Ex. No. : 1 **JAVA INPUT AND OUTPUT**

Date:

Aim: To create a java program to display name of the institution, year of establishment,department and your class.

Algorithm:

Step 1: Start the Program

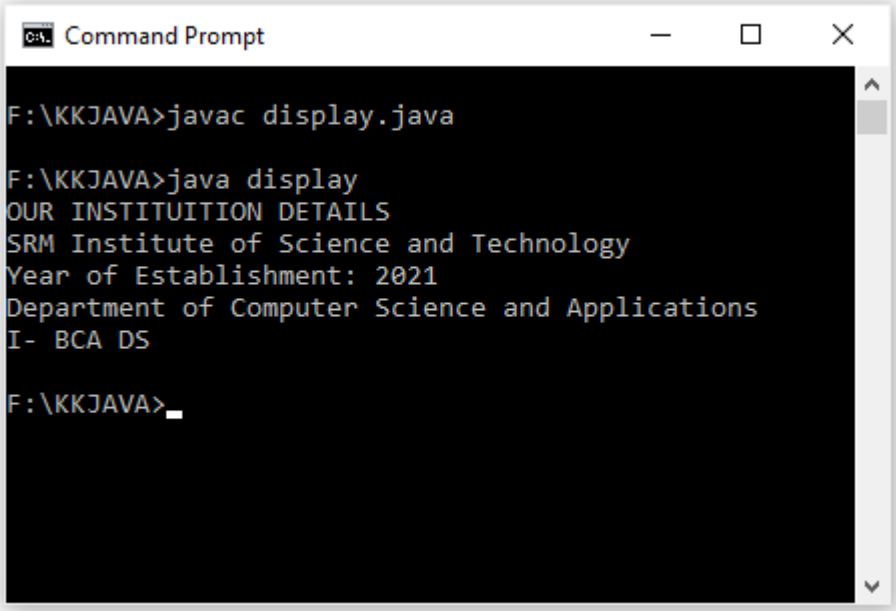
Step2: Define a class and display name of the institution, year of establishment, department and your class using Output method

Step3: Stop the program.

Program:

```
import java.io.*;
class display
{
public static void main(String args[])
{
int year=2021;
System.out.println("OUR INSTITUITION DETAILS");
System.out.println("SRM Institute of Science and Technology");
System.out.println("Year of Establishment: " + year);
System.out.println("Department of Computer Science and Applications");
System.out.println("I- BCA DS");
}
}
```

OUTPUT


Command Prompt
F:\KKJAVA>javac display.java
F:\KKJAVA>java display
OUR INSTITUITION DETAILS
SRM Institute of Science and Technology
Year of Establishment: 2021
Department of Computer Science and Applications
I- BCA DS
F:\KKJAVA>

Ex. No. :2 **TYPE CONVERSION**

Date:

Aim: To create a java program to do type conversion.

Algorithm:

Step 1: Start the program.

Step 2: Define a class and its methods.

Step 3: Declare and initialize variables ; i=100.

Step 4: Perform type conversion like long and float

Long l=i, float f=I;

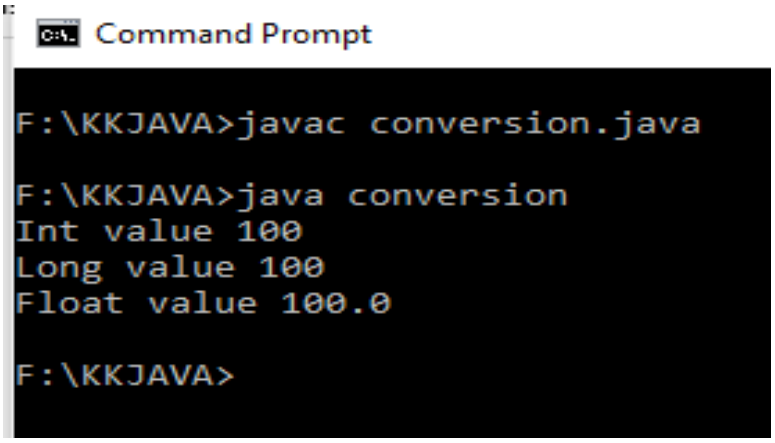
Step 5: Display the result.

Step 6: Stop the program.

Program / Source Code:

```
// Automatic type conversion program

import java.io.*;
class conversion
 {
    public static void main(String[] args)
   {
      int i = 100;

     // Integer to long type
      long l = i;

       // long to float type
      float f = l;

      // Print and display commands
      System.out.println("Int value " + i);
      System.out.println("Long value " + l);
      System.out.println("Float value " + f);
   }
}
```

OUTPUT:


Command Prompt
F:\KKJAVA>javac conversion.java
F:\KKJAVA>java conversion
Int value 100
Long value 100
Float value 100.0
F:\KKJAVA>

Ex. No. : 3 JAVA OPERATORS
Date:

Aim: To create a Java program to implement Operators .

Algorithm:

Step 1: Start the Program

Step 2: Define a class and declare variables like a,b,add, sub, mul, div, moddiv;

Step 3 : assign value to the variables a=10, b=20;

Step 4: perform arithmetic operations add=a+b, sub= b-a, mul=a*b, div=a/2, moddiv=a%3

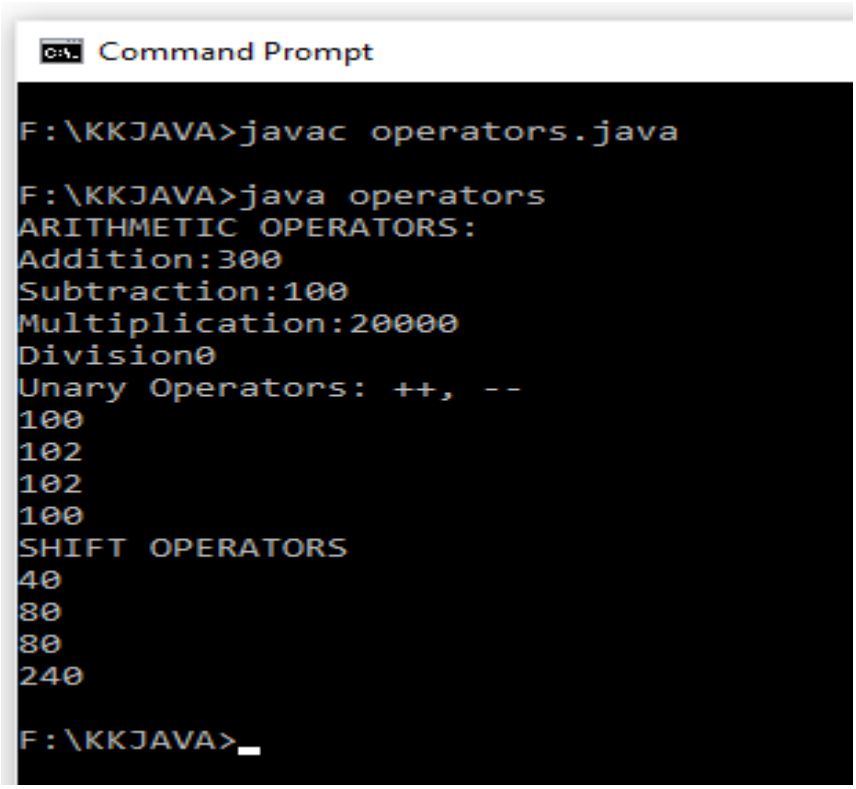
Step 5: Display the results

Step3: Stop the program.

