JAMAL MOHAMED COLLEGE (AUTONOMOUS)

College with Potential for Excellence

Accredited (3rd Cycle) with ‘A’ Grade by NAAC

DBT Star Scheme & DST-FIST Funded

(Affiliated to Bharathidasan University)

TIRUCHIRAPPALLI – 620 020

DEPARTMENT OF COMPUTER APPLICATIONS

BACHELOR OF COMPUTER APPLICATIONS

SEMESTER – III

JAVA PROGRAMMING LAB

CERTIFICATE

Name :

Register Number :

Roll Number :

This is to certify that this is the bonafide record of practical work done in the Computer Centre of Jamal Mohamed College, Tiruchirappalli - 20 during the Year 2022 - 2023.

A. ZULIGHA SHAFANA	R. SENTHAMIL SELVI
Staff in-charge	Staff in-charge

Submitted for the Bachelor of Computer Applications Practical Examination held at Jamal Mohamed College, Tiruchirappalli - 20, on __________

Tiruchirappalli - 20

Date : ________

Internal Examiner External Examiner

CONTENTS

Register Number: 21UCA381	**Using Control Statements - Find the prime numbers between 1 to 100**
Exercise No: 1	
Page No: 01	

Aim:

To write a java program to find the prime numbers between 1 to 100 using control statement.

Procedure:

1.Start the program

2.Set the visibility as public for the class in the program.

3.Initilize the value of num as “0” and maxcheck as “100”.

4.Set the condition using for loop for(i=2;i<=maxcheck;i++)

5.Print the prime numbers between 1 to 100.

6.Stop the execution of the program

Program:

```
public class primeNumbersFoundber {
public static void main(String[] args) {
int i;
int num = 0;
int maxCheck = 100;
boolean isPrime = true;
String primeNumbersFound = "";
for (i = 2; i <= maxCheck; i++) {
isPrime = CheckPrime(i);
if (isPrime) {
primeNumbersFound = primeNumbersFound + i + " ";
}
}
System.out.println("Prime numbers from 1 to " + maxCheck + " are:");
System.out.println(primeNumbersFound);
}
public static boolean CheckPrime(int numberToCheck) {
int remainder;
for (int i = 2; i <= numberToCheck / 2; i++) {
remainder = numberToCheck % i;
if (remainder == 0) {
return false;
}
}
return true;
}
}
```

Output:

```
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97.
```

Result:

Thus, the prime numbers from 1 to 100 are displayed and executed successfully.

Register Number: 21UCA381	**Using Control Statements - Count the number of digits for given integer using while loop**
Exercise No: 2	
Page No: 03	

Aim:

To write a java program to Count the number of digits for given integer using while loop.

Procedure:

1.Start the program

2.Set the visibility as public for the class in the program.

3.Declare the value of count as 0

4.Set the condition using while loop while (temp! =0)

5.Now print the number of digits for the given integer

6. Stop the execution of the program.

Program:

```
import java.util.Scanner;
public class CountDigits {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);
int num, temp;
int count = 0;
System.out.print("Enter any number : ");
num = scanner.nextInt();
temp = num;
while(temp != 0) {
count++;
temp /= 10;
}
System.out.println("\nTotal digits in " + num + " : "  + count);
}
}
```

Output:

```
Enter any number: 345678

Total digits in 345678: 6
```

Result:

Thus, the program to Count the number of digits for given integer using while loop displayed and executed successfully.

Register Number: 21UCA381	**Using Control Statements - Find the smallest and biggest digit in 6 digits number using for loop**
Exercise No: 3	
Page No: 05	

Aim:

To write a java program to Count the number of digits for given integer using while loop.

Procedure:

1. Start the program
2. Import packages in the starting of program. Here we have imported utility, language, io package.
3. Declare the value of largest as “0”and smallest as “9”
4. Set the condition using for loop
5. Set the visibility as public for the class in the program
6. Now print the largest and smallest numbers
7. Stop the execution of the program.

