# GURU NANAK COLLEGE
## (AUTONOMOUS)
Chennai – 600 042.

BACHELOR OF COMPUTER SCIENCE

**[DEPARTMENT OF COMPUTER SCIENCE]**

2023 -2024

**Name** : DHARSHINI. S

**Reg. No** : 22131H1058013

**Year** : **II** **Semester : III**

**Subject Code** : **20UCSC306P**

**Subject** : **PROGRAMMING IN JAVA LAB**

# GURU NANAK COLLEGE(AUTONOMOUS)

**Chennai – 600 042.**

## DEPARTMENT OF COMPUTER SCIENCE

## BONAFIDE CERTIFICATE

| | | |
|---|---|---|
| **NAME** | : | DHARSHINI. S |
| **REG NO** | : | 22131410058013 |
| **CLASS** | : | II – 'A' |

*This is to certify that this is the bonafide record of the practical work done in* Programming In Java Practical. *at Guru Nanak College Computer Lab, during the Year 2023-2024.*

**Staff-In-Charge**

**Head of the programme**

Dr. R. RAJINI SUPENDRANATH
Associate Professor & Head
Computer Science Programme
Deputy Dean Academics
GURU NANAK COLLEGE (Autonomous)
CHENNAI-600 042.

*Submitted for the* End Semester Practical Examination. *B.Sc., Computer Science Practical Examination held on* 03.10.2023 *at Guru Nanak College, Chennai – 42.*

S. Jeyalakshmi

**Internal Examiner**

GURU NANAK COLLEGE AUTONOMOUS 42 B.Sc Computer Science

03/10/2023

**External Examiner**

# INDEX

| Exp No. 01 | Calculate Simple And Compound Interest |
|---|---|
| 27.06.2023 | By using String Buffer Class |

AIM:

To write a java program to calculate simple and compound interest.

ALGORITHM:

Step 1 : Start.

Step 2 : Read principle P, number of year n, rate of interest r.

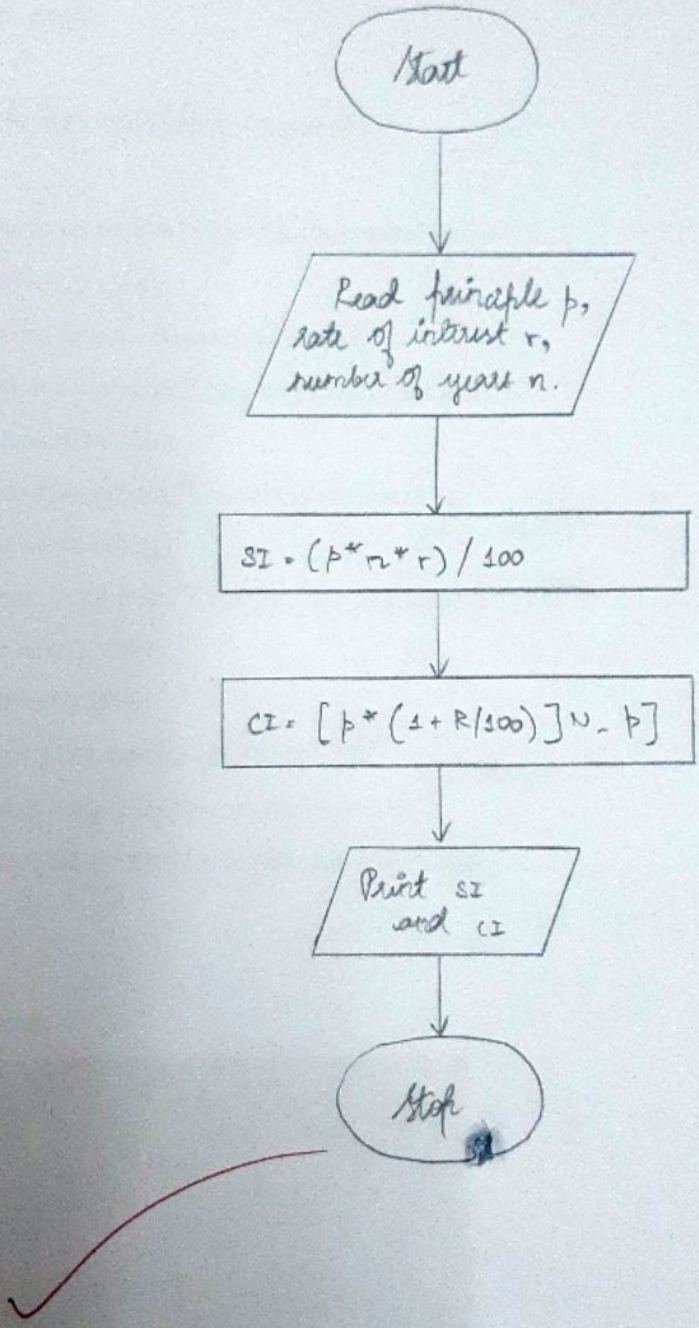
Step 3 : Calculate simple interest, SI = P*n*r/100.

Step 4 : Calculate compound interest, CI = [P*(1 + R/100)^N - P]

Step 5 : Print SI and CI

Step 6 : Stop.

FLOWCHART:


Start
Read principle p, rate of interest r, number of years n.
SI = (p*n*r) / 100
CI = [p*(1+R/100)]N - p]
Print SI and CI
Stop

# 1. SIMPLE AND COMPOUND INTEREST

```
import java.io.*;
import java.util.*;
class interest
{
public static void main(String args[])
{
System.out.println("Simple & Compound Interest");
Double p,n,r,si,ci;
Scanner sc=new Scanner(System.in);
System.out.println("Enter the Principle amount:");
p=sc.nextDouble();
System.out.println("Enter the No.of years:");
n=sc.nextDouble();
System.out.println("Enter the Rate of interest");
r=sc.nextDouble();
si=(p*n*r)/100;
ci=(p*(Math.pow((1+(r/100)),n)))-p;
System.out.println("Simple Interest="+si);
System.out.println("Compound Interest="+ci);
}
}
```

**Output:**

```
D:\ubai>javac interest.java

D:\ubai>java interest
Simple & Compound Interest
Enter the Principle amount:
50000
Enter the No.of years:
2
Enter the Rate of interest
9
Simple Interest=9000.0
Compound Interest=9405.000000000007
```

RESULT:

Thus the program has been successfully executed.

| Exp No: 2 | Largest Of Three Numbers |
|---|---|
| 03.02.2023 | |

Aim:

To write a java program to print largest of three given numbers.

Algorithm:

Step 1 : Start

Step 2 : Read the number a, b and c.

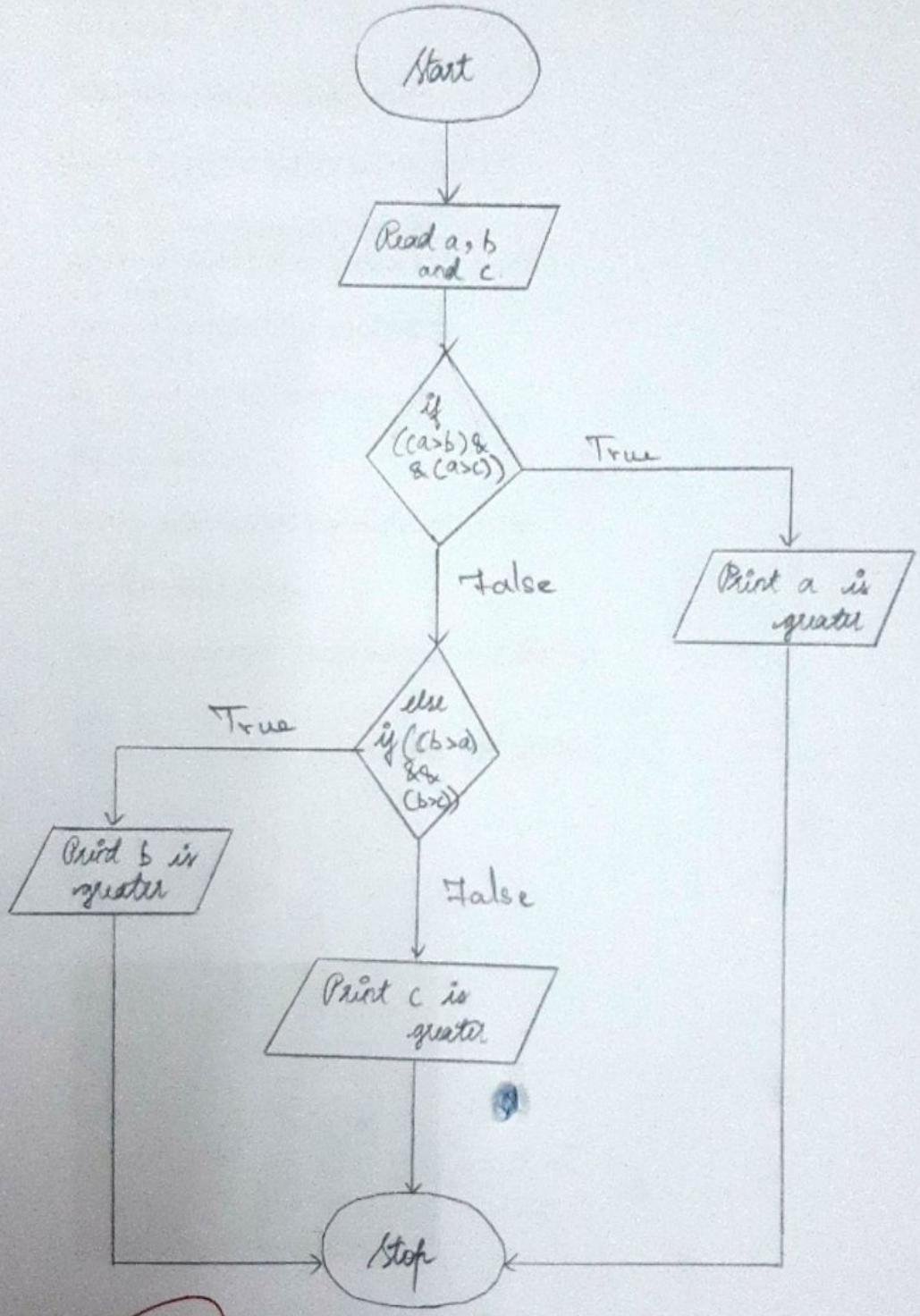
Step 3 : Check if (a>b) && (a>c) if yes, print a is largest and go to step 6.

Step 4 : Otherwise check if (b>a) && (b>c) if yes, print b is largest and go to step 6.

Step 5 : Otherwise print c is largest.

Step 6 : Stop.