Program :

```
import java.io.*;
class operators
{
public static void main(String args[])
{
        int a=100,b=200,add, sub, mul,div;
        add=a+b;
        sub=b-a;
        mul=a*b;
        div=a/b;
        System.out.println("ARITHMETIC OPERATORS:");
        System.out.println("Addition:" + add);
        System.out.println("Subtraction:"+ sub);
        System.out.println("Multiplication:"+ mul);
        System.out.println("Division"+ div);
        System.out.println("Unary Operators: ++, --");
        System.out.println(a++);
        System.out.println(++a);
        System.out.println(a--);
        System.out.println(--a);
        System.out.println("SHIFT OPERATORS");
        System.out.println(10<<2);
        System.out.println(10<<3);
        System.out.println(20<<2);
        System.out.println(15<<4);
}}
```

OUTPUT


Command Prompt
F:\KKJAVA>javac operators.java
F:\KKJAVA>java operators
ARITHMETIC OPERATORS:
Addition:300
Subtraction:100
Multiplication:20000
Division0
Unary Operators: ++, --
100
102
102
100
SHIFT OPERATORS
40
80
80
240
F:\KKJAVA>

Ex. No. :4 **LOOPING STATMENTS**

Date:

Aim: To create a Java program to implement looping statements

Algorithm:

Step 1: Start the Program

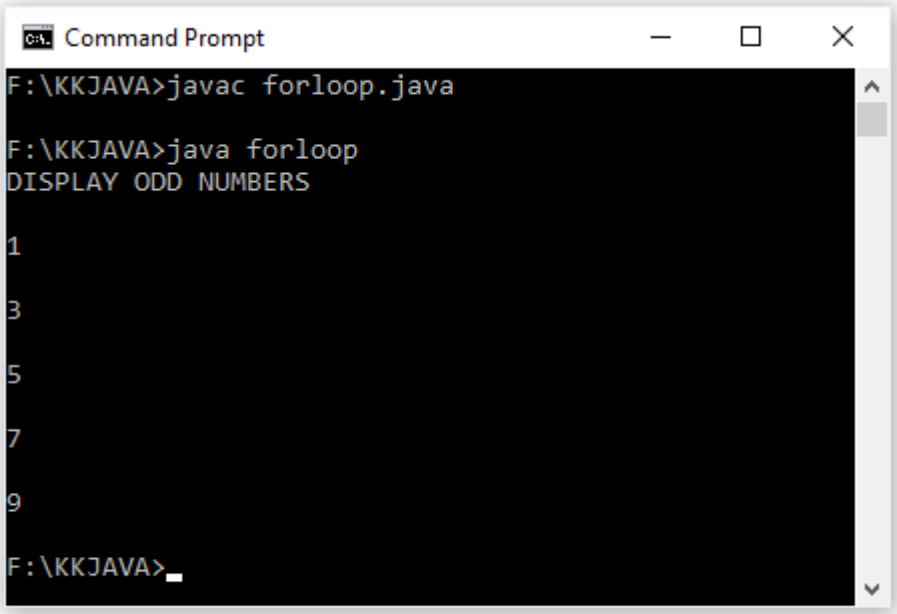
Step2: Define a class to display odd numbers using for-loop

Step3: Stop the program.

Program:

```
// FOR-LOOP : DISPLAY ODD NUMBER
import java.io.*;
public class forloop
{
public static void main(String args[ ])
{
System.out.println(“ DISPLAY ODD NUMBERS");
for(int i=1;i<=10;i=i+2)
  {
System.out.println("\n"+i);
   }
}
 }
```

OUTPUT


Command Prompt
F:\KKJAVA>javac forloop.java
F:\KKJAVA>java forloop
DISPLAY ODD NUMBERS
1
3
5
7
9
F:\KKJAVA>

Ex. No. :5 WHILE STATMENT

Date:

Aim: To write a java program to implement while-looping statements

Algorithm:

Step 1: Start the Program

Step2: Define a class to find sum of the series 1+2+3….+10 using while-loop.

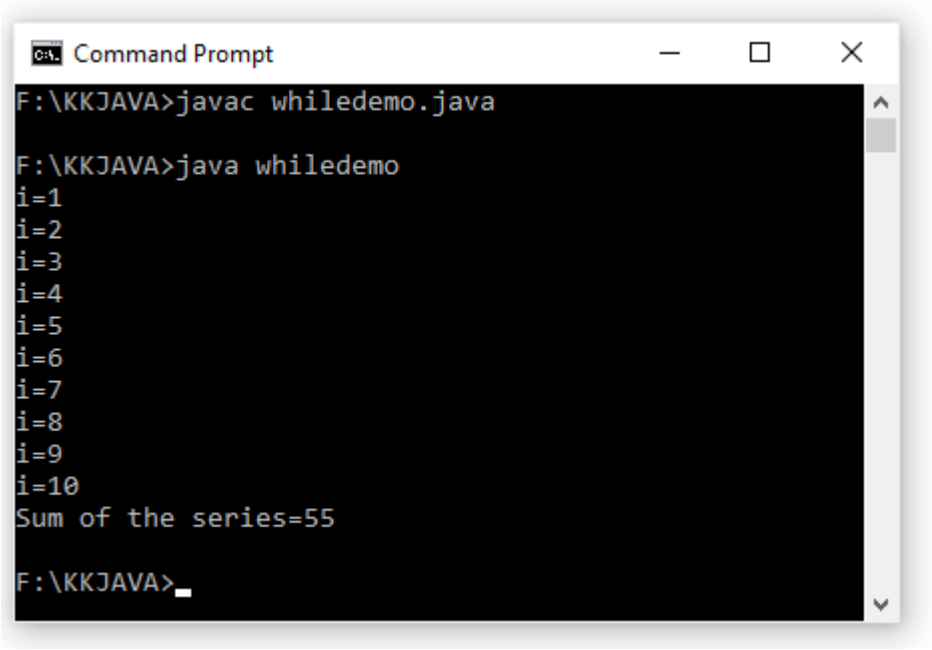
Step3: Stop the program

Program:

WHILE STATEMENT: sum of the series 1+2+3+4+5+6…+10

```
import import java.io.*;
class whiledemo
{
public static void main(String args[] )
{
int i=1, sum=0;
while(i<=10)
  {
System.out.println("i="+i);
sum=sum+i;
i++;
}
System.out.println("Sum of the series="+sum);
 }

}
```

OUTPUT


Command Prompt
F:\KKJAVA>javac whiledemo.java
F:\KKJAVA>java whiledemo
i=1
i=2
i=3
i=4
i=5
i=6
i=7
i=8
i=9
i=10
Sum of the series=55
F:\KKJAVA>

Ex. No. :6 DO-WHILE STATMENT

Date:

Aim: To write a java program to implement Do-while looping statements

Algorithm:

Step 1: Start the Program

Step2: Define a class to Print the even numbers from 2 to 20 using D-while-loop.

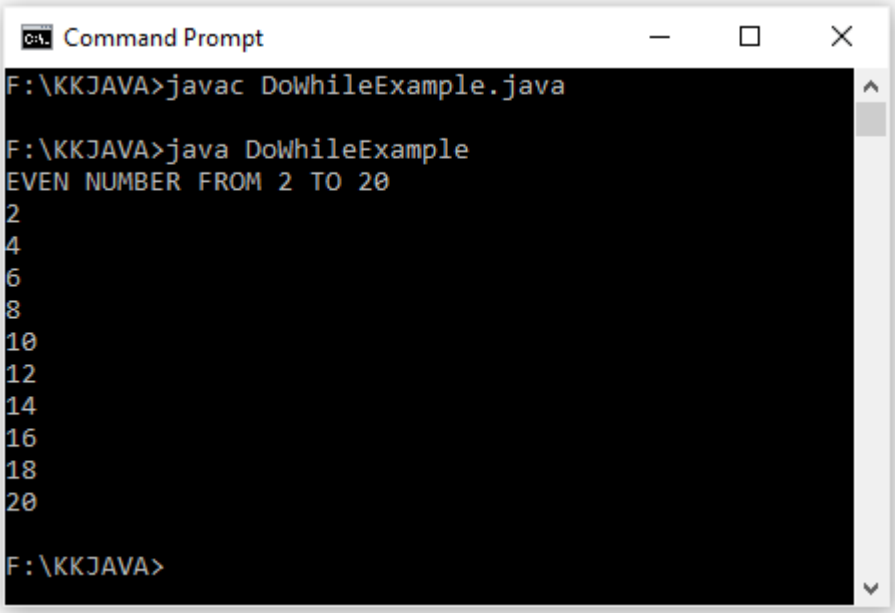
Step3: Stop the program

Program:

// DoWhileExample.java to Print the even numbers from 2 to 20

```
public class DoWhileExample
 {
public static void main(String args[] )
{
   int i=2;
   System.out.println("EVEN NUMBER FROM 2 TO 20");
   do
  {
    System.out.println(i);
      i=i+2;
   }
   while(i<=20);
  }
}
```

OUTPUT:


Command Prompt
F:\KKJAVA>javac DoWhileExample.java
F:\KKJAVA>java DoWhileExample
EVEN NUMBER FROM 2 TO 20
2
4
6
8
10
12
14
16
18
20
F:\KKJAVA>

Ex. No. 7: **ARRAYS**

Date:

Aim : To implement array concept using Java

Algorithm:

Step 1: Start the program

Step 2: Create a class

Step 3: Declare 2D array variables

Step 5: Declare methods with for loop to give input

Step 6: Perform Matrix Addition

Step 7: Display the output

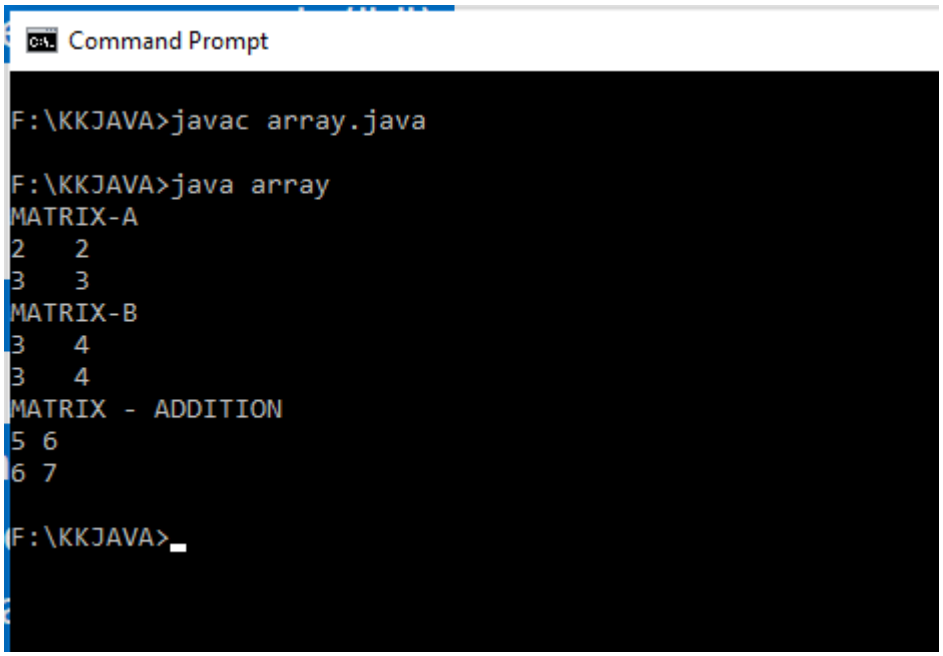
Step 8: Stop the program

Program:

```
//MATRIX ADDITION
import java.io.*;

class array
{

inta[][]= new int[2][2];
intb[][]=new int[2][2];
intc[][]=new int[2][2];
inti,j;

void get()
{
System.out.println("MATRIX-A");
for(i=0;i<a.length;i++)
{
for(j=0;j<a.length;j++)
{
a[i][j]=i+2;
System.out.print(a[i][j]+"   ");
}
System.out.println(" ");
}
System.out.println("MATRIX-B");
for(i=0;i<b.length;i++)
{
for(j=0;j<b.length;j++)
{
b[i][j]=j+3;
System.out.print(b[i][j]+"   ");
}
System.out.println(" ");
}
}

void cal()
 {
for(i=0;i<a.length;i++)
for(j=0;j<b.length;j++)
{
```

```
c[i][j]=a[i][j]+b[i][j];
}
}

void disp()
{
System.out.println("MATRIX - ADDITION");
for(i=0;i<a.length;i++)
{
for(j=0;j<b.length;j++)
{
System.out.print(c[i][j]+" ");
}
System.out.println(" ");
}
}

public static void main(String args[])
{
 array ar=new array();
ar.get();
ar.cal();
ar.disp();

}
}
```

OUTPUT


Command Prompt
F:\KKJAVA>javac array.java
F:\KKJAVA>java array
MATRIX-A
2 2
3 3
MATRIX-B
3 4
3 4
MATRIX - ADDITION
5 6
6 7
F:\KKJAVA>

SWITCH-CASE

Ex. No. :8
Date:

Aim : To implement switch-case statement using Java.

Algorithm:

Step 1: Start the program

Step 2: Read days of a week

Step 3: Create switch with multiple statements

Step 4: Enter your choice

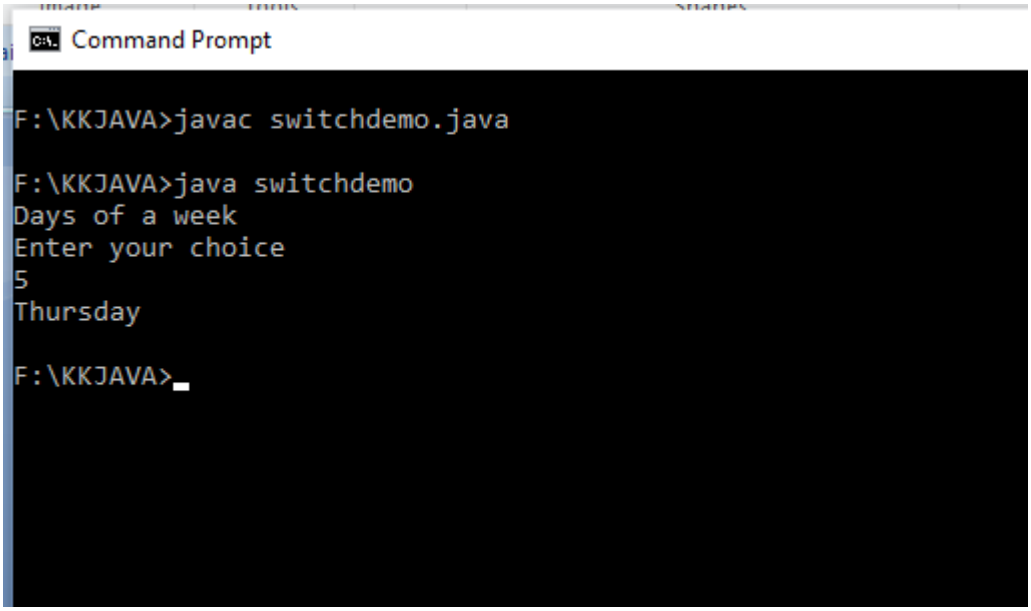
Step 5: Display the day as output

Step 6: stop the program.

Program:

```
importjava.io.*;
class switchdemo
{
public static void main(String args[])
{
int day;
Scanner console=new Scanner(System.in);
System.out.println("Days of a week");
System.out.println("Enter your choice");
day=console.nextInt();
switch(day)
{
case 1:
        System.out.println("Sunday");
        break;
case 2:
        System.out.println("Monday");
        break;
case 3:
        System.out.println("Tuesday");
        break;
case 4:
        System.out.println("Wednesday");
        break;
case 5:
        System.out.println("Thursday");
        break;
case 6:
        System.out.println("Friday");
        break;
case 7:
        System.out.println("Saturday");
        break;
}
}
}
```

OUTPUT


Command Prompt
F:\KKJAVA>javac switchdemo.java
F:\KKJAVA>java switchdemo
Days of a week
Enter your choice
5
Thursday
F:\KKJAVA>

COMMAND LINE ARGUEMENT AND STRING FUNCTIONS

Ex. No. :9

Date:

Aim : To implement command line arguments and string functions using Java.

Algorithm:

Step 1: Start the program

Step 2: Declare variables with 3 data types int, float and string.

Step 3: Get 3 inputs from the command line during runtime as given below.

java cmdline 12 15.5 AMMU

Step 4: Convert numeric input to respective data types.

Step 5: Perform Calculation and display results of numeric input.

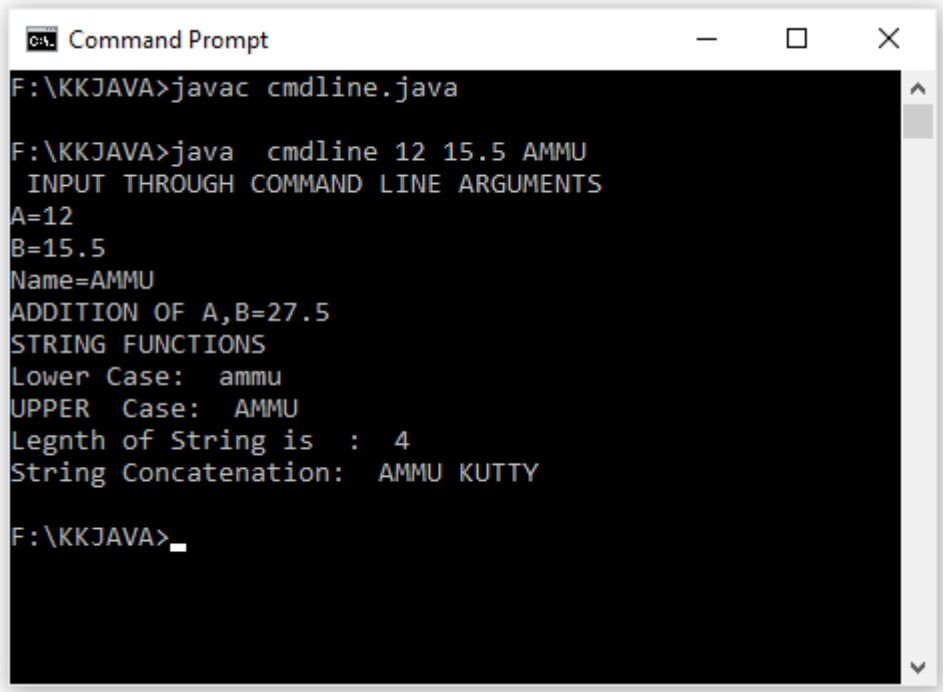
Step 6: Implement String functions on the given input string value
toLowerCase(), toUpperCase(), length() and concat().