Program:

```
import java.util.*;
import java.lang.*;
import java.io.*;
class Gfg
{
static void Digits(int n)
{
int largest = 0;
int smallest = 9;
while(n != 0)
{
int r = n % 10;
largest = Math.max(r, largest);
smallest = Math.min(r, smallest);
n = n / 10;
}
System.out.println(largest + " " + smallest);
}
public static void main (String[] args) throws java.lang.Exception
{
int n = 234619;
Digits(n);
}
}
```

Output

```
9 1
```

Result:

Thus, the program to find the smallest and biggest digit in 6 digits number using for loop displayed and executed successfully.

Register Number: 21UCA381	**Using String handling functions - Find the ASCII character value of your name**
Exercise No: 4	
Page No: 07	

Aim:

To write a java program to find the ASCII character value of your name using String handling functions.

Procedure:

1. Start the program
2. Set the visibility as public for the class in the program.
3. Set the condition using for loop: for(int i=0;i<str.length();i++)
4. Print the respective ASCII value for the given name.
5. Stop the execution of the program.

Program:

```
public class ASCIIstring
{
public static void main(String[] args)
{
String str = "VIMALA";
int sum=0;
for(int i=0; i<str.length(); i++)
{
int asciiValue = str.charAt(i);
sum = sum+ asciiValue;
}
System.out.println("ASCII of " + str + "=" + sum);
}
}
```

Output:

```
ASCII of VIMALA = 442
```

Result:

Thus, the program finds the ASCII character value of your name using String handling functions.

Register Number: 21UCA381	**Using String handling functions - Count the total number of vowels and consonants in a given string**
Exercise No: 5	
Page No: 08	

Aim:

To write a java program to count the total number of vowels and consonants in a given string.

Procedure:

1. Start the program.

2. Set the visibility as public for the class in the program.

3. Declare the vcount as “0” and cCount as "0".

4. Now, declare the condition using for loop: for(int i=0;i<str.length();i++)

5. Print the total number of vowels and consonants in the given string.

6. Stop the execution of the program

Program:

```
public class CountVowelConsonant {

public static void main(String[] args) {

int vCount = 0, cCount = 0;

String str = "This is a really simple sentence";

str = str.toLowerCase();

for(int i = 0; i < str.length(); i++) {

if(str.charAt(i) == 'a' || str.charAt(i) == 'e' || str.charAt(i) == 'i' || str.charAt(i) == 'o' || str.charAt(i)
== 'u') {

vCount++;

}
else if(str.charAt(i) >= 'a' && str.charAt(i)<='z') {

cCount++;

}

}

System.out.println("Number of vowels: " + vCount);

System.out.println("Number of consonants: " + cCount);

}

}
```

Output:

```
Number of vowels: 10

Number of consonants: 17
```

Result:

Thus, the program counts the total number of vowels and consonants in a given string.

Register Number: 21UCA381	**To find the perimeter of circle and rectangle using class and objects**
Exercise No: 6	
Page No: 10	

Aim:

To write a java program to find perimeter of circle and rectangle using class and object.

Procedure:

1. Start the program

2. Set the visibility as public

3. Initialize the variables. Here we are initializing r,l,b,s1,s2,s3.

4. Write the formula to print the perimeter of circle, rectangle.

5. Print the perimeter of circle and rectangle using class and object.

6. Stop the execution of the program.

Program:

```
import java.util.Scanner;
public class Perimeter
{
int r, l, b, s1, s2, s3;
double pi = 3.14,perimeter;
Scanner s = new Scanner(System.in);
void circle()
{
System.out.print("Enter radius of circle:");
r = s.nextInt();
perimeter = 2 * pi * r;
System.out.println("Perimeter of circle:"+perimeter);
}
void rectangle()
{
System.out.print("Enter length of rectangle:");
l = s.nextInt();
System.out.print("Enter breadth of rectangle:");
b = s.nextInt();
perimeter = 2 * (l + b);
System.out.println("Perimeter of rectangle:"+perimeter);
}
public static void main(String[] args)
{
Perimeter obj = new Perimeter();
obj.circle();
obj.rectangle();
}
}
```

Output:

```
Enter radius of circle:4

Perimeter of circle:25.12

Enter length of rectangle:5

Enter breadth of rectangle:3

Perimeter of rectangle:16.0
```

Result:

Thus, the java program to find perimeter of circle and rectangle using class and object was executed successfully.