FLOWCHART :


Start
Read a, b and c
if ((a>b) && (a>c))
True
Print a is greater
False
else if ((b>a) && (b>c))
True
Print b is greater
False
Print c is greater
Stop

## 2. LARGEST OF THREE NUMBERS

```
import java.io.*;
import java.util.*;
class largest
{
public static void main(String args[])
{
System.out.println("Largest of three numbers");
int a,b,c;
Scanner sc=new Scanner(System.in);
System.out.println("Enter number 1");
a=sc.nextInt();
System.out.println("Enter number 2");
b=sc.nextInt();
System.out.println("Enter number 3");
c=sc.nextInt();
if((a>b)&&(a>c))
{
System.out.println(a+" is greater than "+b+" and "+c);
}
else if((b>a)&&(b>c))
{
System.out.println(b+" is greater than "+a+" and "+c);
}
else
System.out.println(c+" is greater than "+a+" and "+b);
}
}
```

**Output:**

```
D:\ubai>javac largest.java

D:\ubai>java largest
Largest of three numbers
Enter number 1
16
Enter number 2
25
Enter number 3
12
25 is greater than 16 and 12
```

RESULT:

Thus the program has been successfully executed.

| Exp No: 3 | TO ILLUSTRATE CLASS AND OBJECT |
|---|---|
| 03.07.2023 | |

AIM:

To write a java program to creat class and object.

ALGORITHM:

Step 1 : Start

Step 2 : Declare length and width.

Step 3 : Calculate area = length * width

Step 4 : Return the value.

Step 5 : Creat main class rect area.

Step 6 : Declare area 1 and area 2.

Step 7 : Creat the object 1 and object 2 for class rectangle.

Step 8 : Accessing the class members area 1.

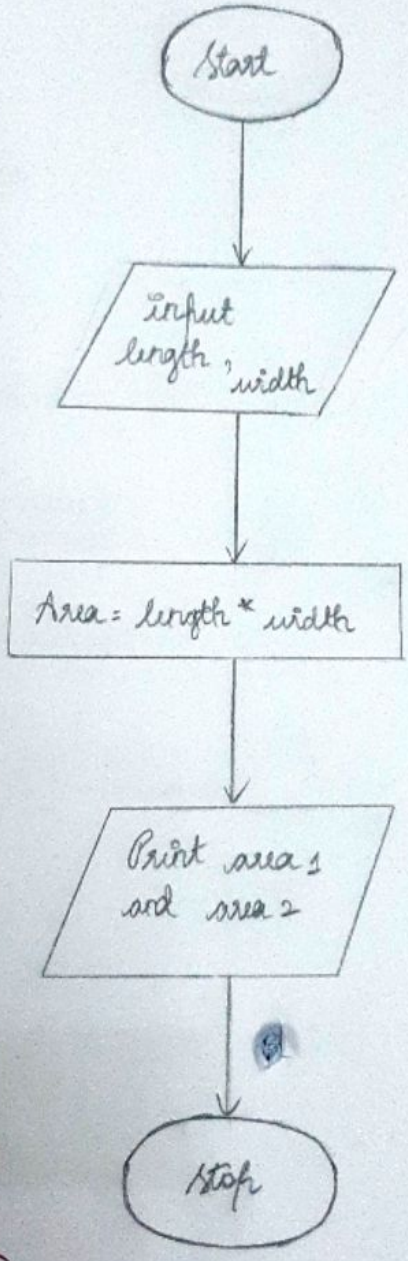
Step 9 : Access the class member area 2.

Step 10 : Print area 1.

Step 11 : Print area 2.

Step 12 : Stop.

FLOWCHART :


Start
Input length, width
Area = length * width
Print area 1 and area 2
Stop

## 3. ILLUSTRATION OF CLASS AND OBJECT

```
class rectangle
{
int length,width;
void getdata(int x,int y)
{
length=x;
width=y;
}
int rectarea()
{
int area=length*width;
return(area);
}
}
class rectarea
{
public static void main(String args[])
{
int area1,area2;
rectangle rect1=new rectangle();
rectangle rect2=new rectangle();
rect1.length=15;
rect1.width=10;
area1=rect1.length*rect1.width;
rect2.getdata(20,12);
area2=rect2.rectarea();
System.out.println("area1="+area1);
System.out.println("area2="+area2);
}
}
```

**Output:**

```
D:\ubai>javac rectarea.java

D:\ubai>java rectarea
area1=150
area2=240
```

RESULT:

Thus the program has been successfully executed.

Exp No: 4

06.07.2023

# RANDOM NUMBERS

AIM:

To create a java program for generating random numbers using random class.

ALGORITHM:

Step 1: Start.

Step 2: Create the class with name generate random

Step 3: Open a main function with public access specifier. void main return type and auto initialization.

Step 4: Create an object Rand for a Random class to access the integer variables rand_int1 and rand_int2.

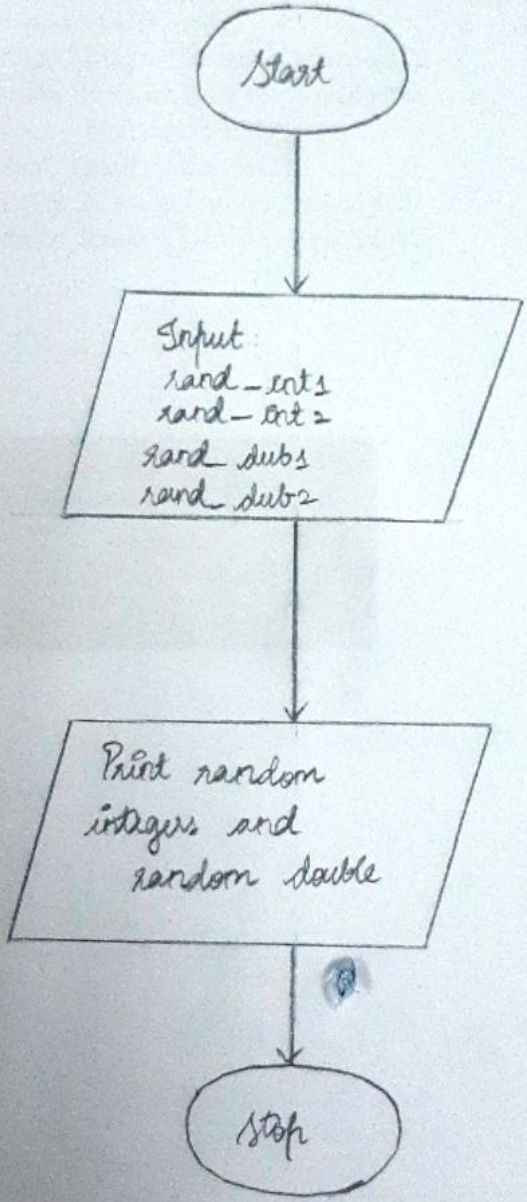
Step 5: Print the Random integer values.

Step 6: Create the variable rand_dub1, rand_dub2 to store double random values.

Step 7: Print random double values.

Step 8: Stop.

FLOWCHART:


Start
Input
rand_int1
rand_int2
rand_dub1
rand_dub2
Print random integers and random double
Stop

## 4. GENERATE RANDOM NUMBERS

```
import java.util.Random;
public class generaterandom
{
public static void main(String args[])
{
Random rand = new  Random();
int rand_int1 = rand.nextInt(1000);
int rand_int2 = rand.nextInt(1000);
System.out.println("Random Integers: "+rand_int1);
System.out.println("Random Integers: "+rand_int2);
double rand_dub1 = rand.nextDouble();
double rand_dub2 = rand.nextDouble();
System.out.println("Random Doubles: "+rand_dub1);
System.out.println("Random Doubles: "+rand_dub2);
}
}
```

**Output:**

```
D:\ubai>javac generaterandom.java

D:\ubai>java generaterandom
Random Integers: 644
Random Integers: 893
Random Doubles: 0.32511939373350174
Random Doubles: 0.7914386628134115
```

RESULT: Thus the program has been successfully executed.

| Exp No: 5 | CALENDAR CLASS |
|---|---|
| 06.03.2023 | |

AIM:

To write a java program to write calendar class.

ALGORITHM:

Step 1 : Start.

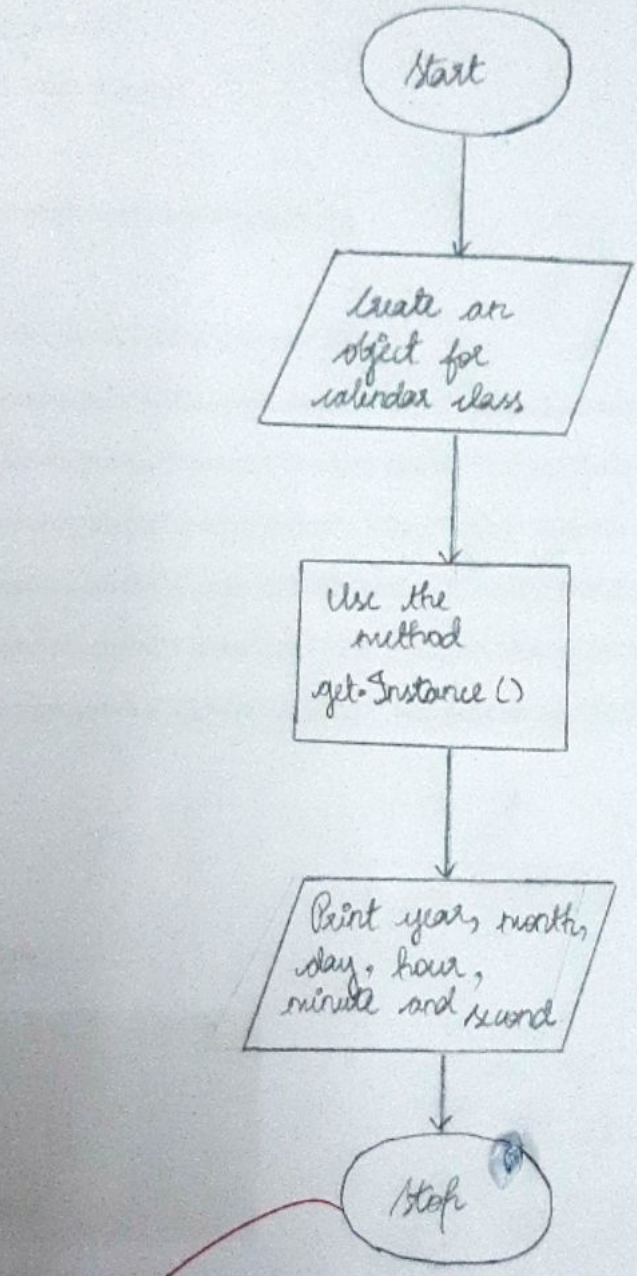
Step 2 : Create an object for calendar class

Step 3 : Set the system calendar class.

Step 4 : Display the year, month, day of month, hour, minute and seconds.

Step 5 : Stop.

