SRM Institute of Science and Technology

(Deemed to be University U/S 3 of UGC Act, 1956)

Faculty of Science and Humanities

Tiruchirappalli, Tamil Nadu-621 105

Department of Computer Applications

PRACTICAL RECORD

NAME :

REGISTER NO :

COURSE : MCA (General)

SEMESTER/ YEAR : I / I

SUBJECT CODE : PCA25C01J

SUBJECT NAME : Object Oriented Programming Using Java

November 2025

SRM Institute of Science and Technology

(Deemed to be University U/S 3 of UGC Act, 1956)

Faculty of Science and Humanities

Tiruchirappalli, Tamil Nadu-621 105

Department of Computer Applications

BONAFIDE CERTIFICATE

This is to certify that the bonafide work is done by Mr/Ms. _____ Register No. _____in **Object Oriented Programming Using Java (Subject Code: PCA25C01J)** at Computer Lab, SRMIST, Tiruchirappalli, in November 2025.

STAFF IN-CHARGE **HEAD OF THE DEPARTMENT**

Submitted for the University Practical Examination held at SRMIST, Tiruchirappalli, Department of Computer Applications on _____.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

INDEX

S. No.	Date	Name of the Program	Page No.	Signature
1	16.07.2025	JAVA INPUT AND OUTPUT	04	
2	23.07.2025	TYPE CONVERSION	07	
3	30.07.2025	JAVA OPERATORS	10	
4	13.08.2025	LOOPING STATMENTS	13	
5	20.08.2025	WHILE STATMENT	16	
6	20.08.2025	DO-WHILE STATMENT	19	
7	27.08.2025	ARRAYS	22	
8	03.09.2025	SWITCH-CASE	26	
9	10.09.2025	COMMAND LINE ARGUEMENT AND STRING FUNCTIONS	29	
10	17.09.2025	INHERITANCE	32	
11	24.09.2025	EXCEPTIONS	35	
12	01.10.2025	MULTI-THREADING	38	
13	01.10.2025	PACKAGE	41	
14	08.10.2025	INTERFACE	44	
15	08.10.2025	LAYOUT MANAGER (FLOW LAYOUT)	47	

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

Date: 16.07.2025

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

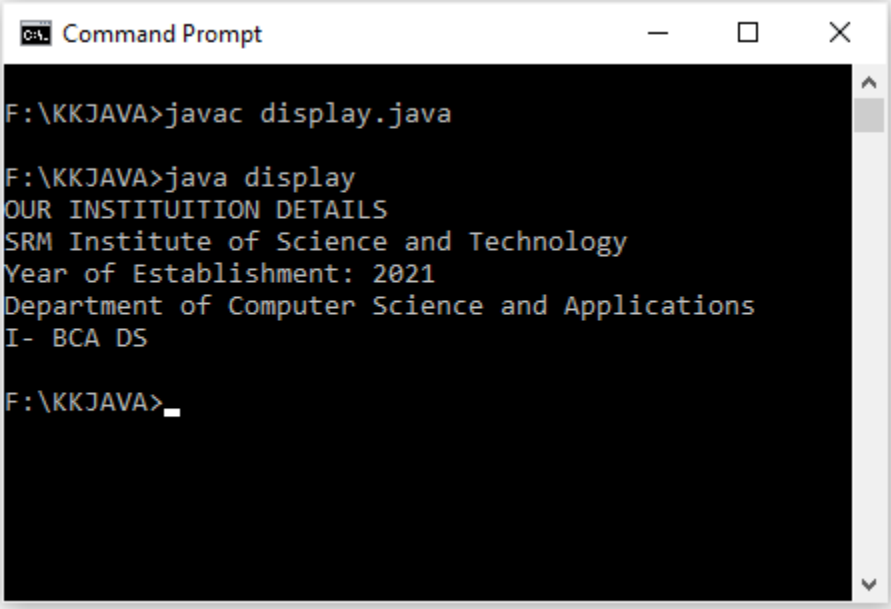
Step2: Define a class and display name of the institution, year of establishment, department and your class using Output method

Step3: Stop the program.

PROGRAM

```
import java.io.*; class
display
{
public static void main(String args[])
{
int year=2025;
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 Applications");
System.out.println("I- MCA-GEN AI");
}
}
```

OUTPUT



RESULT

The program runs successfully, displaying the institution's details as specified.