Register Number: 21UCA381	To demonstrate the following inheritance - Single Inheritance
Exercise No: 7	
Page No: 12	

Aim:

To write a java program to demonstrate single inheritance

Procedure:

1. Start the program.

2. Set the visibility as public for the class in the program.

3. Set the class name as “Maine”.

4. Print the program.

5. Stop the execution of the program.

Program:

```
class Animal

{

void Action_eat()

{

System.out.print("eating...");

}

}

class Dog extends Animal

{

void Action_bark()

{

System.out.print("barking...");
}

}

class Maine

{

public static void main(String args[]){

Dog d=new Dog();

System.out.print("The dog is ");

d.Action_bark();

System.out.print("\nThe dog is ");

d.Action_eat();

}

}
```

Output:

```
The dog is barking...
The dog is eating...
```

Result:

Thus, the java program to demonstrate single inheritance was executed successfully.

Register Number: 21UCA381	**To demonstrate the following inheritance - Multilevel Inheritance**
Exercise No: 8	
Page No: 14	

Aim:

To write a java program to demonstrate multilevel inheritance.

Procedure:

1.Start the program.

2.Set the class name as "person".

3.Now inherit class employee, hourly employee from person.

4.Give the class name as "Multilevel Inheritance".

5.Display the output.

Program:

```
class person
{
private String name;
person(String s)
{
setName(s);
}
public void setName(String s)
{
name = s;
}
```

```
{

public String getName()

{

return name;

}

public void display()

{

System.out.println("Name = " + name);

}

}

class Employee extends person

{

private int empid;

Employee(String sname,int id) //Constructor Method

{

super(sname);

setEmpid(id);

}

public void setEmpid(int id)

{

empid = id;

}

public int getEmpid()

{

return empid;

}
```

```
public void display()
{
super.display();
System.out.println("Empid = " + empid);
}
};
class HourlyEmployee extends Employee
{
private double hourlyRate;
private int hoursWorked;
HourlyEmployee(String sname,int id,double hr,int hw)
{
super(sname,id);
hourlyRate = hr;
hoursWorked = hw;
}
public double GetGrosspay()
{
return (hourlyRate * hoursWorked);
}
public void display()
{
super.display();
System.out.println("Hourly Rate = " + hourlyRate);
System.out.println("Hours Worked = " + hoursWorked);
System.out.println("Gross pay = " + GetGrosspay());

}

};

class MultilevelInheritance

{

public static void main(String[] args)

{

HourlyEmployee emp = new HourlyEmployee("Dinesh Thakur",1,15,1800);

emp.display();

}

}
```

Output:

```
Name = Dinesh Thakur

Empid = 1

Hourly Rate = 15.0

Hours Worked = 1800

Gross pay = 27000.0
```

Result:

Thus, the java program to demonstrate the multilevel inheritance executed successfully.

<table>
<tr><td><b>Register Number:</b> 21UCA381</td><td rowspan="3"><b>To demonstrate the concepts - Area of the shapes (interface)</b></td></tr>
<tr><td><b>Exercise No: 9</b></td></tr>
<tr><td><b>Page No: 18</b></td></tr>
</table>

Aim:

To write a java program to demonstrate the concept of interface.

Procedure:

1.Start the program

2.Declare the value of pi as "3.14"

3.Set the visibility as the public for the class in program.

4.Now print the area of rectangle and circle.

5.Stop the execution of the program.

Program:

```
interface area
{
double pi = 3.14;
double calc(double x,double y);
}
class rect implements area
{
public double calc(double x,double y)
{
return(x*y);
}
}
class cir implements area
{
public double calc(double x,double y)
{
return(pi*x*x);
}
}
class test7
{
public static void main(String arg[])
{
rect r = new rect();
cir c = new cir();
area a;
a = r;
System.out.println("\nArea of Rectangle is : " +a.calc(10,20));
a = c;
System.out.println("\nArea of Circle is : " +a.calc(15,15));
}
}
```

Output:

```
Area of Rectangle is : 200.0
Area of Circle is : 706.5
```

Result:

Thus, the java program to demonstrate the concept of interface was developed and executed successfully.