FLOWCHART:


Start
Create an object for calendar class
Use the method get. Instance ()
Print year, month, day, hour, minute and second
Stop

## 5. CALENDAR CLASS

```
import java.util.*;

public class calendar2

{

public static void main(String []args)

{

Calendar cal=Calendar.getInstance();

System.out.println("Current Calendar's Year:"+cal.get(Calendar.YEAR));

System.out.println("Current Calendar's Month:"+cal.get(Calendar.MONTH));

System.out.println("Current Calendar's Day:"+cal.get(Calendar.DATE));

System.out.println("Current HOUR:"+cal.get(Calendar.HOUR));

System.out.println("Current MINUTE:"+cal.get(Calendar.MINUTE));

System.out.println("Current SECOND:"+cal.get(Calendar.SECOND));

}

}
```

**OUTPUT:**

RESULT: Thus the program has been successfully executed.

| Exp No: 6 | To Illustrate Constructor |
|---|---|
| 06.07.2023 | |

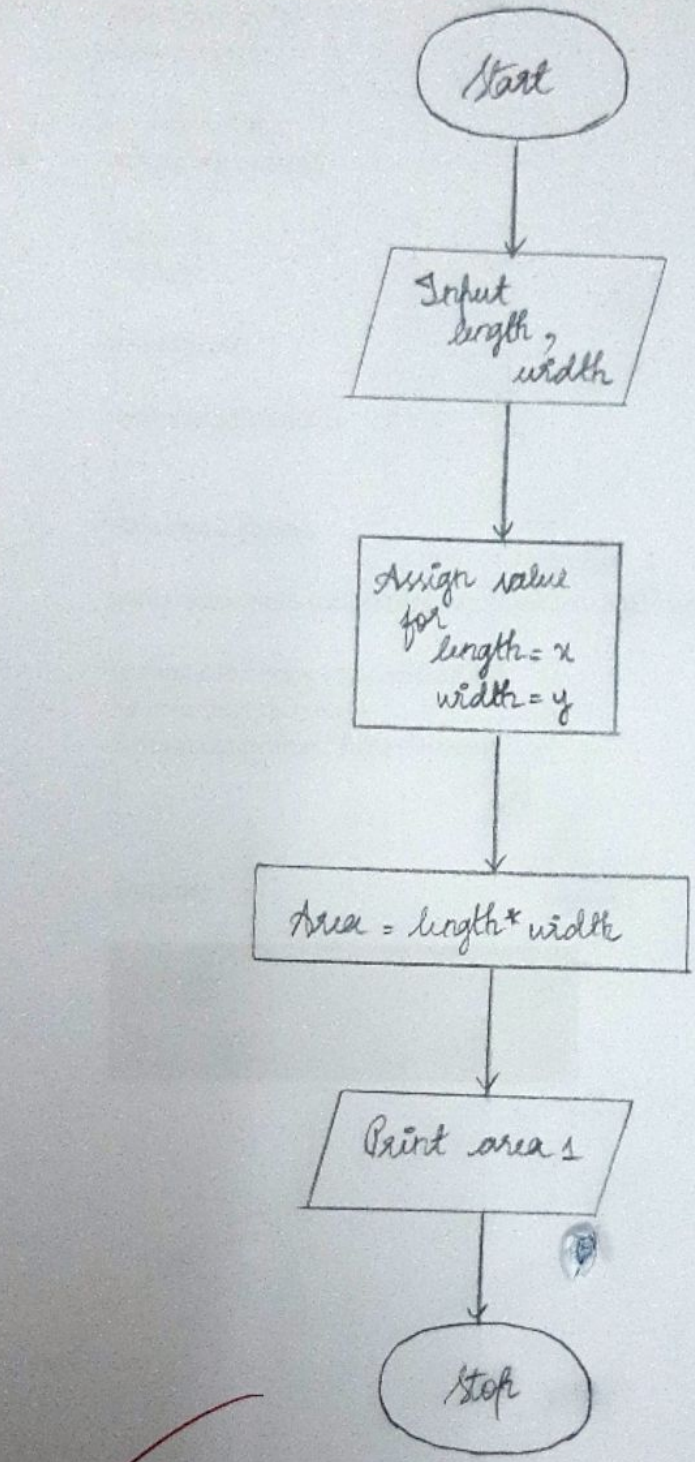
AIM:

To write a Java program to create a constructor.

ALGORITHM:

Step 1 : Start.
Step 2 : Create the class with class name.
Step 3 : Declare length and width.
Step 4 : Assign length = x, width = y
Step 5 : Calculate length * width.
Step 6 : Create the main class.
Step 7 : Create the object for the class rectangle.
Step 8 : Access the class member with object
Step 9 : Print area 1.
Step 10 : Stop.

FLOWCHART:


Start
Input length, width
Assign value for length = x width = y
Area = length * width
Print area 1
Stop

## 6. ILLUSTRATION OF CONSTRUCTOR

```
import java.io.*;
class rectangle
{
int length,width;
rectangle(int x,int y)
{
length=x;
width=y;
}
int rectarea()
{
return(length*width);
}
}
class rectanglearea
{
public static void main(String args[])throws IOException
{
rectangle rect=new rectangle(15,10);
int area=rect.rectarea();
System.out.println("Area="+area);
}
}
```

**Output:**

```
D:\ubai>javac rectanglearea.java

D:\ubai>java rectanglearea
Area=150
```

RESULT:

Thus the program has been successfully executed.

| Exp No: 7 | METHOD OVERLOADING |
| --- | --- |
| 11.02.2023 | |

AIM:

To write a java program to illustrate method overloading.

ALGORITHM:

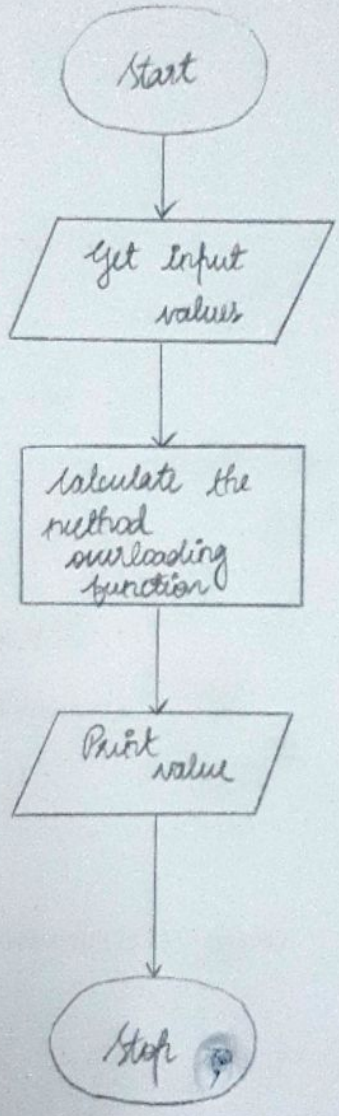
Step 1 : Start

Step 2 : Define the overload multiple methods with the same 'test' but different number and type of arguments.

Step 3 : Call the overload method 'test' a number of times with different arguments and illustrate that the appropriate method is invoked.

Step 4 : Stop.

FLOWCHART :


Start
Get Input values
Calculate the method overloading function
Print value
Stop

## 7. METHOD OVERLOADING

```
class overloaddemo{
void test()
{
System.out.println("No Parameters");
}
void test(int a)
{
System.out.println("a:"+a);
}
void test(int a,int b)
{
System.out.println("a and b:"+a+" "+b);
}
double test(double a)
{
System.out.println("double a:"+a);
return a*a;
}
}
class overload
{
public static void main(String args[])
{
overloaddemo ob=new overloaddemo();
double result;
ob.test();
ob.test(10);
ob.test(10,20);
result=ob.test(123.25);
System.out.println("Result of ob.test(123.25):"+result);
}
}
```

**Output:**

```
D:\ubai> javac overload.java

D:\ubai> java overload
No Parameters
a:10
a and b: 10 20
double a: 123.25
Result of ob.test(123.25):15190.5625
```

RESULT:

Thus the program has been successfully executed.

| Exp No: 8 | To Illustrate Inheritance |
| --- | --- |
| 11.07.2023 | |

AIM:

To write a java program to work inheritance

ALGORITHM:

Step 1 : Start

Step 2 : Declare a class with class name room.

Step 3 : Declare a variable and also assign value to the variable.

Step 4 : Declare a method area() and return the value.

Step 5 : Extend the class with superclass.

Step 6 : Declare a variable in subclass and also extending the superclass.

Step 7 : Declare a method in subclass volume()

Step 8 : Assign the value to the variable.

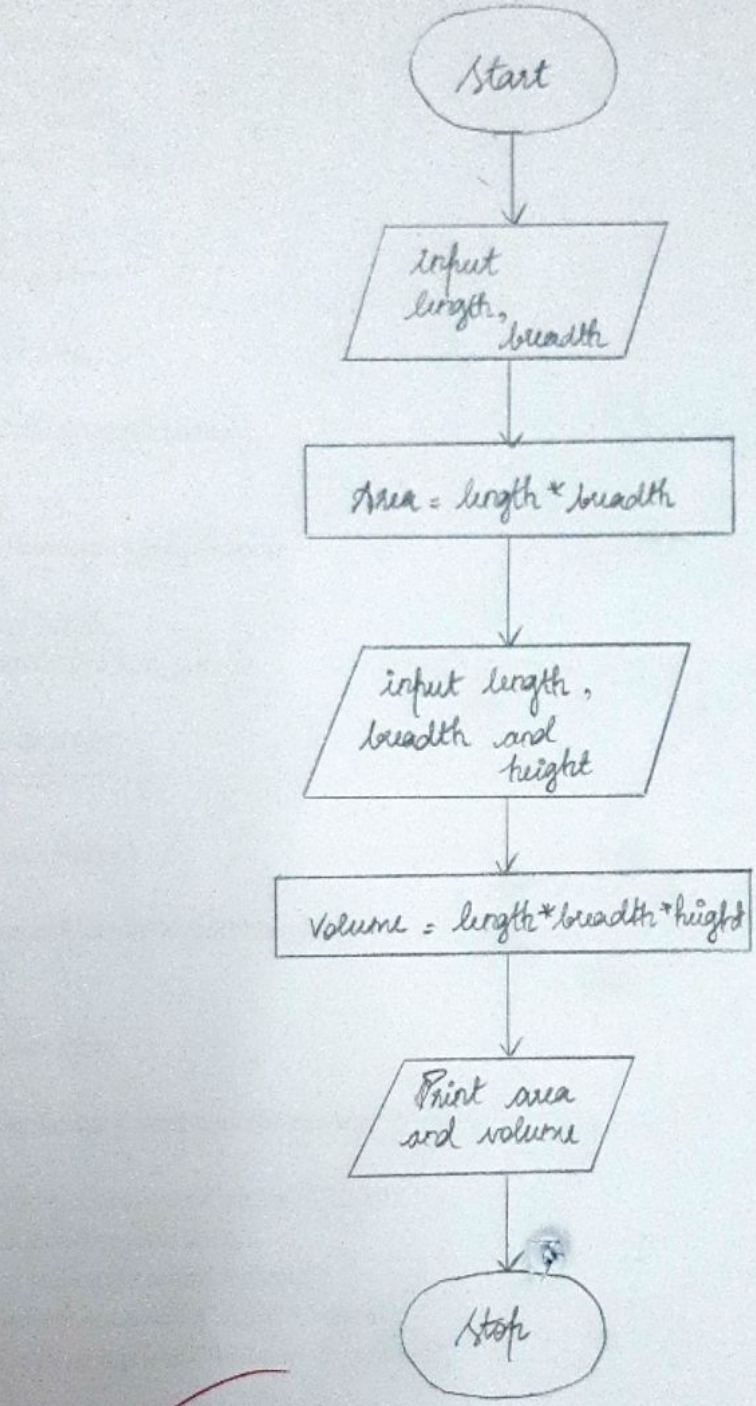
Step 9 : Create an object for superclass and subclass.