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

Date: 23.07.2025

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;

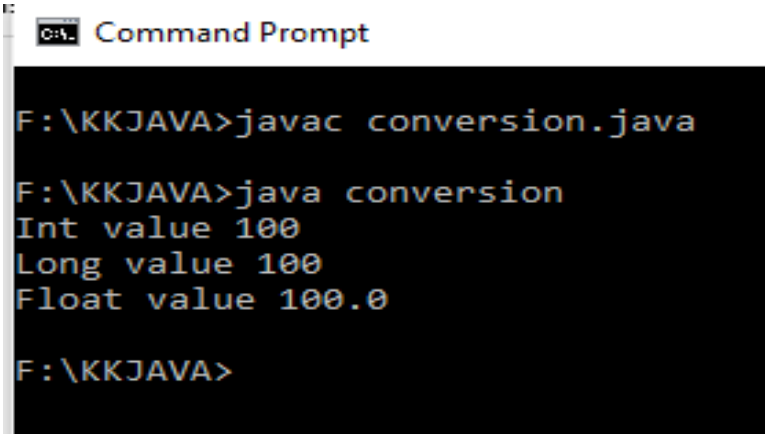
Step 5: Display the result.

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:



RESULT

The values are successfully converted and displayed.

Ex. No. : 3
Date: 30.07.2025

JAVA OPERATORS

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>_
```

RESULT

The program demonstrates various operators in Java, including arithmetic, unary, and shift operators and runs successfully.

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

Date: 13.08.2025

AIM

To create a Java program to implement looping statements

ALGORITHM

Step 1: Start the Program

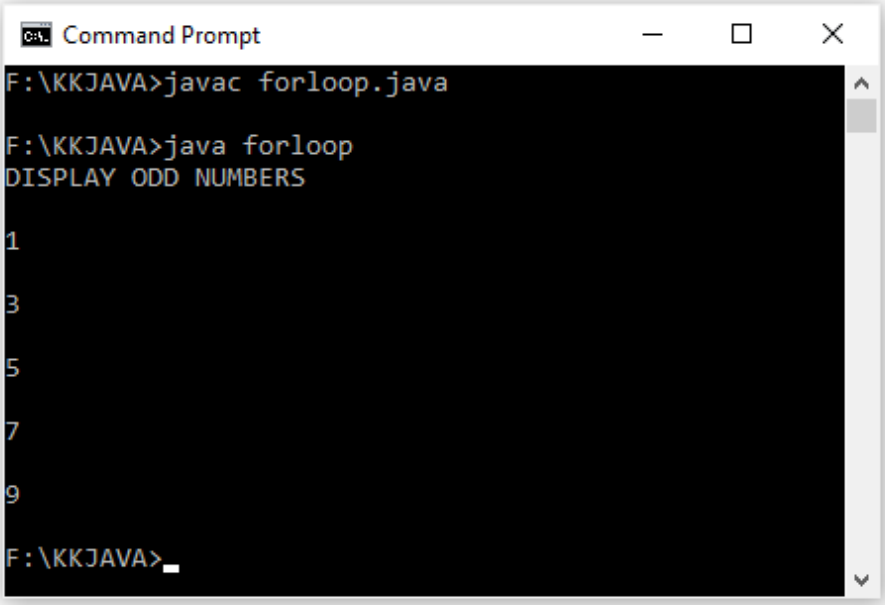
Step2: Define a class to display odd numbers using for-loop Step3:

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



RESULT

The program uses a for loop to display odd numbers from 1 to 10 and runs successfully.