}

Register Number: 21UCA381	**To demonstrate the concepts - Abstract Class**
Exercise No: 10	
Page No: 21	

Aim:

To write a java program using abstract class.

Procedure:

1.Start the program

2.Set the visibility as public for the class in the program.

3.Print the output the abstract class.

4.Stop the execution of the program.

Program:

```
abstract class Animal{

public abstract void sound();

}

public class Dog extends Animal{

public void sound(){

System.out.println("Woof");

}

public static void main(String args[]){

Animal obj = new Dog();

obj.sound();

}

}
```

Output:

```
Woof
```

Result:

Thus, the java program using abstract class was developed and executed successfully.

Register Number: 21UCA381	**Using package to prepare an EB bill / Telephone bill / Student mark sheet with suitable fields**
Exercise No: 11	
Page No: 23	

Aim:

To write a java program to prepare an EB bill using packages.

Procedure:

1.Start the program.

2.Create the package customer data

3.Now import the package utility

4.Initialize variable cname, type, bn, current,previous, tbill, units.

5.Set the values for variable.

6.Print the output.

7.Stop the execution of program.

Program:

Customerdata.java

```
package customerdata;
import java.util.*;
public class Customerdata
{
Scanner in = new Scanner(System.in);
Scanner ins = new Scanner(System.in);
String cname,type;
int bn;
double current,previous,tbill,units;
public void getdata()
{
```

```
System.out.print ("\n\t Enter consumer number ");
bn = in.nextInt();
System.out.print ("\n\t Enter Type of connection (D for Domestic or C for Commercial) ");
type = ins.nextLine();
System.out.print ("\n\t Enter consumer name ");
cname = ins.nextLine();
System.out.print ("\n\t Enter previous month reading ");
previous= in.nextDouble();
System.out.print ("\n\t Enter current month reading ");
current= in.nextDouble();
}
public void calc()
{
units=current-previous;
if(type.equals("D"))
{
if (units<=100)
tbill=1 * units;
else if (units>100 && units<=200)
tbill=2.50*units;
else if(units>200 && units<=500)
tbill= 4*units;
else
tbill= 6*units;
}
else
{
if (units<=100)
tbill= 2 * units;
else if(units>100 && units<=200)
tbill=4.50*units;
else if(units>200 && units<=500)
tbill= 6*units;
else
tbill= 7*units;
}
}
public void display()
{
System.out.println("\n\t Consumer number = "+bn);
System.out.println ("\n\t Consumer name = "+cname);
if(type.equals("D"))
System.out.println ("\n\t type of connection = DOMESTIC ");
else
System.out.println ("\n\t type of connection = COMMERCIAL ");
System.out.println ("\n\t Current Month Reading = "+current);
System.out.println ("\n\t Previous Month Reading = "+previous);
System.out.println ("\n\t Total units = "+units);
System.out.println ("\n\t Total bill = RS "+tbill);
}
}
```

Ebill.java

```
import customerdata.*;
import java.util.*;
public class Ebill
{
public static void main (String args[])
{
Customerdata ob = new Customerdata();
ob.getdata();
ob.calc();
ob.display();
}
}
```

Output:

```
Consumer number: 1234

Consumer Name: AAA

Type of Connection: D

Current Month Reading: 1311

Previous Month Reading: 1111

Total Units: 200

Total Bill: 500
```

Result:

Thus, the java program to prepare an EB bill using packages was developed and executed successfully.

Register Number: 21UCA381	**Demonstrate multiple catch clauses**
Exercise No: 12	
Page No: 27	

Aim:

To write a java program to demonstrate multiple catch clauses.

Procedure:

1.Start the program.

2.Set the visibility as public for the class in the program.

3.Give the class name as “multiplecatchblock1”.

4.Use the keyword “clause” in the program.

5.Print the output of the program.

6.Stop the execution of the program.

Program:

```
public class MultipleCatchBlock1 {
public static void main(String[] args) {
try{
int a[]=new int[5];
a[5]=30/0;
}
catch(ArithmeticException e)
{
System.out.println("Arithmetic Exception occurs");
}
catch(ArrayIndexOutOfBoundsException e)
{
System.out.println("ArrayIndexOutOfBounds Exception occurs");
}
catch(Exception e)
{
System.out.println("Parent Exception occurs");
}
System.out.println("rest of the code");
}
}
```

Output:

```
Arithmetic Exception occurs

rest of the code
```

Result:

Thus the java program to demonstrate multiple catch clause was developed and executed successfully.