Step 10 : Print area.

Step 11 : Print volume.

Step 12 : Stop the program.

FLOWCHART :


Start
input length, breadth
Area = length * breadth
input length, breadth and height
Volume = length*breadth*height
Print area and volume
Stop

## 8. ILLUSTRATION OF INHERITANCE

```
class room{
int length;
int breadth;
room(int x,int y)
{
length=x;
breadth=y;
}
int area()
{
return(length*breadth);
}
}
class sroom extends room
{
int height;
sroom(int x,int y,int z)
{
super(x,y);
height=z;
}
int volume()
{
return(length*breadth*height);
}
}
class inher
{
public static void main(String args[])
{
sroom room1=new sroom(14,12,10);
int area1=room1.area();
int volume1=room1.volume();
System.out.println("Area1="+area1);
System.out.println("Volume="+volume1);
}}
```

**Output:**

```
D:\ubai>javac inher.java

D:\ubai>java inher
Area1=168
Volume=1680
```

RESULT:

Thus the program has been successfully executed.

| Exp No: 9 | METHOD OVERRIDING |
| --- | --- |
| 11.01.2023 | |

AIM:

To write a java program to illustrate method overriding.

ALGORITHM:

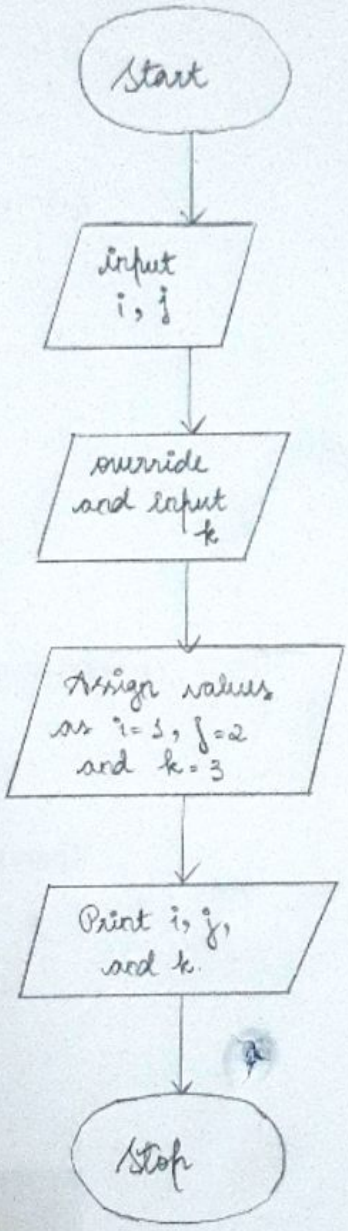
Step 1: Start

Step 2: Define and overload multiple methods with the same name 'test' but with different number and type of arguments.

Step 3: Call the overload method 'test' a number of times with different arguments and illustrate that the appropriate method is invoked.

Step 4: Stop.

FLOWCHART :


Start
Input i, j
override and input k
Assign values as i=3, j=2 and k=3
Print i, j, and k.
Stop

## 9. METHOD OVERRIDING

```
class over
{
int i,j;
over(int a,int b)
{
i=a;
j=b;
}
void show()
{
System.out.println("i and j:"+i+" "+j);
}
}
class over1 extends over
{
int k;
over1(int a,int b,int c)
{
super(a,b);
k=c;
}
void show()
{
System.out.println("i="+i+" j="+j+" k="+k);
}
}
class override
{
public static void main(String args[])
{
over1 ob=new over1(1,2,3);
ob.show();
}
}
```

**Output:**

```
D:\ubai>javac override.java

D:\ubai>java override
i=1 j=2 k=3
```

RESULT:

Thus the program has been successfully executed.

| Exp No: 10 | PACKAGES |
| --- | --- |
| 14.07.2023 | |

AIM:

To write a java program to implement packages.

ALGORITHM:

Step 1 : Start

Step 2 : Create a package package1.

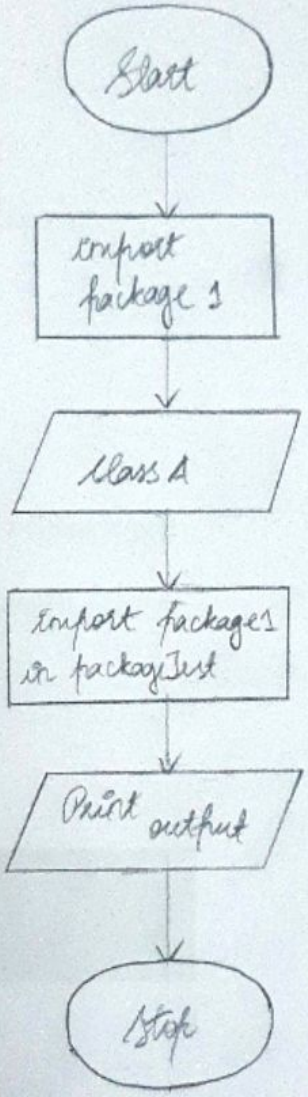
Step 3 : Create a class packagetest1 and import the package in the program.

Step 4 : Create an object obj A to access function in packages.

Step 5 : Print the output.

Step 6 : Stop.

FLOWCHART :


Start
import package 1
class A
import package1 in package2
Print output
Stop

10. **PACKAGES**

**classA.java**

```
package package1;
public class classA
{
public void displayA()
{
System.out.println("Class A");
}
}
```

**packagetest.java**

```
import package1.*;
class packagetest
{
public static void main(String args[])
{
classA objA=new classA();
objA.displayA();
}
}
```

**Output:**

```
D:\ubai>javac packagetest.jav

D:\ubai>java packagetest
Class A
```

RESULT:

Thus the program has been successfully executed.

| Exp No: 11 | To Illustrate Thread |
|---|---|
| 14.07.2023 | |

AIM: To write a java program to create a thread.

ALGORITHM:

Step 1: Start.

Step 2: Start the main thread by creating a program.

Step 3: The child thread is created by the initialization of an object of new thread, which is derived from thread class. This involves the new threads class conductor.

Step 4: In constructor of new thread class start() method is called to began execution of new child thread.

Step 5: The start() method involves the run() method, which is the entry point for the new thread. This causes the child threads for loop to begin.

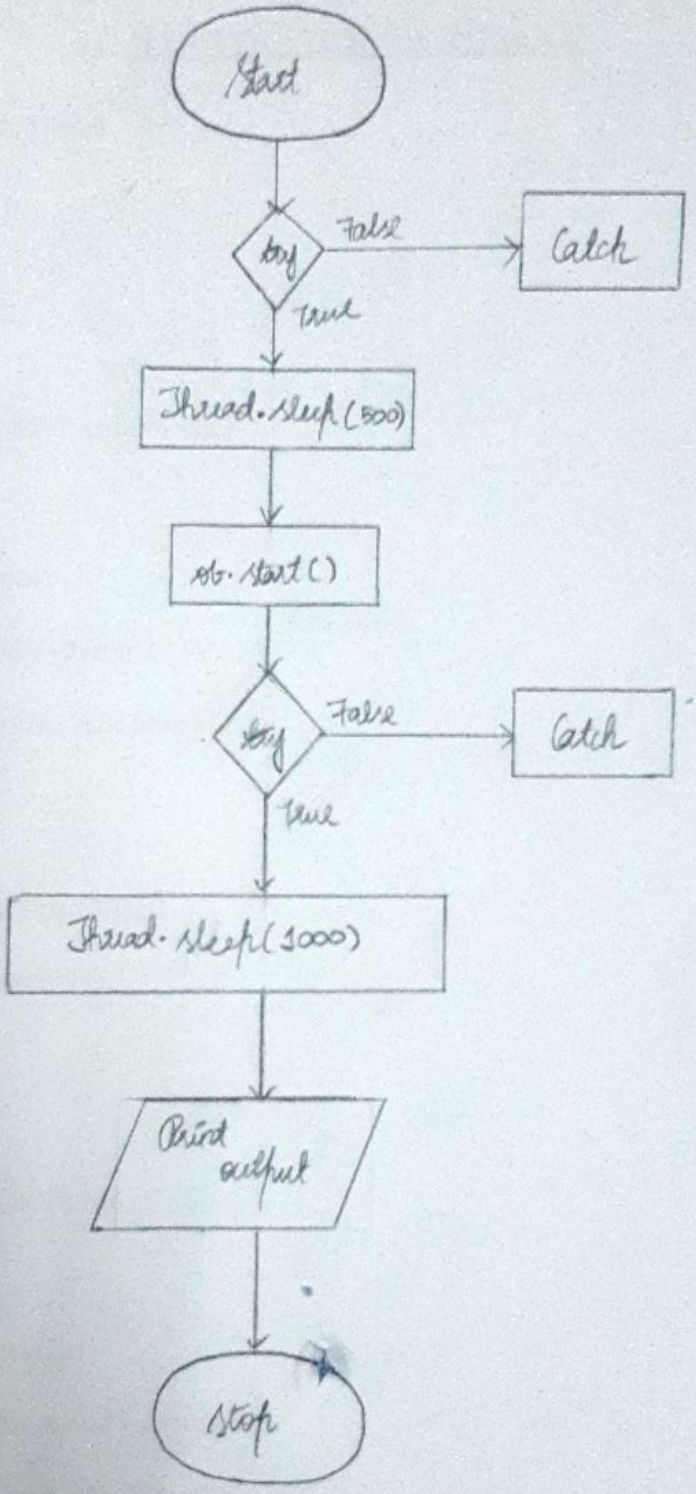
Step 6: After called start() new threads constructor written to main().

Step 7: When the main thread resumes, it enters its for loop.

Step 8: The both threads continues sharing the CPU until their loops.

Step 9: Stop.

