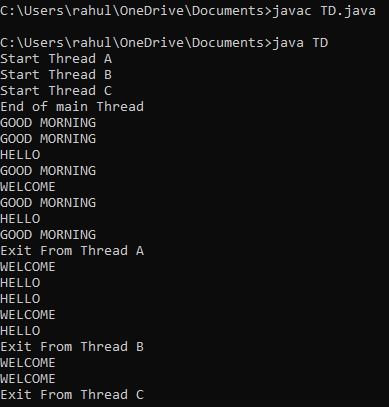
threadC.start();

System.out.println("End of main Thread");

}

}

**OUTPUT:**



**Program No:18**

**AIM:**

Write a JAVA program which uses FileInputStream / FileOutPutStream Classes.

**SOURCE CODE:**

import java.io.FileInputStream;

import java.io.FileOutputStream;

class IOTest{

public void readFile(){

try {

FileInputStream fis = new FileInputStream("C:/Users/pc/OneDrive/Desktop/program/java/ex1.txt");

FileOutputStream fos = new FileOutputStream("C:/Users/pc/OneDrive/Desktop/program/java/ex2.txt");

int i;

while((i=fis.read())!=-1){

fos.write(i);

}

System.out.println("Content writen successfully.");

} catch (Exception e) {

e.printStackTrace();

}

}

}

public class lab18 {

public static void main(String args[]){

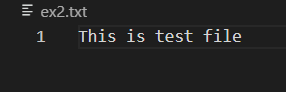
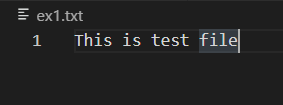
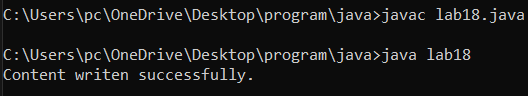
IOTest obj = new IOTest();

obj.readFile();

}

}

**OUTPUT:**



**Program No:19**

**AIM:**

Write a JAVA program to list all the files in a directory including the files present in all its subdirectories.

**SOURCE CODE:**

import java.io.File;

public class MC

{

private static void listFiles(String path)

{

File folder = new File(path);

File[] files = folder.listFiles();

for (File file : files)

{

if (file.isFile())

{

System.out.println(file.getName());

}

else if (file.isDirectory())

{

listFiles(file.getAbsolutePath());

}

}

}

public static void main(String[] args)

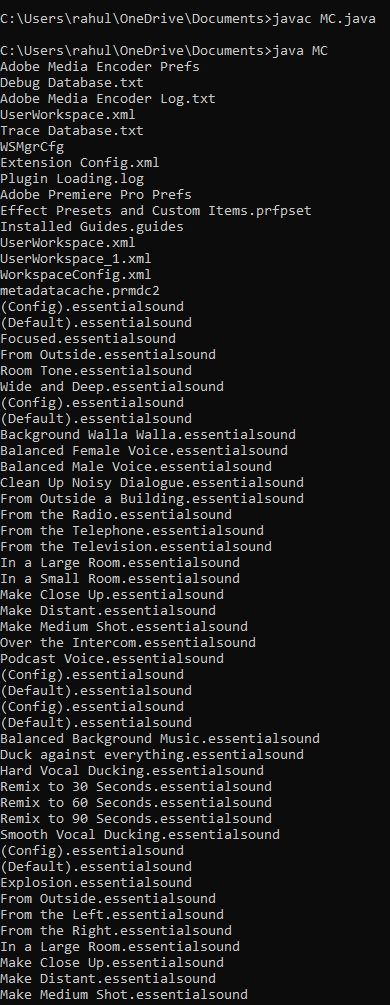
{

listFiles("C:\\Users\\rahul\\OneDrive\\Documents\\Adobe");

}

}

**OUTPUT:**



**Program No:20**

**AIM:**

Write a JAVA program to demonstrate the life cycle of applet

**SOURCE CODE:**

import java.applet.Applet;

import java.awt.Graphics;

import java.awt.\*;

/\*<applet code="alc.class" width="350" height="150"> </applet>\*/

public class alc extends Applet

{

public void init()

{

setBackground(Color.RED);

System.out.println("init() method is called");

}

public void start()

{

System.out.println("Start() method is called");

}

public void paint(Graphics g)

{

System.out.println("Paint(() method is called");

}

public void stop()

{

System.out.println("Stop() method is Called");

}

public void destroy() {

System.out.println("Destroy() method is Called");

}

}

**OUTPUT:**