Register Number: 21UCA381	**Using Thread concept to solve the following - Display the System date and time with specific time interval using extends Thread class**
Exercise No: 13	
Page No: 29	

Aim:

To write a java program to print date and time using extends thread class.

Procedure:

1.Start the program.

2.Import utility package.

3.Set the visibility as public for the class in the program.

4.Insert a tryblock.

5.Now give catch class.

6.Insert a catch class inside a tryblock.

7.Print the output.

8.Stop the execution of the program.

Program:

```
import java.util.*;

public class DiffDemo extends Thread{

public static void main(String args[]) {

try {

long start = System.currentTimeMillis( );

System.out.println(new Date( ) + "\n");

Thread.sleep(5*60*10);
System.out.println(new Date( ) + "\n");

long end = System.currentTimeMillis( );

long diff = end - start;

System.out.println("Difference is : " + diff);

} catch (Exception e) {

System.out.println("Got an exception!");

}

}

}
```

Output:

```
Wed Dec 08 06:57:01 GMT 2021

Difference is : 3096
```

Result:

Thus, the java program to display the date and time using extend thread class was developed and executed successfully.

Register Number: 21UCA381	**Using Thread concept to solve the following - Display a set of numbers. If 25 even numbers have been displayed stop the thread and initiate a new thread class for displaying 25 odd numbers**
Exercise No: 14	
Page No: 31	

Aim:

To write java program to display the date and time using extend thread class.

Procedure:

1.Start the program.

2.Set the class name as timethread and use extend keyword.

3.Give the condition using for loop.

4.Now set the condition to print even and odd thread using if class.

5.Display the output.

6.Stop the execution of the program.

Program:

```
public class timerthread extends Thread{
public static void main(String[] args) {
Thread t1 = new Thread() {
public void run() {
for (int i = 1; i <= 50; i++) {
if (i % 2 == 0) {
System.out.println("even thread " + i);
}}}};
t1.start();
Thread t2 = new Thread() {
public void run() {
```

```
for (int i = 1; i <= 50; i++) {

if (i % 2 != 0) {

System.out.println("odd thread " + i);

}}}};

t2.start();

}}
```

Output:

```
odd thread 1

odd thread 3

even thread 2

odd thread 5

odd thread 7

even thread 4

odd thread 9

odd thread 11

even thread 6

odd thread 13

even thread 8

odd thread 15

odd thread 17

even thread 10

odd thread 19

odd thread 21
```

```
even thread 12
odd thread 23
odd thread 25
even thread 14
even thread 16
odd thread 27
even thread 18
odd thread 29
even thread 20
odd thread 31
even thread 22
odd thread 33
even thread 24
odd thread 35
even thread 26
odd thread 37
even thread 28
odd thread 39
even thread 30
odd thread 41
even thread 32
odd thread 43
even thread 34
odd thread 45
even thread 36
even thread 38

even thread 40

odd thread 47

even thread 42

odd thread 49

even thread 44

even thread 46

even thread 48

even thread 50
```

Result:

Thus, the java program to display date and time using extend thread class was developed and executed successfully.

Register Number: 21UCA381	**Find the properties of a given directory name**
Exercise No: 15	
Page No: 35	

Aim:

To write a java program to find the properties of a given directory name.

Procedure:

1. Start the program

2. Import required packages to the program

3. Read the type from the user

4. Use getFreeSpace() and getTotalSpace() methods to know the properties of folder name

5. Stop the execution of the program

Program:

```
import java.util.Scanner;

import java.io.File;

class file_directory

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the type you want to know: A/B/C");

String type=sc.nextLine();

String path = "D:\\college\\";

File file = new File(path);

switch (type) {
```

```
case "A":

System.out.println("Total Space: "+file.getTotalSpace());

break;

case "B":

System.out.println("Free Space: "+file.getFreeSpace());

break;

case "C":

System.out.println("Space used: "+(file.getTotalSpace() - file.getFreeSpace()));

break;

default:

System.out.println("Unknown space type: ");

}

}

}
```

Output:

```
Free Space: 1234567234
```

Result:

Thus, the java program for finding properties of a given directory name was developed and executed successfully.