FLOWCHART:


Start
try
False
Catch
true
Thread.sleep(500)
ob.start()
try
False
Catch
true
Thread.sleep(1000)
Print output
Stop

## 11. ILLUSTRATION OF THREAD

```
class newthread extends Thread
{
public void run()
{
try
{
for(int i=1;i<=5;i++)
{
System.out.println("Child Thread i=" +i);
Thread.sleep(500);
}
}
catch(InterruptedException e)
{
System.out.println("Child interrupted.");
}
System.out.println("Exiting child thread.");
}
}
class mainthread
{
public static void main(String args[])
{
newthread ob=new newthread();
ob.start();
try
{
for(int j=1;j<=5;j++)
{
System.out.println("Main Thread j=" +j);
Thread.sleep(1000);
}
}
catch(InterruptedException e)
{
System.out.println("Main thread interrupted.");
}
System.out.println("Main thread exiting");
}
}
```

**Output:**

```
D:\ubai>javac mainthread.java

D:\ubai>java mainthread
Child Thread i=1
Main Thread j=1
Child Thread i=2
Main Thread j=2
Child Thread i=3
Child Thread i=4
Main Thread j=3
Child Thread i=5
Exiting child thread.
Main Thread j=4
Main Thread j=5
Main thread exiting
```

RESULT: Thus the program has been successfully executed.

Exp No: 12

14.07.2023

# To Illustrate Exception Handling

AIM:

To write a java program to illustrate exception handling.

ALGORITHM:

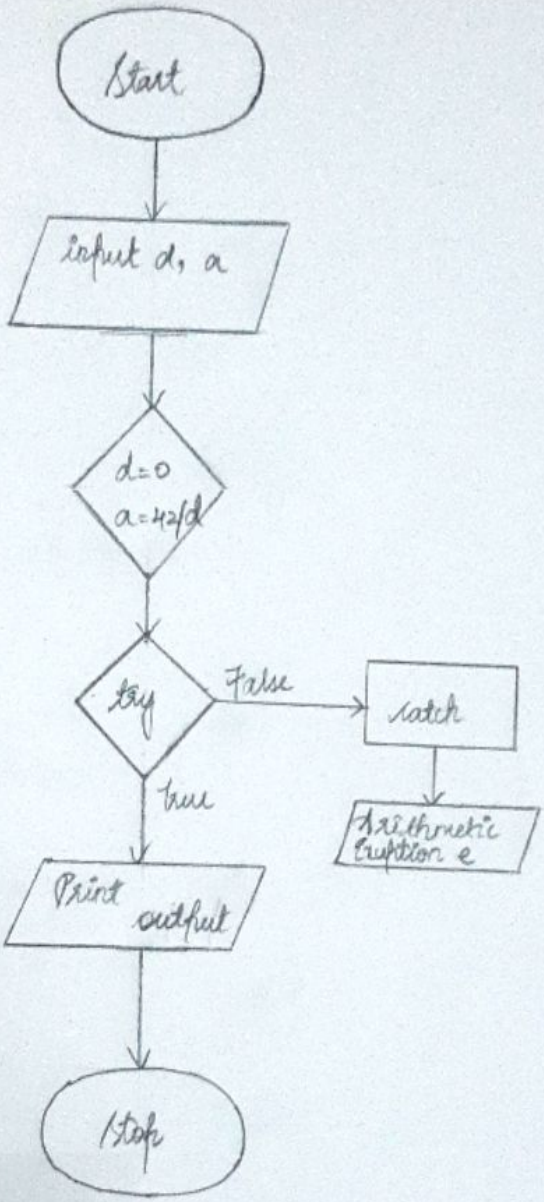
Step 1: Start

Step 2: To handle a runtime error, the code to be monitored for error in enclosed inside a try block.

Step 3: A catch clause is included that specifies the exception type that is to be caught. The exception handle code is given in catch clause.

Step 4: Stop.

FLOWCHART:


Start
input d, a
d=0
a=a/d
try
False
catch
Arithmetic Exception e
true
Print output
Stop

## 12. ILLUSTRATION OF EXCEPTION HANDLING

```
class exceptiontest
{
public static void main(String args[])
{
int d,a;
try{
d=0;
a=42/d;
System.out.println("This will not be printed");
}
catch(ArithmeticException e)
{
System.out.println("Division by Zero");
}
System.out.println("After catch statement");
}
}
```

**Output:**

```
D:\ubai>javac exceptiontest.java

D:\ubai>java exceptiontest
Division by Zero
After catch statement
```

RESULT: Thus the program has been successfully executed.

| Exp No: 13 | GENERATE VARIOUS SHAPES USING APPLET |
|---|---|
| 18.08.2023 | |

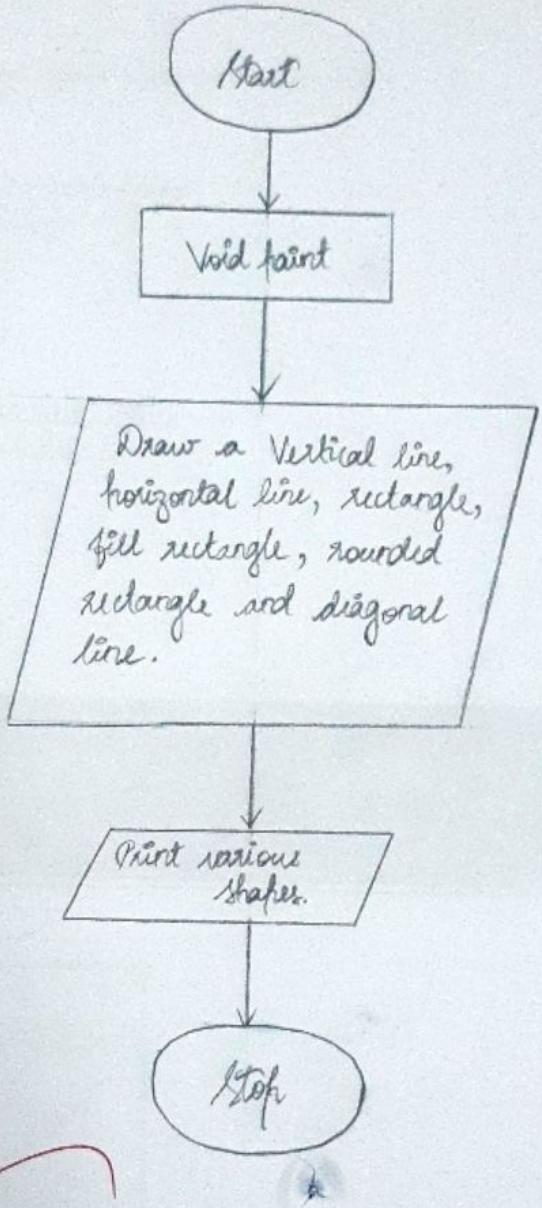
AIM:

To write a Java program to generate various shapes using applet.

ALGORITHM:

Step 1: Start.
Step 2: Draw a vertical line.
Step 3: Draw a horizontal line.
Step 4: Draw a rectangle.
Step 5: Draw a filled rectangle.
Step 6: Draw a rounded rectangle.
Step 7: Draw a diagonal line.
Step 8: Stop.

FLOWCHART:


Start
Void paint
Draw a Vertical line, horizontal line, rectangle, fill rectangle, rounded rectangle and diagonal line.
Print various shapes.
Stop

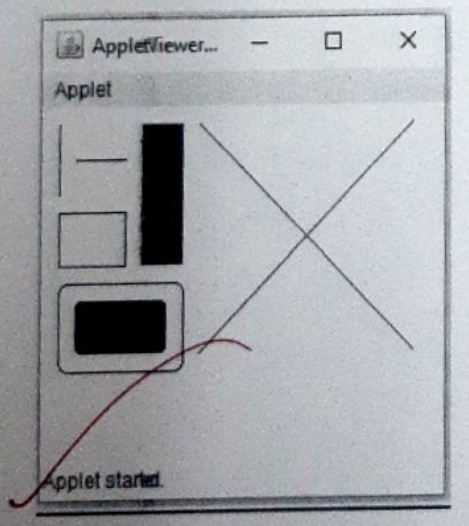
## 13. VARIOUS SHAPES USING APPLET

```
import java.awt.*;
import java.applet.*;
/*
<applet code="linerectangle.java" width=250 height=200>
</applet>
*/
public class linerectangle extends Applet{
public void paint(Graphics g){
g.drawLine(10,10,10,50);
g.drawLine(20,30,50,30);
g.drawRect(10,60,40,30);
g.fillRect(60,10,30,80);
g.drawRoundRect(10,100,80,50,10,10);
g.fillRoundRect(20,110,60,30,5,5);
g.drawLine(100,10,230,140);
g.drawLine(100,140,230,10);
}}
```

**Output:**

```
D:\ubai javac linerectangle.java
Note: linerectangle.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

D:\ubai appletviewer linerectangle.java
Warning: Applet API and AppletViewer are deprecated.
```



RESULT:

Thus the program has been successfully executed.

Exp No: 14

23.08.2023

# POINT CLASS MANIPULATION

AIM:

To write a java applet program to point class manipulation.

ALGORITHM:

Step 1 : Start.

Step 2 : Initialize the width, height, $x_1 = 100$, $y_1 = 100$.

Step 3 : The new object the point and to pass argument of the style (10, 10).

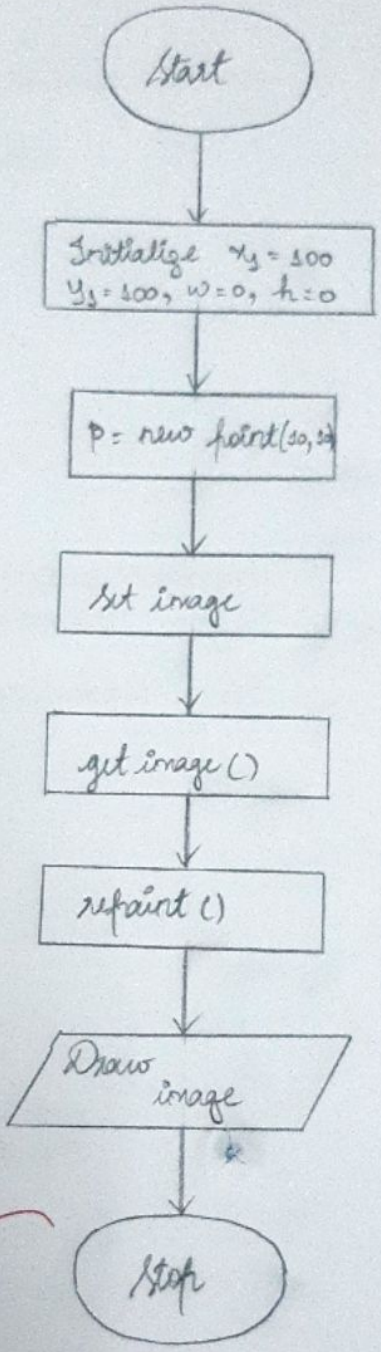
Step 4 : Set the image to get to initial the method in the applet.

Step 5 : Set to add mouse listeners to draw the image in the frames.