Step 7: stop the program.

//PROGRAM : COMMAND LINE ARGUEMENT AND STRING FUNCTIONS

```
import java.io.*;
class cmdline
{
public static void main(String args[])  //args[0]=12 args[1]= 15.5 args[2]= AMMU
{
int a;
float b,c;
String s;
a=Integer.parseInt(args[0]);
b=Float.parseFloat(args[1]);
s=args[2];
c=a+b;
System.out.println(" INPUT THROUGH COMMAND LINE ARGUMENTS");
System.out.println("A="+a);
System.out.println("B="+b);
System.out.println("Name="+s);
System.out.println("ADDITION OF A,B=" + c);
System.out.println("STRING FUNCTIONS");
System.out.println("Lower Case:  "+ s.toLowerCase());
System.out.println("UPPER  Case:  "+ s.toUpperCase());
System.out.println("Legnth of String is  :  "+ s.length());
System.out.println("String Concatenation:  "+ s.concat(" KUTTY"));
}
}
```

OUTPUT:


Command Prompt
F:\KKJAVA>javac cmdline.java
F:\KKJAVA>java cmdline 12 15.5 AMMU
INPUT THROUGH COMMAND LINE ARGUMENTS
A=12
B=15.5
Name=AMMU
ADDITION OF A,B=27.5
STRING FUNCTIONS
Lower Case: ammu
UPPER Case: AMMU
Legnth of String is : 4
String Concatenation: AMMU KUTTY
F:\KKJAVA>

INHERITANCE

Ex. No. :10
Date:

Aim: To implement Inheritance concept using Java.

Algorithm:

Step 1: Start the program

Step 2: Create super class Room with methods

Step 3: Declare variable in the super class

Step 4: Create Sub class and extend the super class

Step 4: Create main class

Step 5: Initialise variables in the main class

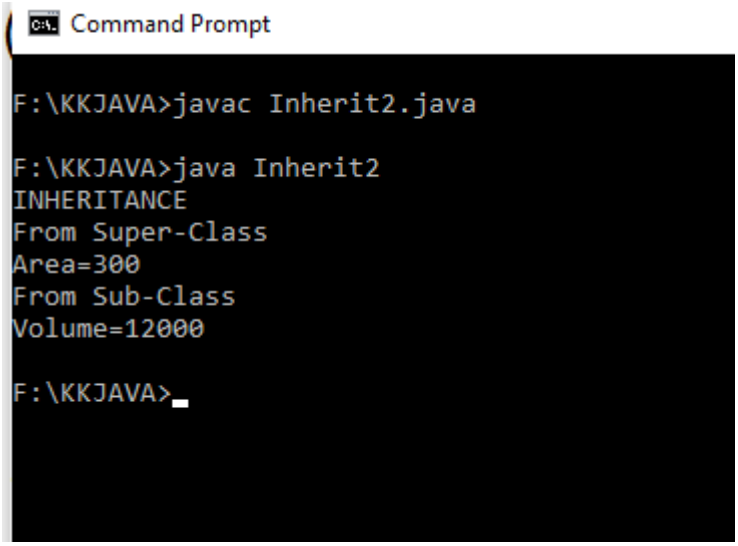
Step 6: Create object for subclass and call methods to perform calculation.

Step 7: Display the output

Program:

```
import java.io.*;
class Room
{
intlength,width;
Room(int x, int y)
{
length=x;
width=y;
}
intarea()
{
return(length*width);
}
}
class PoojaRoom extends Room
{
int height;
PoojaRoom(intx,int y, int z)
{
super(x,y);
height=z;
}
intvolume()
{
return(length*width*height);
}
}

public class Inherit2
{
public static void main(String args[])
{
PoojaRoom P=new PoojaRoom(10,30,40);
int A=P.area();
int V=P.volume();
System.out.println("INHERITANCE");
System.out.println("From Super-Class");
System.out.println("Area="+A);
System.out.println("From Sub-Class");
System.out.println("Volume="+V);
} }
```

OUTPUT


Command Prompt
F:\KKJAVA>javac Inherit2.java
F:\KKJAVA>java Inherit2
INHERITANCE
From Super-Class
Area=300
From Sub-Class
Volume=12000
F:\KKJAVA>

EXCEPTIONS

Ex. No. :11
Date:

Aim : To implement exceptions using Java.

Algorithm:

Step 1: Start the program

Step 2: Declare array A with size 3

Step 3: Take out array variables

Step 4: Display the output

Step 5: Stop the program

Step 6: Compile and run the program

Step 7: It show I/O exception array index out of bounds

Step 8: Stop the program

1. Division by Zero Exception

Program:

```
import java.io.*;
class exceptiondemo
{
public static void main(String args[])
{
int a=10,b=5,c=5;
int x=a/(b-c);
System.out.println("X="+ x);
int y=a/(b+c);
System.out.println("Y="+ y);
}
}
```

2. Array Index Out of Bounds Exception

```
import java.io.*;
class exceptiondemo2
{
public static void main(String args[])
{
inta[ ]={10,20,30};
System.out.println("A[0]="+a[0]);
System.out.println("A[1]="+a[1]);
System.out.println("A[2]="+a[2]);
System.out.println("A[3]="+a[3]);
}
}
```

OUTPUT-1

Command Prompt

```
F:\KKJAVA>javac exceptiondemo.java

F:\KKJAVA>java exceptiondemo
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at exceptiondemo.main(exceptiondemo.java:9)

F:\KKJAVA>
```

OUTPUT-2

Command Prompt

```
F:\KKJAVA>javac exceptiondemo2.java

F:\KKJAVA>java exceptiondemo2
A[0]=10
A[1]=20
A[2]=30
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 3
        at exceptiondemo2.main(exceptiondemo2.java:12)

F:\KKJAVA>
```

MULTI- THREADING

Ex. No. :12
Date:

Aim : To implement Multi-Threading using Java.

Algorithm:

Step 1: Start the program

Step 2: Create Three classes and Extend Thread

Step 3: Declare variables and assign values for each Thread

Step 4: Create main class and objects for each class

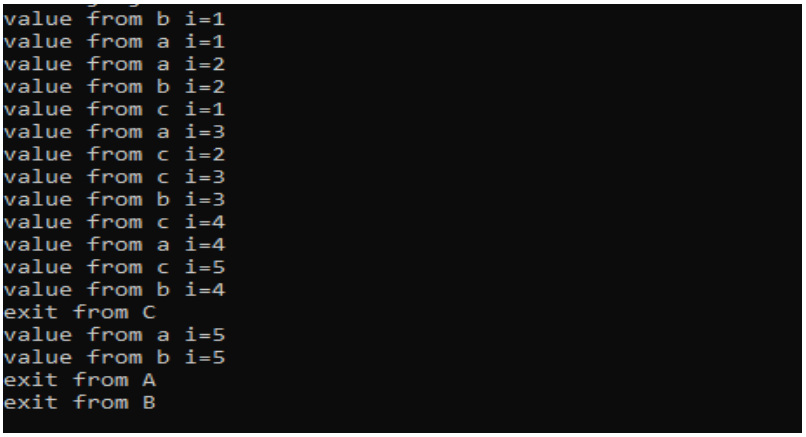
Step 5: Start and Run the Threads through their objects

Step 5: Stop the program

Program:

```
import java.io.*;
class A extends Thread
{
        public void run()
        {
        for(inti=1;i<=5;i++)
        {
        System.out.println("value from a i="+i);
        }
        System.out.println("exit from A");
        }
}
class B extends Thread
{
        public void run()
        {
        for(inti=1;i<=5;i++)
        {
        System.out.println("value from b i="+i);
        }
        System.out.println("exit from B");
        }
}
class C extends Thread
{
        public void run()
        {
        for(inti=1;i<=5;i++)
        {
        System.out.println("value from c i="+i);
        }
        System.out.println("exit from C");
        }
}
classthreaddemo
{
public static void main(String args[])
{
new A().start();
new B().start();
new C().start();
}
}
```

OUTPUT


value from b i=1
value from a i=1
value from a i=2
value from b i=2
value from c i=1
value from a i=3
value from c i=2
value from c i=3
value from b i=3
value from c i=4
value from a i=4
value from c i=5
value from b i=4
exit from C
value from a i=5
value from b i=5
exit from A
exit from B

PACKAGE

Ex. No. :13
Date:

Aim : To implement packages using Java.