Register Number: 21UCA381	**Draw a human face using Graphics class**
Exercise No: 16	
Page No: 37	

Aim:

To write a java program using graphics class to draw a human face.

Procedure:

1.Start the program.

2.Import some of the java packages such as applet, awt.

3.Set the visibility as public

4.Give the color and size as per your need.

5.Print the human face.

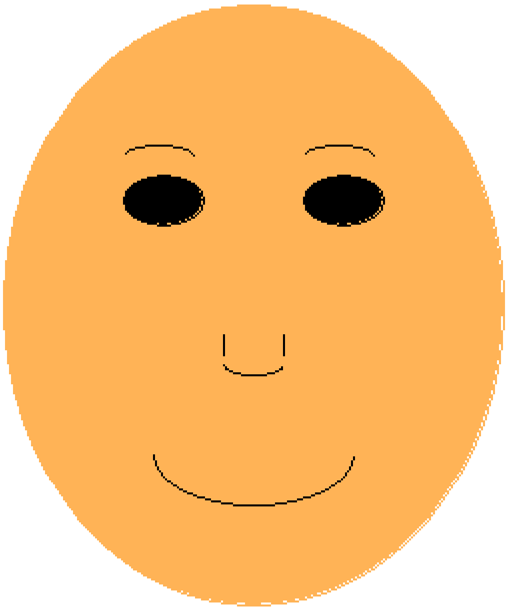
6.Stop the execution of the program.

Program:

```
import java.applet.*;
import java.awt.*;
/*<applet code = "Human_Face.class" width=500 height=500></applet>*/
public class Human_Face extends Applet
{
public void init()
{
setBackground(Color.white);
}
public void paint(Graphics g)
{
```

```
Color clr=new Color(255,179,86);

g.setColor(clr);

g.drawOval(100,100,250,300);

g.fillOval(100,100,250,300);

g.setColor(Color.black);

g.drawOval(160,185,40,25);

g.fillOval(160,185,40,25);

g.drawOval(250,185,40,25);

g.fillOval(250,185,40,25);

g.drawArc(160,170,35,10,0,180);

g.drawArc(250,170,35,10,0,180);

g.drawLine(210,265,210,275);

g.drawLine(240,265,240,275);

g.drawArc(210,275,30,10,0,-180);

g.drawArc(175,300,100,50,0,-180);

}

}
```

Output:



Result:

Thus, the java program to draw a human face using graphics class was developed and executed successfully.

Register Number: 21UCA381	**Demonstrate the layout managers - BorderLayout**
Exercise No: 17	
Page No: 40	

Aim:

To write a java program to demonstrate the layout managers.

Procedure:

1.Start the program.

2.Import the packages from java such as awt, swing.

3.Set the visibility as public.

4.Give the class name as border layout example.

5.Now add buttons and their respective values.

6.Set the size as (300,300), set the visibility as true.