Step 6 : Calling the method "repaint()" to repaint the before location of the image.

Step 7 : Stop.

FLOWCHART:


Start
Initialize x = 100
y = 100, w = 0, h = 0
p = new point(10,
Set image
get image ()
repaint ()
Draw image
Stop

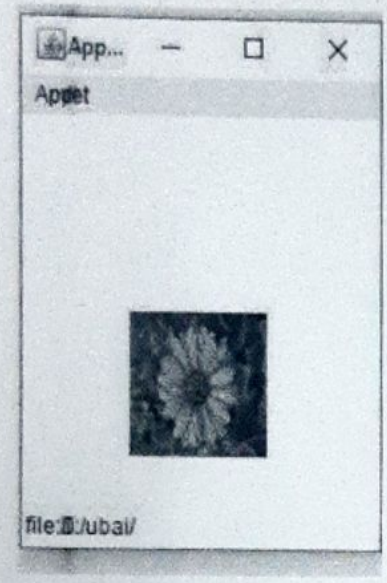
# 14. POINT CLASS MANIPULATION

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code="impoint.java" width=200 height=200>
</applet>*/
public class impoint extends Applet
{
Point  p;
int w=0;
int h=0;
int x1=100;
int y1=100;
Image theImage;
public void init()
{
p=new Point(10,10);
theImage=getImage(getDocumentBase(),"flowers.jpg");
addMouseListener(new MouseAdapter()
{
public void mousePressed(MouseEvent me)
{
p.move(me.getX(),me.getY());
x1=p.x;
y1=p.y;
w+=15;
h+=15;
repaint();
});}
public void paint(Graphics g)
{
g.drawImage(theImage,x1,y1,w,h,this);
showStatus(getCodeBase().toString());
}
}
```

**Output:**

```
D:\ bai>javac impoint.java
Not  impoint.java uses or overr ies a deprecated API.
Not  Recompile with -Xlint:depr ation for details.

D:\ bai>appletviewer impoint.jav
War ng: Applet API and AppletVi ver are deprecated.
```



RESULT:

Thus the program has been successfully executed.

EXP NO: 15

01.09.2023

# DRAW A HUMAN FACE

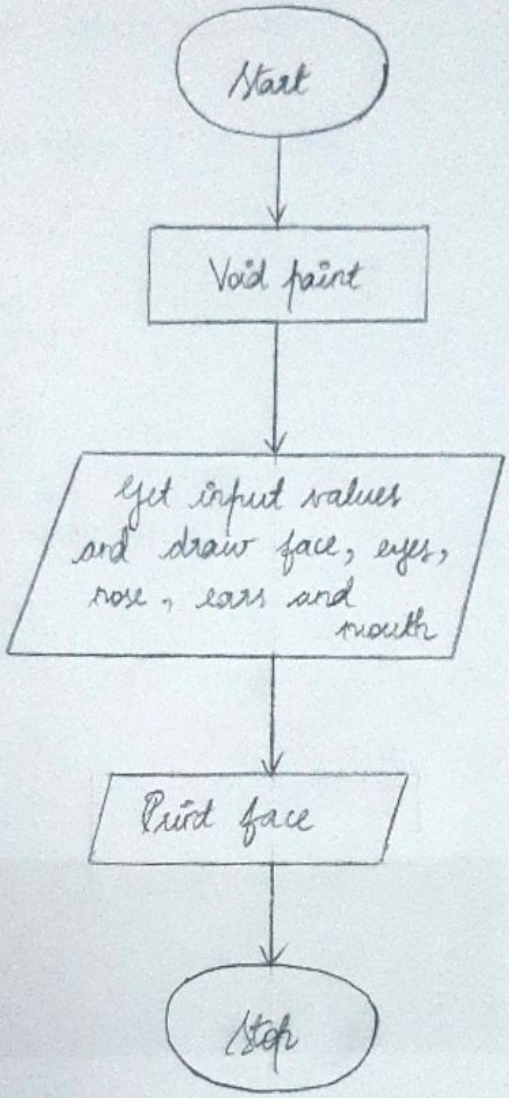
AIM:

To write a java program to draw a human face.

ALGORITHM:

Step 1: Start.
Step 2: Draw an outline of a face.
Step 3: Draw left and right eyes.
Step 4: Draw nose.
Step 5: Draw ears.
Step 6: Draw a mouth.
Step 7: Stop.

FLOWCHART:


Start
Void paint
get input values and draw face, eyes, nose, ears and mouth
Print face
Stop

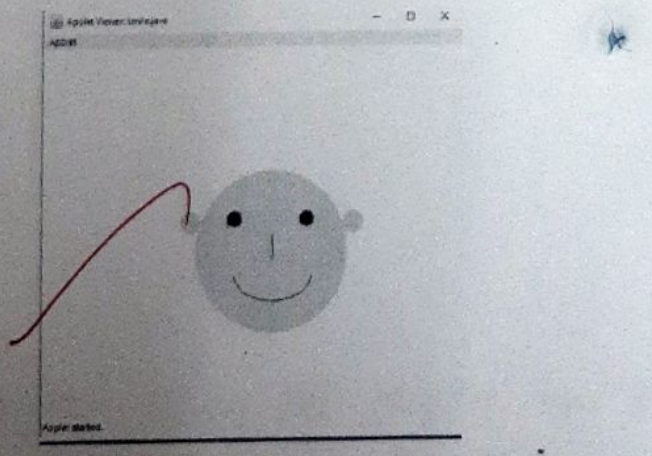
## 15. HUMAN FACE

```
import java.awt.*;
import java.applet.*;
/*
<applet code="smile.java" width=550 height=450>
</applet>
*/
public class smile extends Applet
{
public void paint(Graphics g)
{
g.setColor(Color.yellow);
g.fillOval(200,150,200,200);
g.setColor(Color.black);
g.fillOval(240,200,20,20);
g.fillOval(335,200,20,20);
g.drawLine(300,230,300,260);
g.drawArc(250,250,100,60,180,180);
g.setColor(Color.yellow);
g.fillOval(393,200,28,28);
g.fillOval(179,200,28,28);
}
}
```

**Output:**

```
D:\ubai>javac smile.java
Note: smile.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

D:\ubai>appletviewer smile.java
Warning: Applet API and AppletViewer are deprecated.
```



RESULT: Thus the program has been successfully executed.

| Exp No: 16 | To Create Checkbox, Choice Radio Button, Label And TextBox |
|---|---|
| 14.09.2023 | |

Aim:

To write a java program to create checkbox, choice radio button, Label and Textbox.

A. Algorithm:

Step 1: Start.
Step 2: Create checkbox.
Step 3: Add checkbox to applet window.
Step 4: Register listener for checkbox.
Step 5: When checkbox is selected or deselected, registered listener, recieve notification from the component.
Step 6: Appropriate event procedure is called and necessary action takes place
Step 7: Stop.

B. Algorithm:

Step 1: Start
Step 2: Create choice control.
Step 3: Add choice control to applet window.
Step 4: Register listener for choice control.
Step 5: When choice control is selected or deselected

Step 5: registered listener recives notification from the components.

Step 6: Appropriate event procedure is called and necessary action takes place.

Step 7: Stop.

C. ALGORITHM:

Step 1: Start.

Step 2: Create checkbox group.

Step 3: Add checkbox to it.

Step 4: Register listener for checkbox group.

Step 5: When choice control is selected or deselected, registered listener recieves notification from the component.

Step 6: Appropriate event procedure is called and necessary action takes place.

Step 7: Stop.

D. ALGORITHM:

Step 1: Start.

Step 2: Create label box and textbox.

Step 3: Add label box and textbox to applet window.

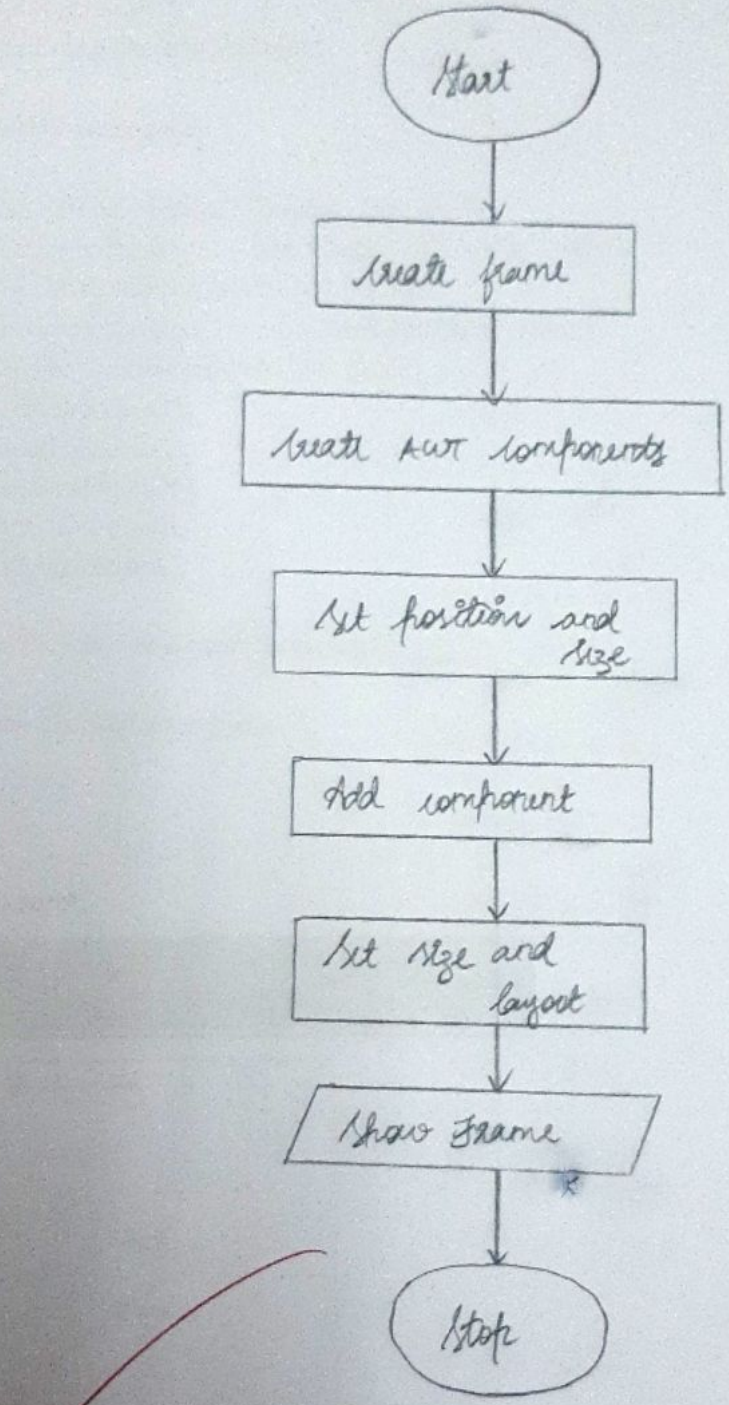
Step 4: Register listener for textbox.

Step 5: When users process interkey, registered listener receives notification from the component.

Step 6: Appropriate event procedure is called and necessary action takes place.

Step 7: Stop.

FLOWCHART :


Start
Create frame
Create AWT components
Set position and size
Add component
Set size and layout
Show Frame
Stop

# 16. AWT CONTROLS – CHECKBOX, CHECKBOXGROUP, LABEL, TEXTFIELD, CHOICE

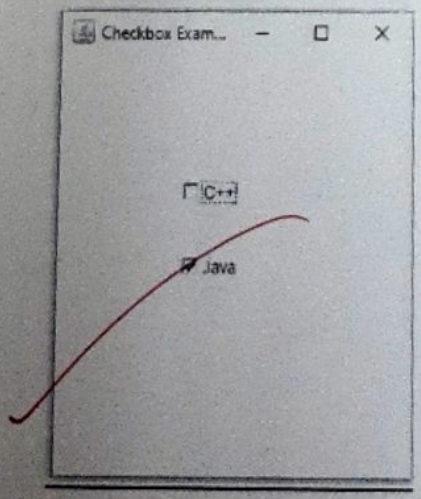
## CHECKBOX