Algorithm:

Step 1: Create a new folder p2

Step 2: Inside p2, create a sub folder P1 and class C2

Step 3: Create a package p1 in class C1 inside folder p1

Step 4: Class c2 as a main class which imports package p1

Step 5:Perform the operation

Step 6: Display the output

Step 7: Stop the program

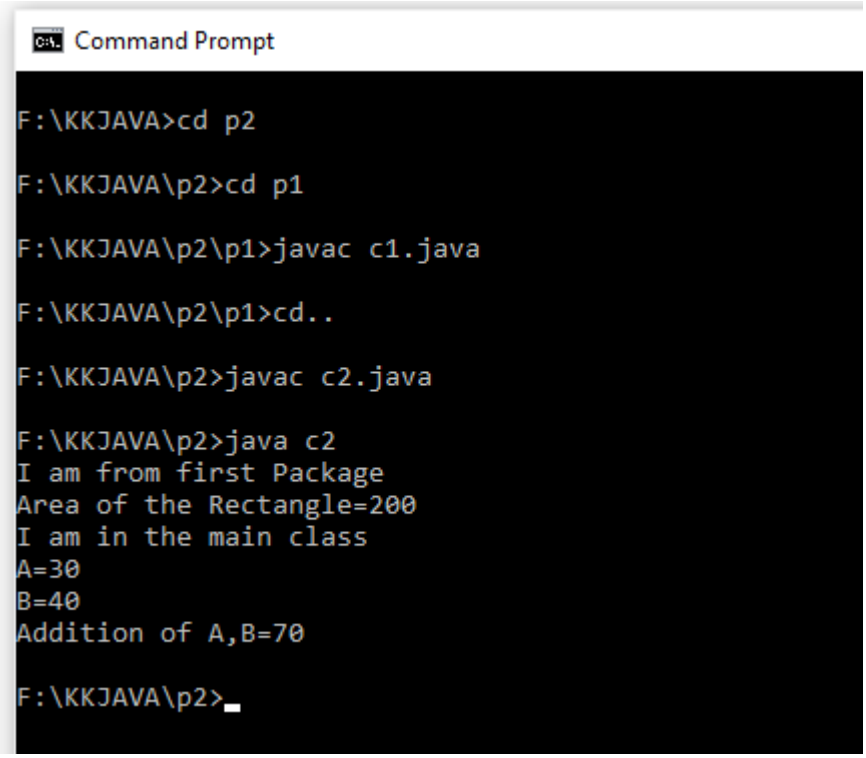
PROGRAM: PACKAGE P1\C1.JAVA

```
package p1;
public class c1
{
public void display()
{
int l=10;
int h=20;
System.out.println("I am from first Package");
int area=l*h;
System.out.println("Area of the Rectangle="+ area);
}
}
```

PROGRAM: FOLDER P2\C2.JAVA

```
import p1.*;
public class c2
{
public static void main(String args[])
{
c1 obj=new c1();
obj.display();
System.out.println("I am in the main class");
int a=30, b=40;
int c=a+b;
System.out.println("A=" +a);
System.out.println("B=" +b);
System.out.println("Addition of A,B=" +c);
}
}
```

OUTPUT


Command Prompt
F:\KKJAVA>cd p2
F:\KKJAVA\p2>cd p1
F:\KKJAVA\p2\p1>javac c1.java
F:\KKJAVA\p2\p1>cd..
F:\KKJAVA\p2>javac c2.java
F:\KKJAVA\p2>java c2
I am from first Package
Area of the Rectangle=200
I am in the main class
A=30
B=40
Addition of A,B=70
F:\KKJAVA\p2>

INTERFACE

Ex. No. :14
Date:

Aim : To implement interface using Java.

Algorithm:

Step 1: Start the program

Step 2: Create an interface I1with a variable.

Step 3: Create a class C1 and implement Interface

Step 4: Access the variable in interface I1 and use in main class to perform calculation.

Step 5: Create a main class with object to call the methods in C1.

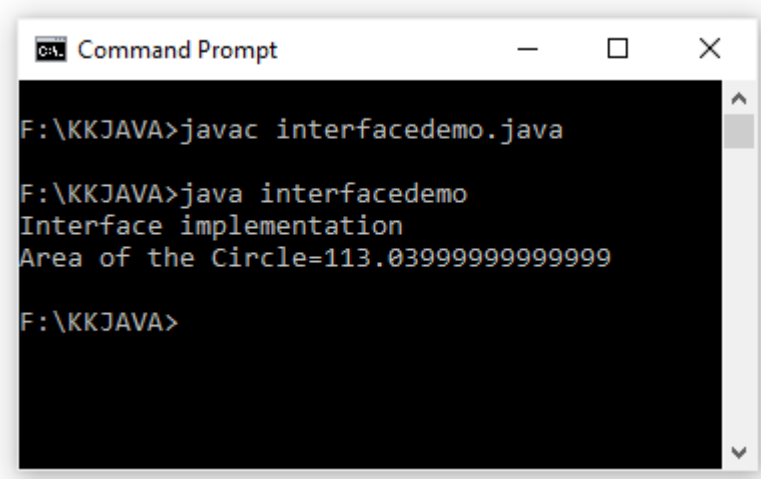
Step 6: Print the results.

Step 7: Stop the program

PROGRAM:

```
import java.io.*;

interface I1
{
int r=6;
}

class C1 implements I1
{
void display()
{
double pi=3.14;
System.out.println("Area of the Circle="+pi*r*r);
}
}
class interfacedemo
{
public static void main(String args[])
{
System.out.println("Interface implementation");
C1 obj=new C1();
obj.display();

}
}
```

OUTPUT


Command Prompt
F:\KKJAVA>javac interfacedemo.java
F:\KKJAVA>java interfacedemo
Interface implementation
Area of the Circle=113.03999999999999
F:\KKJAVA>

LAYOUT MANAGER (FLOW LAYOUT)

Ex. No.: 15

Date:

Aim: To create a flow layout with layout manager using Java.

Algorithm:

Step 1: Start the program

Step 2: Create a class and extend Frame

Step 3: Declare array S[]

Step 4: Create an object for Frame using Frame f= new Frame()

Step 5: Create Buttons and add buttons in flowlayout using FlowLayout()

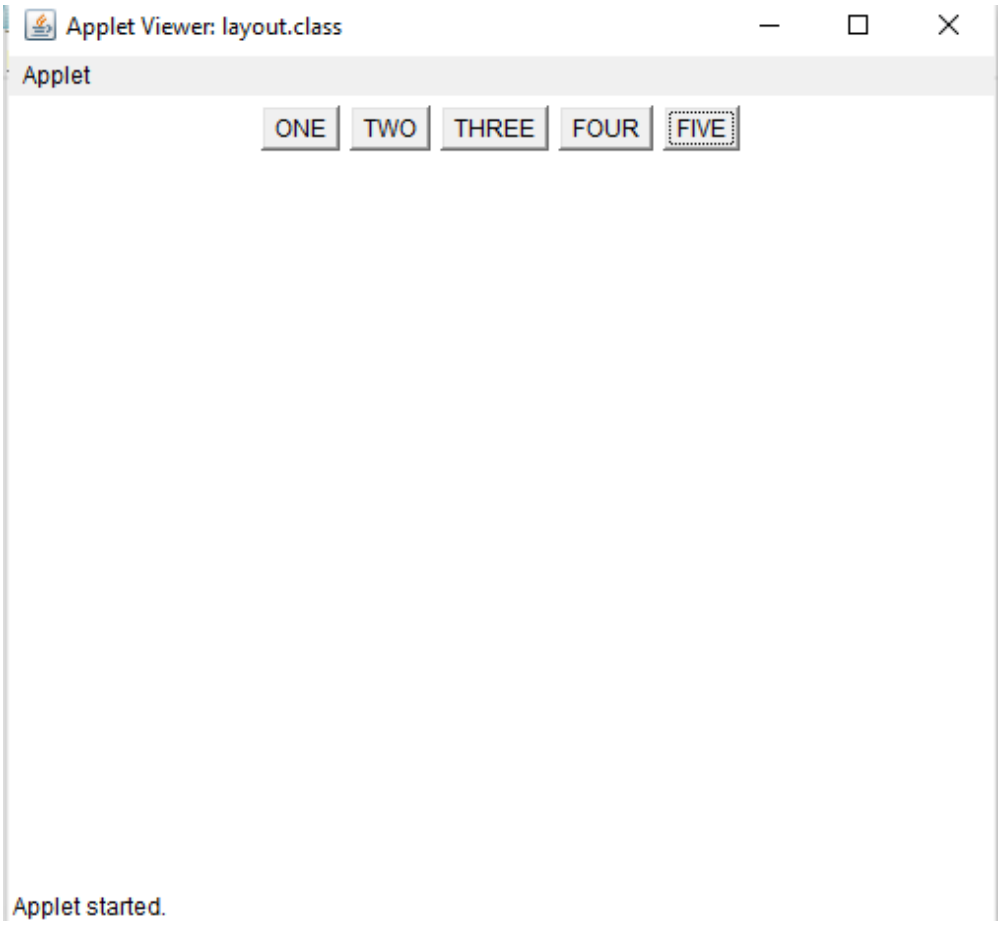
Step 6: Create a main class and create an object for Frame class and show layout.