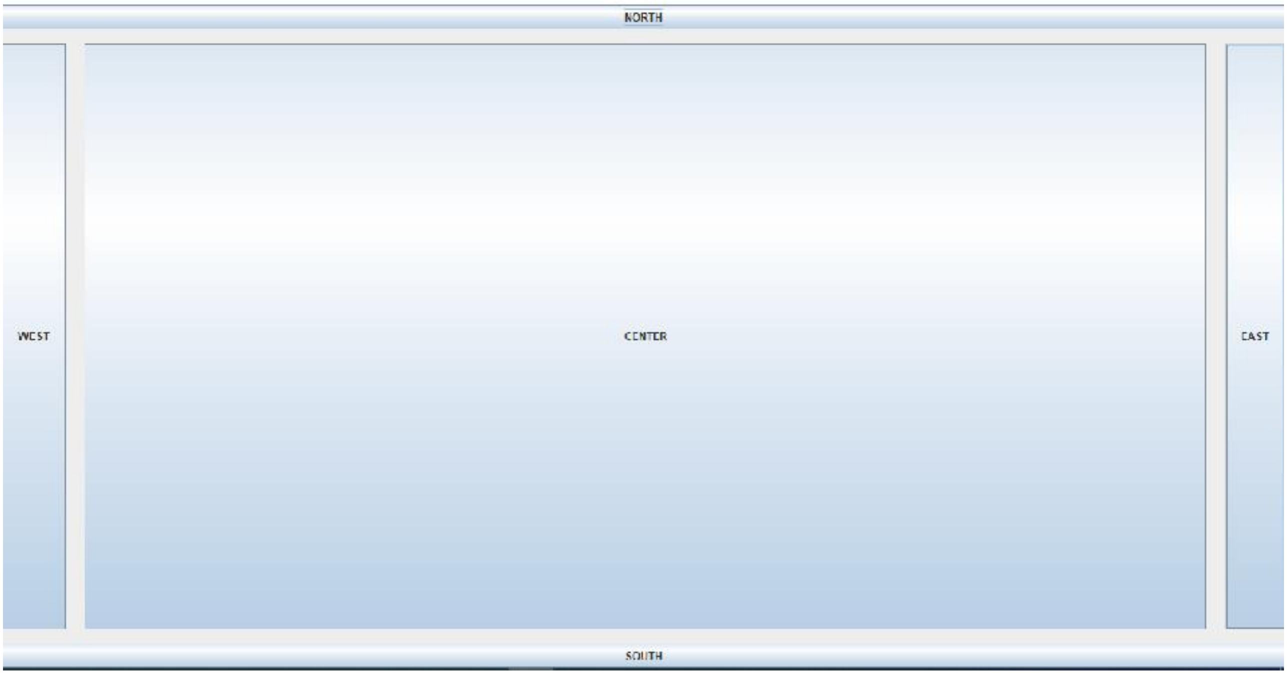
7.Print the output.

8.Stop the execution of the program.

Program:

```
import java.awt.*;

import javax.swing.*;

public class BorderLayoutExample

{

JFrame jframe;

BorderLayoutExample()

{

jframe = new JFrame();

JButton btn1 = new JButton("NORTH");
JButton btn2 = new JButton("SOUTH");

JButton btn3 = new JButton("EAST");

JButton btn4 = new JButton("WEST");

JButton btn5 = new JButton("CENTER");

jframe.setLayout(new BorderLayout(20, 15));

jframe.add(btn1, BorderLayout.NORTH);

jframe.add(btn2, BorderLayout.SOUTH);

jframe.add(btn3, BorderLayout.EAST);

jframe.add(btn4, BorderLayout.WEST);

jframe.add(btn5, BorderLayout.CENTER);

jframe.setSize(300,300);

jframe.setVisible(true);

}

public static void main(String argvs[])

{

new BorderLayoutExample();

}

}
```

Output:



Result:

Thus, the java program to demonstrate the layout managers was developed and executed successfully.

Register Number: 21UCA381	**Demonstrate the layout managers - GridLayout**
Exercise No: 18	
Page No: 43	

Aim:

To write a program to demonstrate the grid layout concept.

Procedure:

1.Start the program

2.Import java packages such as awt, swing

3.Insert the button and their respective value.

4.Set the visibility as public for the class in the program.

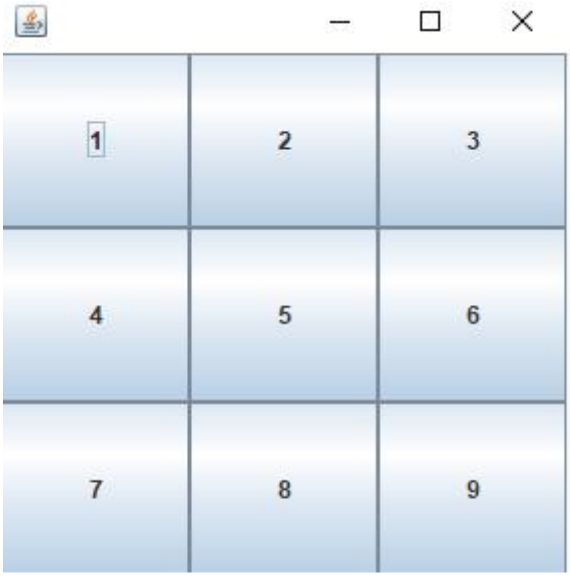
5.Display the output.