```
import java.awt.*;
public class checkboxexample
{
checkboxexample()
{
Frame f= new Frame("Checkbox Example");
Checkbox checkbox1 = new Checkbox("C++");
checkbox1.setBounds(100,100, 50,50);
Checkbox checkbox2 = new Checkbox("Java", true);
checkbox2.setBounds(100,150, 50,50);
f.add(checkbox1);
f.add(checkbox2);
f.setSize(400,400);
f.setLayout(null);
f.setVisible(true);
}
public static void main(String args[])
{
new checkboxexample();
}
}
```

**Output:**

```
D:\ubai>javac checkboxexample.java

D:\ubai>java checkboxexample
```



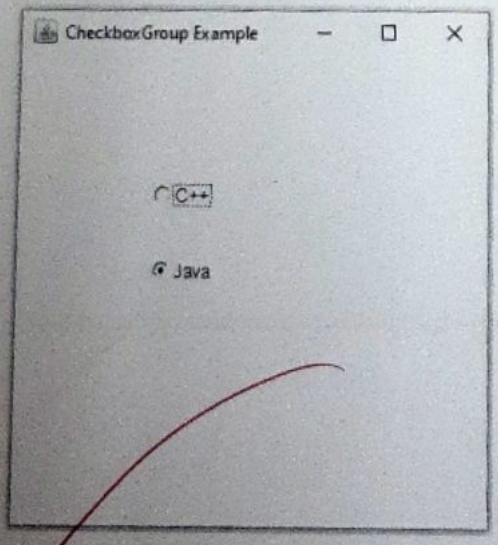
**CHECKBOXGROUP**

```
import java.awt.*;
public class checkboxgroupexample
{
checkboxgroupexample(){
Frame f= new Frame("CheckboxGroup Example");
CheckboxGroup cbg = new CheckboxGroup();
Checkbox checkBox1 = new Checkbox("C++", cbg, false);
checkBox1.setBounds(100,100, 50,50);
Checkbox checkBox2 = new Checkbox("Java", cbg, true);
checkBox2.setBounds(100,150,  50,50);
f.add(checkBox1);
f.add(checkBox2);
f.setSize(400,400);
f.setLayout(null);
f.setVisible(true);
}
public static void main(String args[])
{
new checkboxgroupexample();
}
}
```

**Output:**

```
D:\ubai>javac checkboxgroupexample.java

D:\ubai>java checkboxgroupexample
```



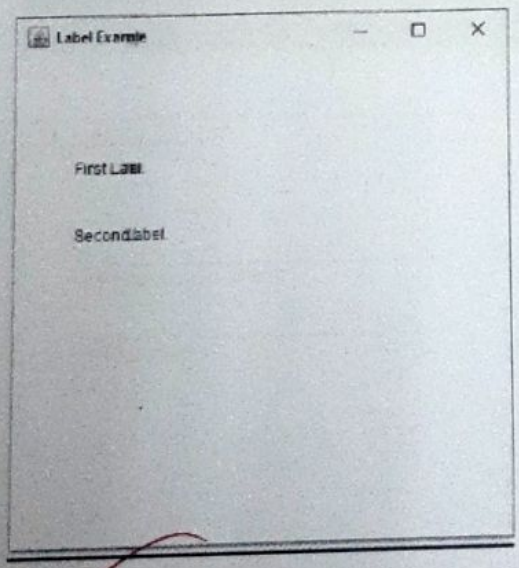
**LABEL**

```
import java.awt.*;
class labelexample{
public static void main(String args[]){
Frame f= new Frame("Label Example");
Label l1,l2;
l1=new Label("First Label.");
l1.setBounds(50,100, 100,30);
l2=new Label("Second Label.");
l2.setBounds(50,150, 100,30);
f.add(l1);
f.add(l2);
f.setSize(400,400);
f.setLayout(null);
f.setVisible(true);
}
}
```

**Output:**

```
D:\ubai> avac labelexample.java

D:\ubai> ava labelexample
```



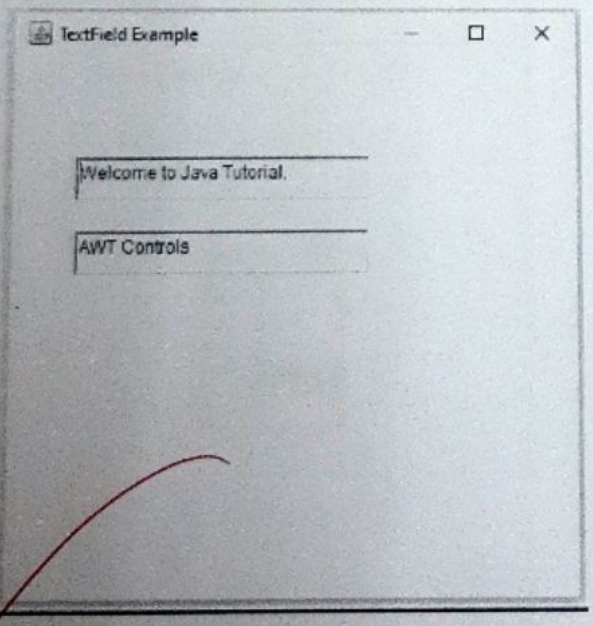
**TEXTFIELD**

```
import java.awt.*;
class textfieldexample
{
public static void main(String args[])
{
Frame f= new Frame("TextField Example");
TextField t1,t2;
t1=new TextField("Welcome to Java Tutorial.");
t1.setBounds(50,100, 200,30);
t2=new TextField("AWT Controls");
t2.setBounds(50,150, 200,30);
f.add(t1);
f.add(t2);
f.setSize(400,400);
f.setLayout(null);
f.setVisible(true);
}
}
```

**Output:**

```
D:\ubai>javac textfieldexample.java

D:\ubai>java textfieldexample
```



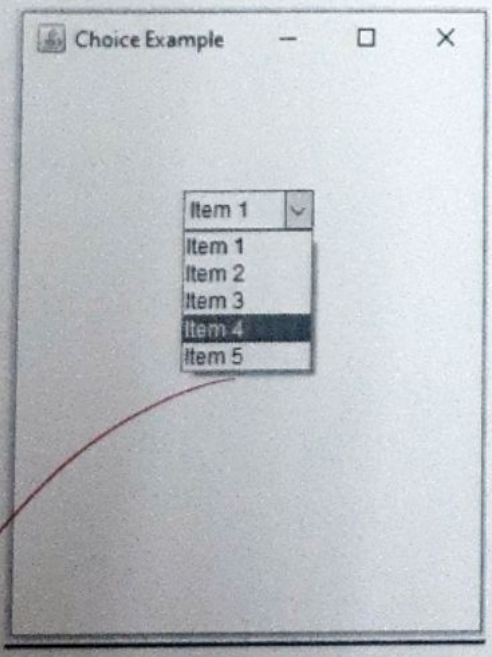
**CHOICE**

```
import java.awt.*;
public class choiceexample
{
public static void main(String args[])
{
Frame f= new Frame("Choice Example");
Choice c=new Choice();
c.setBounds(100,100, 75,75);
c.add("Item 1");
c.add("Item 2");
c.add("Item 3");
c.add("Item 4");
c.add("Item 5");
f.add(c);
f.setSize(400,400);
f.setLayout(null);
f.setVisible(true);
}
}
```

**Output:**