Step 7: Stop the program.

Program:

```
import java.awt.*;
import java.applet.*;
public class layout extends Applet
{
String s[]={"ONE","TWO","THREE","FOUR","FIVE","SIX"};
public void init()
{
for(inti=0;i<5;i++)
{
setLayout(new FlowLayout());
add(new Button(s[i]));
}
}
}
//<applet code="layout.class" height=400 width=500></applet>
```

OUTPUT


Applet Viewer: layout.class
Applet
ONE
TWO
THREE
FOUR
FIVE
Applet started.

GRID LAYOUT

Ex. No.: 16

Date:

Aim: To create a grid layout with layout manager using Java.

Algorithm:

Step 1: Start the program

Step 2: Create a class and extend Applet

Step 3: Declare init( )method

Step 4: Create Buttons and add buttons in Grid layout using

GridLayout()

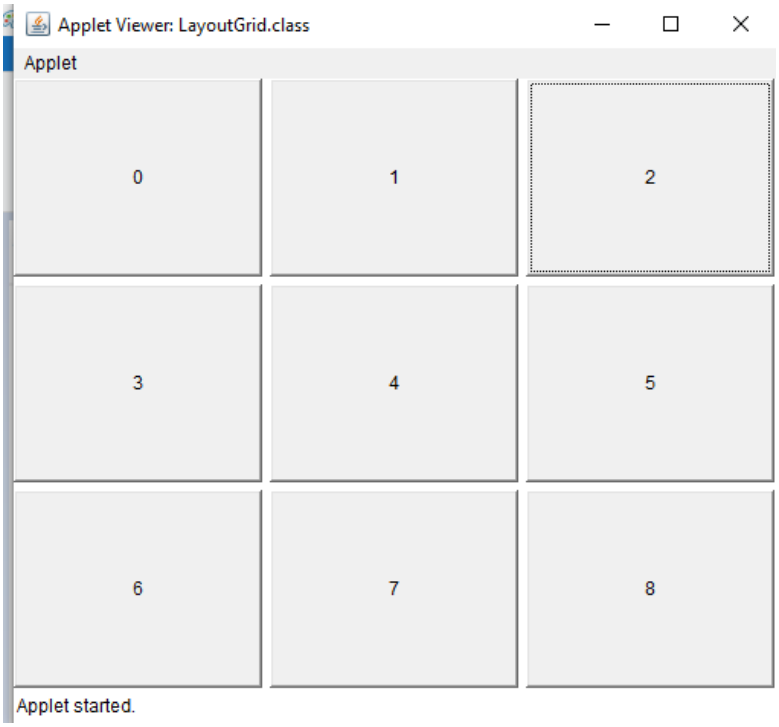
Step 5: Create a main class and create an object for the applet class and show layout.

Step 6: Stop the program.

Program:

```
import java.awt.*;
import java.applet.*;
public class LayoutGrid extends Applet
{
public void init()
{
for(inti=0;i<9;i++)
{
setLayout(new GridLayout(3,3,5,5));
add(new Button(""+i));
}

}
}



//<applet code="LayoutGrid.class" height=400 width=500></applet>
```

OUTPUT


Applet Viewer: LayoutGrid.class
Applet
0
1
2
3
4
5
6
7
8
Applet started.

BORDER LAYOUT

Ex. No.:17

Date:

Algorithm:

Step 1: Start the program

Step 2: Create class and extend Frame

Step 3: Create an object for Frame using Frame f= new Frame()

Step 4: Create Buttons and add buttons in layout usingBorderLayout() method

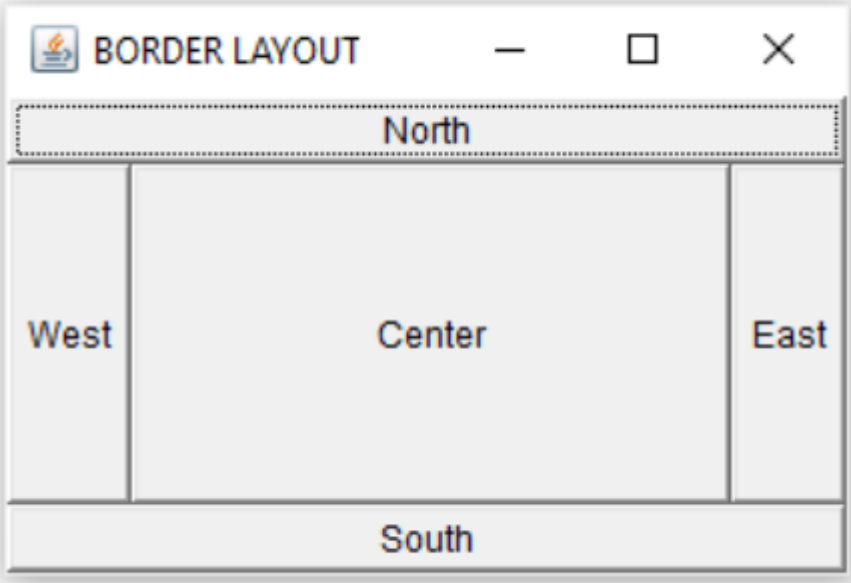
Step 5: Create a main class and create an object for Frame class and show layout.

Step 6: Stop the program

Program:

```
import java.awt.Frame;
import java.awt.Button;
import java.awt.BorderLayout;
public class LayoutBorder extends Frame
{
public LayoutBorder()
{
Frame f=new Frame("BORDERLAYOUT");
Button b1=new Button("North");
Button b2=new Button("South");
Button b3=new Button("East");
Button b4=new Button("West");
Button b5=new Button("Center");

f.add(b1,BorderLayout.NORTH);
f.add(b2,BorderLayout.SOUTH);
f.add(b3,BorderLayout.EAST);
f.add(b4,BorderLayout.WEST);
f.add(b5,BorderLayout.CENTER);
f.setSize(300,200);
f.setVisible(true);
}
public static void main(String args[])
{
LayoutBorder LB =new LayoutBorder();
}
}
```

OUTPUT



APPLET AND SHAPES

Ex. No.: 18

Date:

Aim : To create a java program to create shapes in an applet

Algorithm:

Step 1: Start the program

Step 2: Create class and extend Applet

Step 4: Create paint( ) method to draw different shapes in an applet

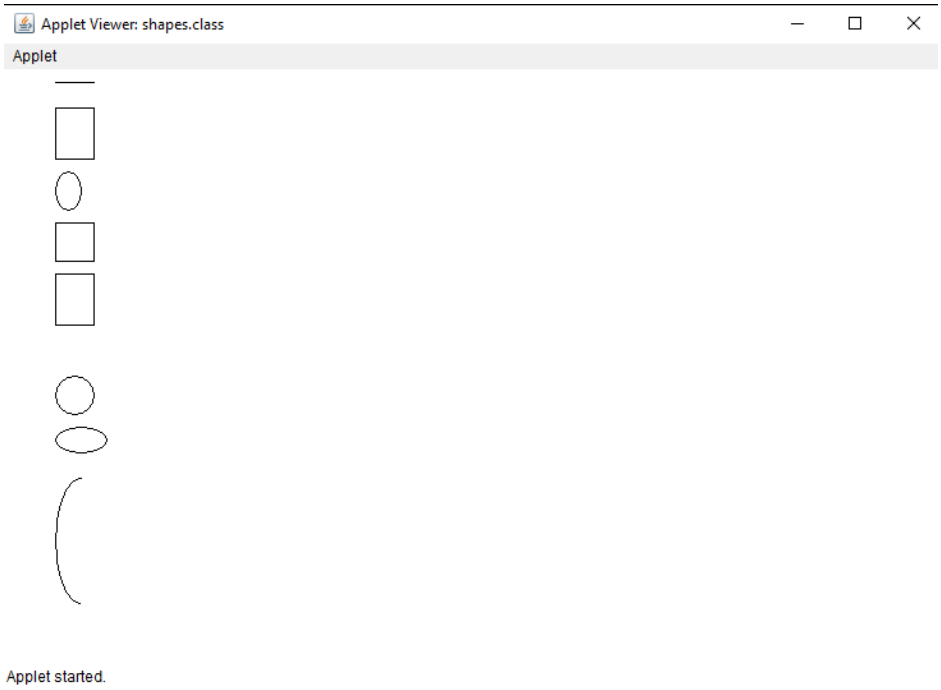
Step 5: Create a main class to pain the shapes

Step 6: Stop the program

Program:

```
import java.awt.*;
import java.applet.*;
public class shapes extends Applet
{
public void paint(Graphics g)
{
g.drawLine(40,10,70,10);
g.drawRect(40,30,30,40);
g.drawRoundRect(40,80,20,30,40,50);
g.drawRect(40,120,30,30); // Square
g.drawRect(40,160,30,40); // Rectangle
g.drawOval(40,240,30,30); //Circle
g.drawOval(40,280,40,20);  //Ellipse
g.drawArc(40,320,40,98,90,180);
}
}

//<applet code=shapes.class width=200 height=200></applet>
```

OUTPUT



APPLET AND FILL SHAPES

Ex. No.: 19

Date:

Aim : To create a Java program to fill shapes in an applet

Algorithm:

Step 1: Start the program

Step 2: Create class and extend Applet

Step 4: Create paint( ) method to draw different shapes and fill method in an applet

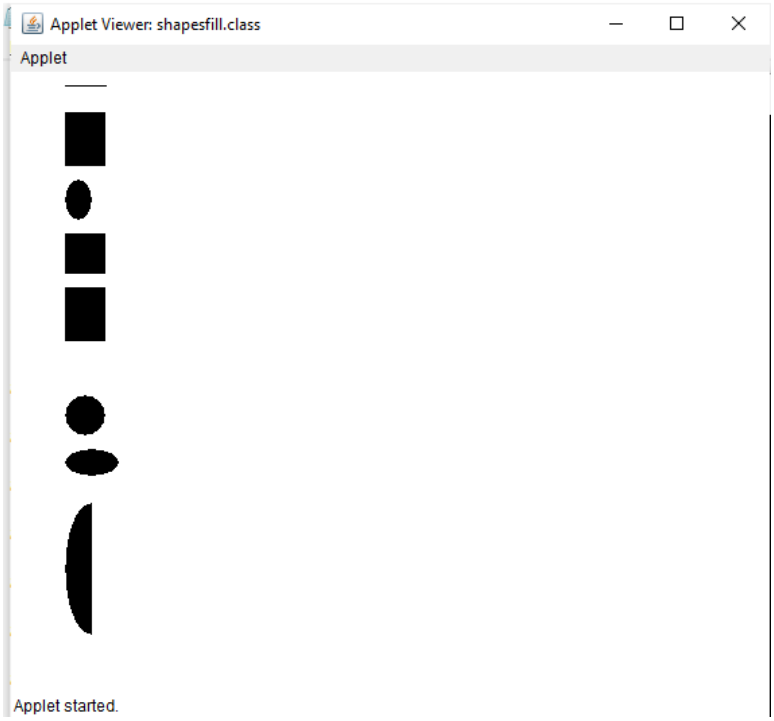
Step 5: Create a main class to pain the shapes

Step 6: Stop the program

Program:

```
import java.awt.*;
import java.applet.*;
public class shapesfill extends Applet
{
public void paint(Graphics g)
{
g.drawLine(40,10,70,10);
g.fillRect(40,30,30,40);
g.fillRoundRect(40,80,20,30,40,50);
g.fillRect(40,120,30,30);
g.fillRect(40,160,30,40);
g.fillOval(40,240,30,30); //Circle
g.fillOval(40,280,40,20);  //Ellipse
g.fillArc(40,320,40,98,90,180);
}
}

//<applet code=shapesfill.class width=200 height=200></applet>
```

OUTPUT



GUI COMPONENTS

Ex. No.: 20
Date:

Aim: To create a Java program to create and add components in an applet.

Algorithm:

Step 1: Start the program

Step 2: Create class which extends applet

Step 3: Create components

Step 4: The components are:

- Label
- TextField
- TextArea
- Checkbox
- Choice
- Button

Step 5: Add the GUI components in the frame

Step 6: Give applet code at the end of the program

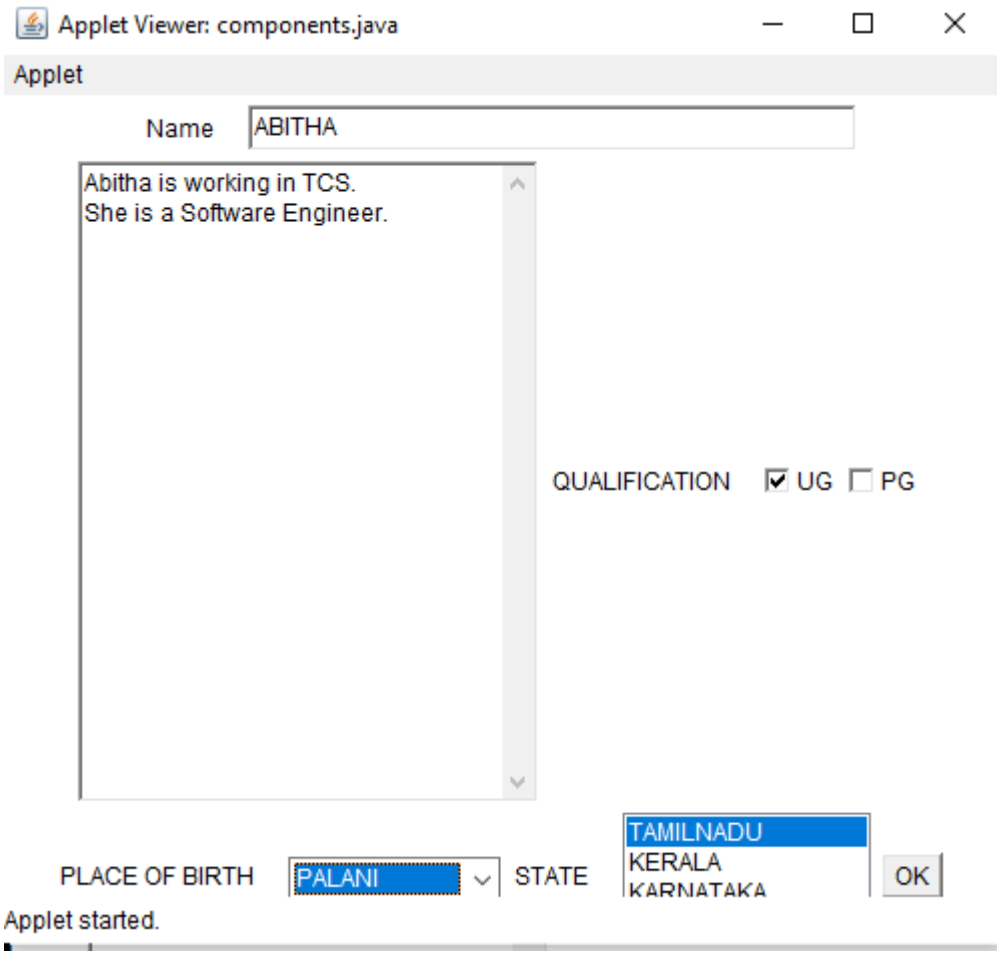
Step 7: Stop the program

Program:

```
import java.awt.*;
import java.applet.*;
public class components extends Applet
{
Label l1=new Label("Name");
TextField t1=new TextField(40);
TextArea ta=new TextArea(20,30);
Label l2=new Label("QUALIFICATION");
Checkbox c1=new Checkbox("UG",true);
Checkbox c2=new Checkbox("PG",false);
Label l3=new Label("PLACE OF BIRTH");
Label l4=new Label("STATE");
Choice ch1=new Choice();
List lt=new List();
Button b=new Button("OK");
public void init()
{
add(l1);
add(t1);
add(ta);
add(l2);
add(c1);
add(c2);
add(l3);

ch1.add("PALANI");
ch1.add("COIMBATORE");
add(ch1);
add(l4);
```

```
lt.add("TAMILNADU");
lt.add("KERALA");
lt.add("KARNATAKA");
add(lt);
add(b);
}
}

//<applet code="components.java" height=400 width=500></applet>
```

OUTPUT


Applet Viewer: components.java
Applet
Name
ABITHA
Abitha is working in TCS.
She is a Software Engineer.
QUALIFICATION
UG
PG
PLACE OF BIRTH
PALANI
STATE
TAMILNADU
KERALA
KARNATAKA
OK
Applet started.