6.Stop the execution of the program

Program:

```
import java.awt.*;

import javax.swing.*;

public class MyGridLayout{

JFrame f;

MyGridLayout(){

f=new JFrame();

JButton b1=new JButton("1");

JButton b2=new JButton("2");

JButton b3=new JButton("3");

JButton b4=new JButton("4");

JButton b5=new JButton("5");
JButton b6=new JButton("6");

JButton b7=new JButton("7");

JButton b8=new JButton("8");

JButton b9=new JButton("9");

f.add(b1); f.add(b2); f.add(b3);

f.add(b4); f.add(b5); f.add(b6);

f.add(b7); f.add(b8); f.add(b9);

f.setLayout(new GridLayout(3,3));

f.setSize(300,300);

f.setVisible(true);

}

public static void main(String[] args) {

new MyGridLayout();

}

}
```

Output:



Result:

Thus, the java program to demonstrate the layout managers was developed and executed successfully.

Register Number: 21UCA381	**Using AWT controls to create a login page**
Exercise No: 19	
Page No: 46	

Aim:

To write a java program using AWT controls to create a login page.

Procedure:

1.Start the program.

2.Import packages such as awt, awt.events

3.Give the class name as my login window.

4.Give extends keyword.

5.Set the visibility as public.

6.Set the respective value.

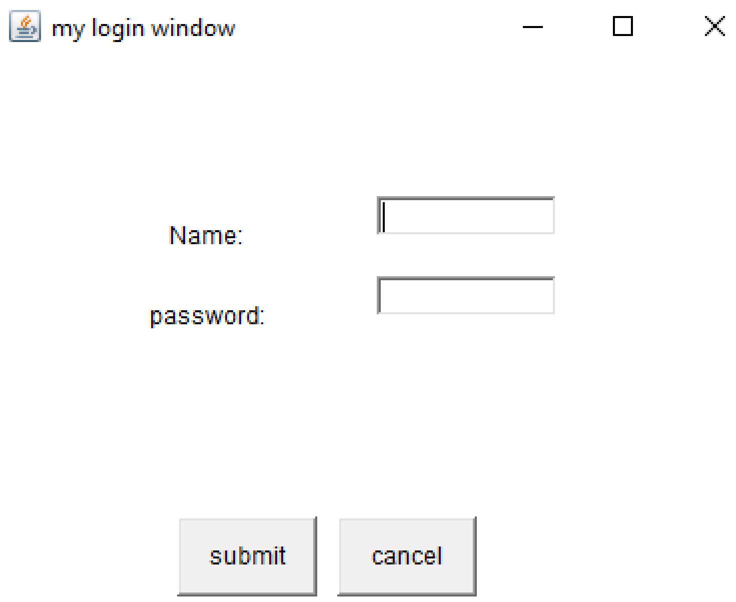
7.Display the output and stop the execution of the program.

Program:

```
import java.awt.*;

import java.awt.event.*;

class MyLoginWindow extends Frame

{

TextField name,pass;

Button b1,b2;

MyLoginWindow()

{

setLayout(new FlowLayout());

this.setLayout(null);
```

```
{
Label n=new Label("Name:",Label.CENTER);
Label p=new Label("password:",Label.CENTER);
name=new TextField(20);
pass=new TextField(20);
pass.setEchoChar('#');
b1=new Button("submit");
b2=new Button("cancel");
this.add(n);
this.add(name);
this.add(p);
this.add(pass);
this.add(b1);
this.add(b2);
n.setBounds(70,90,90,60);
p.setBounds(70,130,90,60);
name.setBounds(200,100,90,20);
pass.setBounds(200,140,90,20);
b1.setBounds(100,260,70,40);
b2.setBounds(180,260,70,40);
}
public static void main(String args[])
{
MyLoginWindow ml=new MyLoginWindow();
ml.setVisible(true);
ml.setSize(400,400);
ml.setTitle("my login window");

}

}
```

Output:



Result:

Thus, the program to create a login page was developed and executed successfully.