```
D:\ubai>javac choiceexample.java

D:\ubai>java choiceexample
```



RESULT:

Thus the program has been successfully executed.

Exp No: 17

20.09.2023

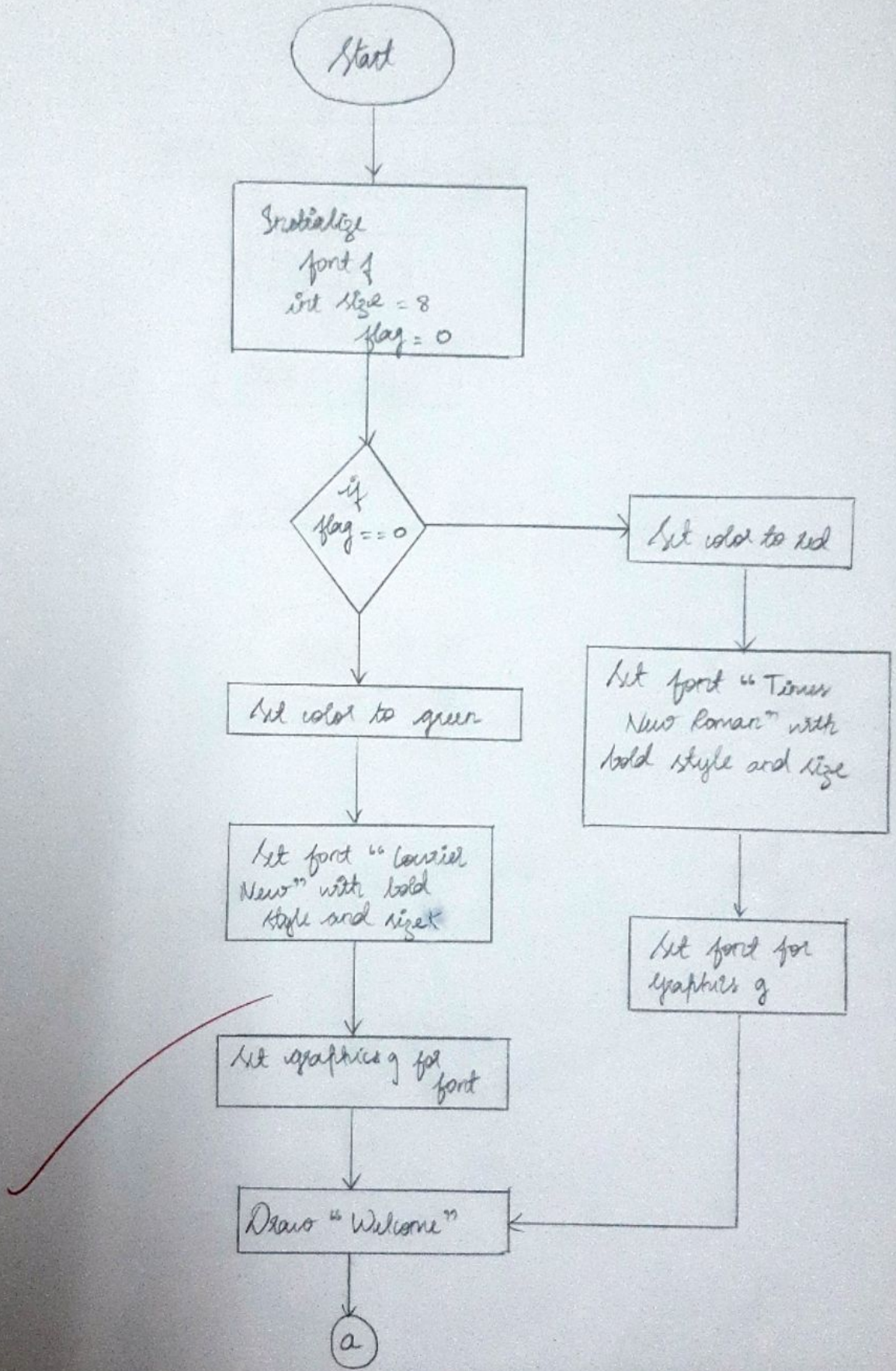
# Change Font And Color

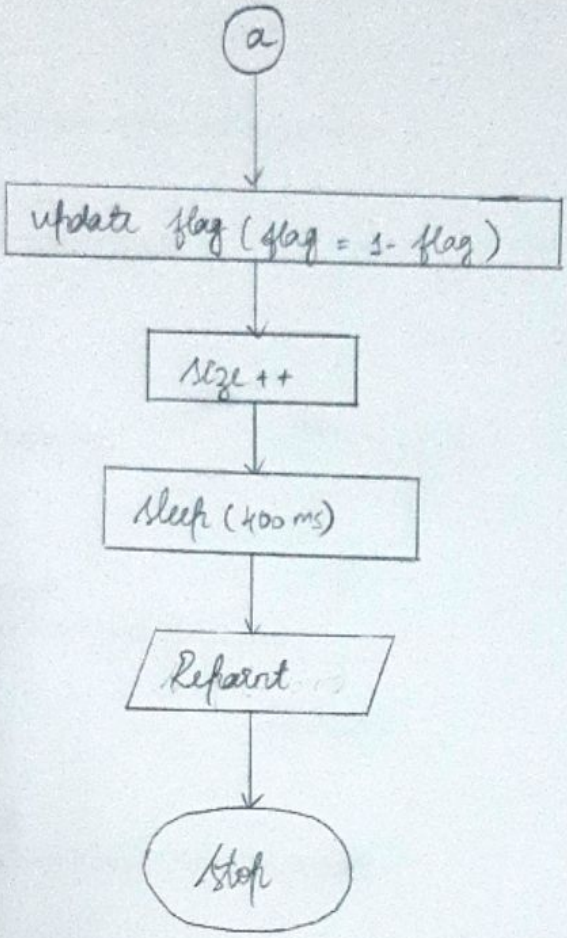
AIM: To write an applet program to change font and color.

ALGORITHM:

Step 1: Start.
Step 2: Set color to
Step 3: Set font to "
Step 4: Display the text "Welcome" in applet window.
Step 5: Set a time delay of 500ms.
Step 6: Change the color to red.
Step 7: Change the font to "TIMES NEW ROMAN"
Step 8: Display the text "WELCOME" with new color and font.
Step 9: Go to step 2.
Step 10: Stop.

FLOWCHART:


Start
Initialize
font f
int size = 8
flag = 0
if
flag == 0
Set color to red
Set color to green
Set font "Times New Roman" with bold style and size
Set font "Courier New" with bold style and size
Set font for graphics g
Set graphics g for font
Draw "Welcome"
a


a
update flag (flag = 1 - flag)
size ++
Sleep (400 ms)
Repaint
Stop

## 17. FONT STYLE AND DIFFERENT COLORS

```
import java.awt.*;
import java.applet.*;
/*
<applet code="fontsize.java" width=500 height=450>
</applet>
*/
public class fontsize extends Applet
{
Font f;
int size=8;
int flag=0;
public void paint(Graphics g)
{
if(flag==0)
{
g.setColor(Color.green);
f=new Font("Courier New",Font.BOLD,size);
g.setFont(f);
}
else
{
g.setColor(Color.red);
f=new Font("Times New Roman",Font.BOLD,size);
g.setFont(f);
}
g.drawString("Welcome",0,300);
flag=1-flag;
size++;
try
{
Thread.sleep(400);
}
catch(Exception e)
{
}
repaint();
}
}
```

**Output:**

```
D:\ubai>javac fontsize.java
Note: fontsize.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

D:\ubai>appletviewer fontsize.java
Warning: Applet API and AppletViewer are deprecated.
```

RESULT

Thus the program has been successfully executed.

Exp No: 18

20.09.2023

# Panel And Layout



AIM:

To write a java program to work with panel and layout.

ALGORITHM:

Step 1 : Start.

Step 2 : Create a class border layout example to set layout.

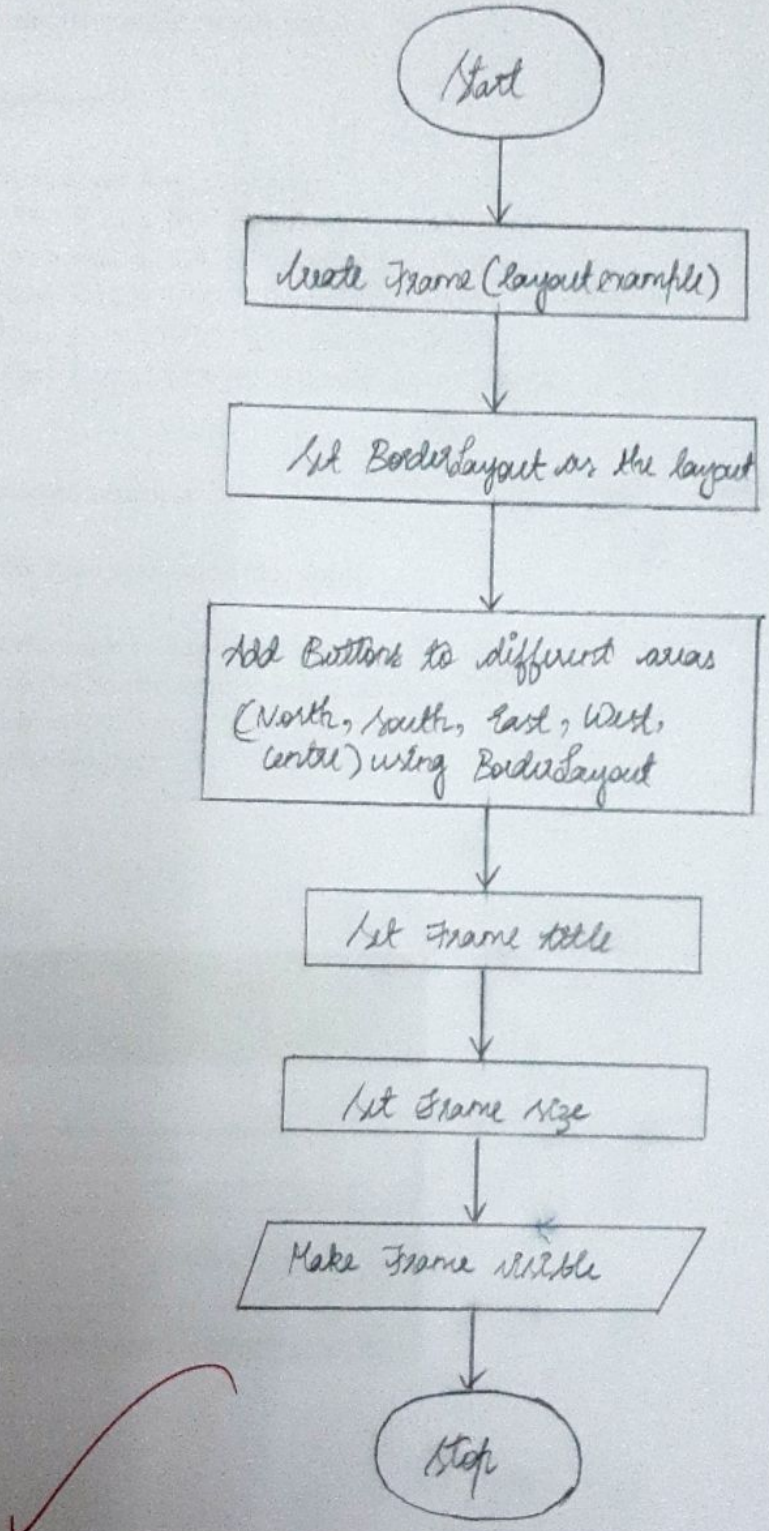
Step 3 : Add the button to display the border layout NORTH, SOUTH, WEST, EAST, CENTER.

Step 4 : Create an object frame to set title, size and visibility of the panel and layout.

Step 5 : Stop.

18

FLOWCHART:



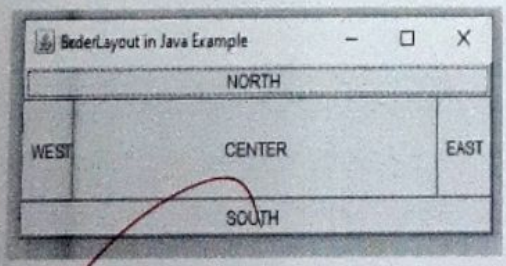
# 18. PANELS AND LAYOUTS

```
import java.awt.*;
class layoutexample extends Frame
{
 layoutexample()
 {
 setLayout(new BorderLayout());
 add(new Button("NORTH"),BorderLayout.NORTH);
 add(new Button("SOUTH"),BorderLayout.SOUTH);
 add(new Button("EAST"),BorderLayout.EAST);
 add(new Button("WEST"),BorderLayout.WEST);
 add(new Button("CENTER"),BorderLayout.CENTER);
 }
}
 classborderexample
 {
 public static void main(String args[])
 {
layoutexample f= new layoutexample();
f.setTitle("BorderLayout in Java Example");
f.setSize(400,150);
f.setVisible(true);
 }
}
```

**Output:**

```
D:\...ai>javac borderexample.java

D:\...ai>java borderexample
```



RESULT:

Thus the program has been successfully executed.