EVENT HANDLING

Ex. No.: 21
Date:

Aim: To create a Java program to implement event handling task in an applet.

Algorithm:

Step 1: Start the program

Step 2: Create a class which extends Applet and implementing MouseListener

Step 3: Initialise an applet using init() method

Step 4: Invoke mouse Events to change colors accordingly

Step 5: The operations are:

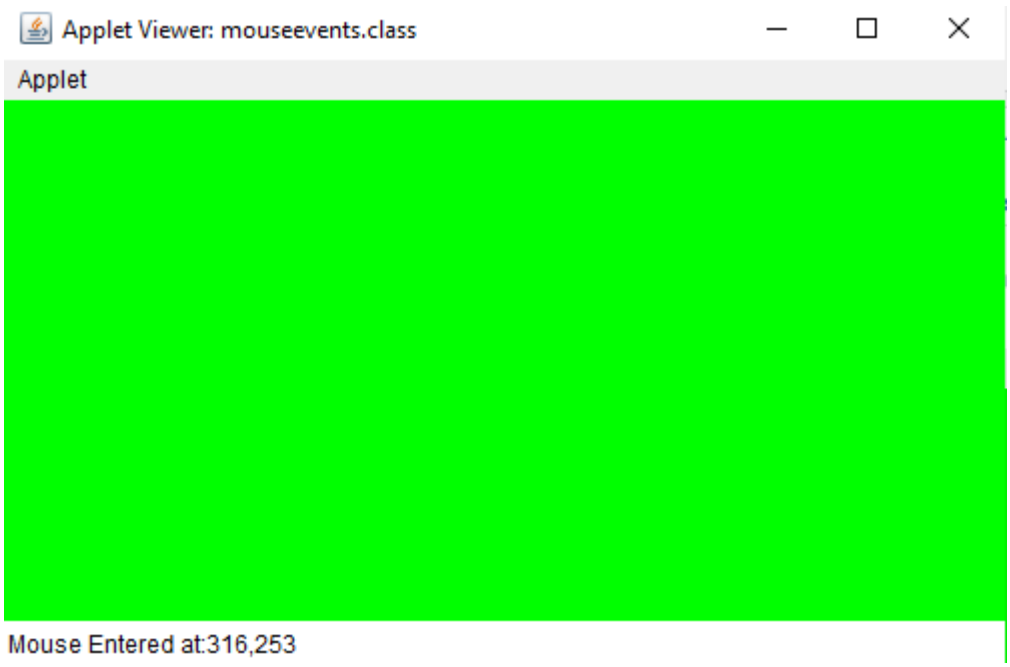
- Clicked
- Entered
- Exited
- Pressed
- Released

Step 6: Stop the program

Program:

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class mouseevents extends Applet implements MouseListener
{
public void init()
{
addMouseListener(this);
}

public void mouseClicked(MouseEvent e)
{
showStatus("Mouse Clicked at:"+e.getX()+","+e.getY());
setBackground(Color.cyan);
repaint();
}
public void mouseEntered(MouseEvent e)
{
setBackground(Color.green);
showStatus("Mouse Entered at:"+e.getX()+","+e.getY());
repaint();
}
public void mouseExited(MouseEvent e)
{
setBackground(Color.blue);
repaint();
showStatus("Mouse Exited at:"+e.getX()+","+e.getY());
}
public void mousePressed(MouseEvent e)
{
setBackground(Color.pink);
repaint();
showStatus("Mouse Pressed at:"+e.getX()+","+e.getY());
}
public void mouseReleased(MouseEvent e)
{
setBackground(Color.red);
repaint();
showStatus("Mouse Released at:"+e.getX()+","+e.getY());
}
}
/*<applet code="mouseevents.class" height="400" width="500"></applet>*/
```

OUTPUT


Applet Viewer: mouseevents.class
Applet
Mouse Entered at:316,253

JAVA IO STREAMS

Ex. No.: 22
Date:

Aim : To create a Java program to implement IO streams.

Algorithm:

Step 1: Start the program

Step 2: Create a class and main function

Step 3: Create DataInputStream class and object to get data from the keyboard (IO)

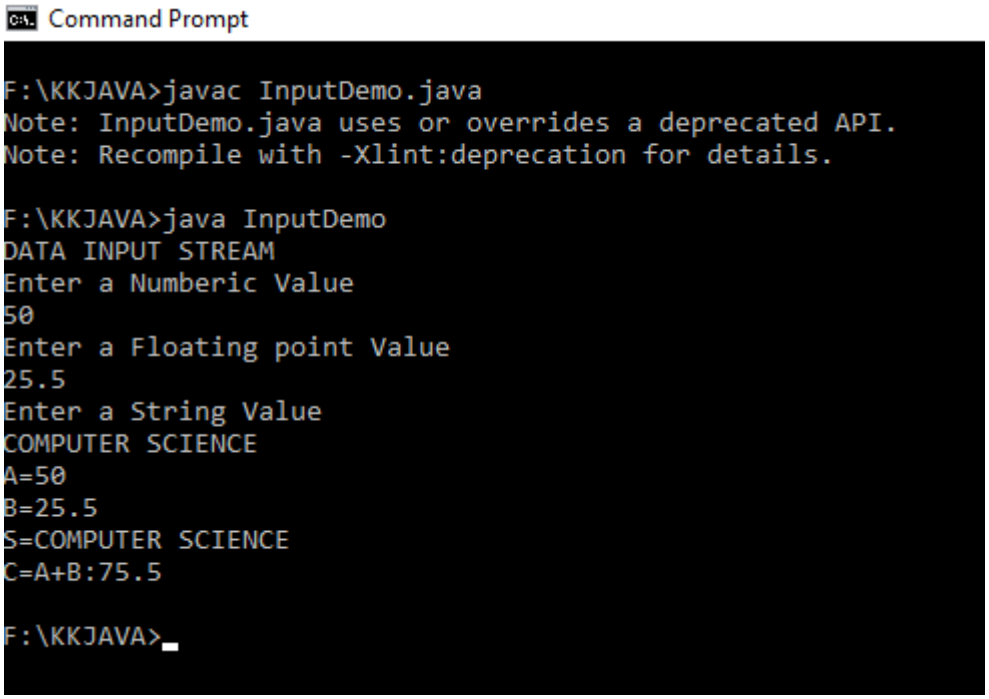
Step 4: Display the results.

Step 6: Stop the program

Program:

```
//Input through keyboard ://InputStream Reader
import java.io.*;
class InputDemo
{
public static void main(String args[])
{
int a;
float b,c;
String s;
try
{
DataInputStream dis=new DataInputStream(System.in);
System.out.println("Enter a Numberic Value");
a=Integer.parseInt(dis.readLine());
System.out.println("Enter a Floating point Value");
b=Float.parseFloat(dis.readLine());
System.out.println("Enter a String Value");
s=dis.readLine();
System.out.println("A="+a);
System.out.println("B="+b);
System.out.println("S="+s);
System.out.println("C=A+B:"+(a+b));
}
catch(Exception e)
{
System.out.println(e);
}
}
}
```

OUTPUT


Command Prompt
F:\KKJAVA>javac InputDemo.java
Note: InputDemo.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
F:\KKJAVA>java InputDemo
DATA INPUT STREAM
Enter a Numberic Value
50
Enter a Floating point Value
25.5
Enter a String Value
COMPUTER SCIENCE
A=50
B=25.5
S=COMPUTER SCIENCE
C=A+B:75.5
F:\KKJAVA>