Ex. No. :5 **WHILE STATMENT**

Date: 20.08.2025

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:

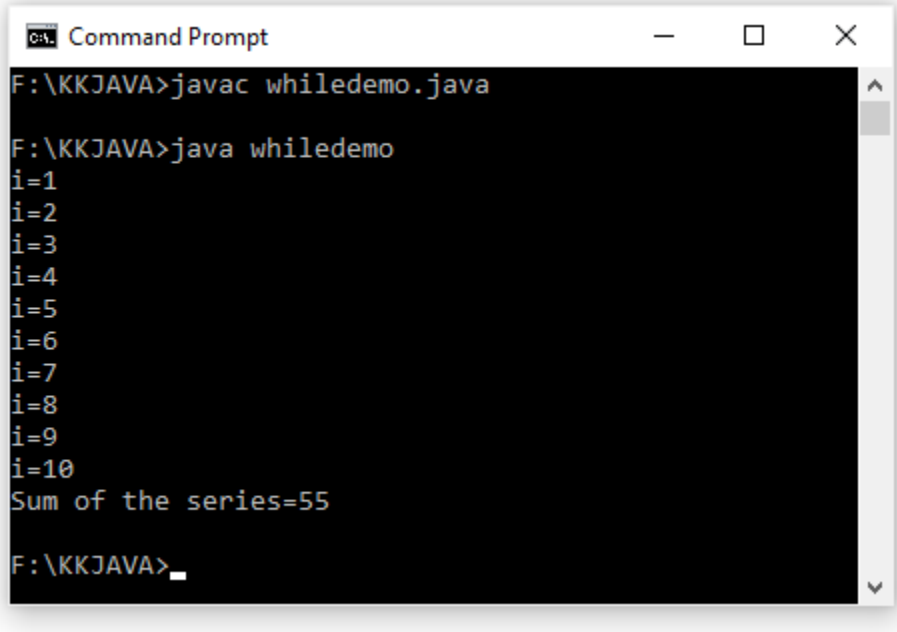
Stop the program

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



RESULT

The program uses a while loop to calculate the sum of the series 1+2+3+...+10 and runs successfully.

Ex. No. :6 **DO-WHILE STATMENT**

Date: 20.08.2025

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:

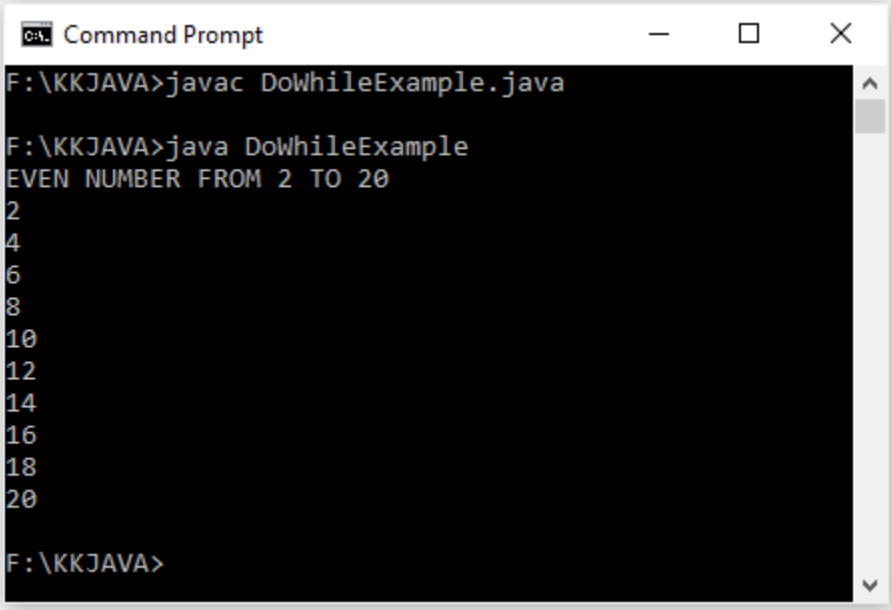
Stop the program

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



RESULT

The program uses a do-while loop to print the even numbers from 2 to 20 and program runs successfully.

Ex. No. 7 **ARRAYS**
Date: 27.08.2025

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

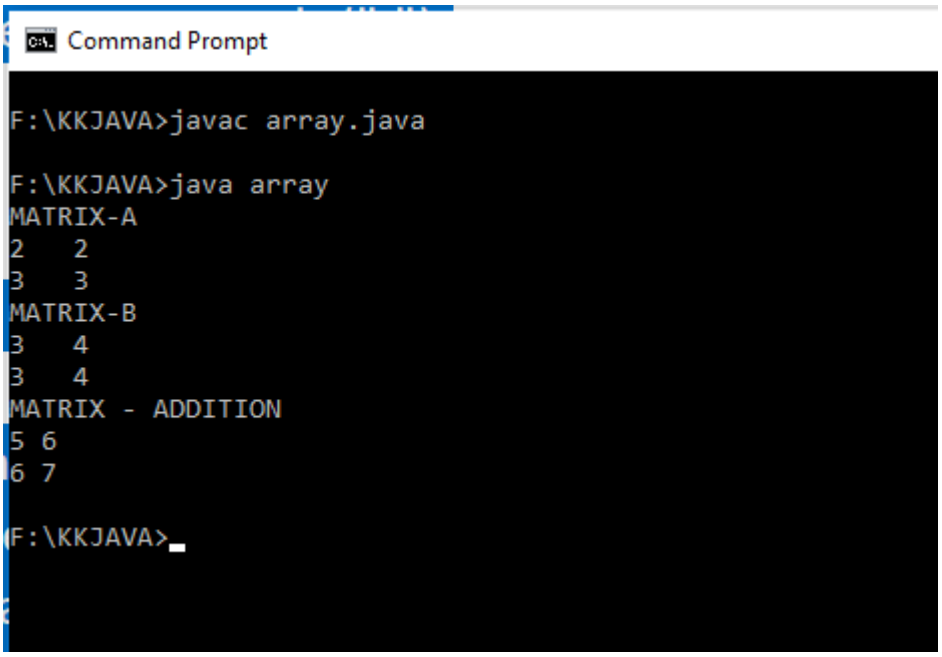
8: Stop the program

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



RESULT

The program performs matrix addition on two 2x2 matrices A and B, resulting matrix C (Matrix Addition) runs successfully.

SWITCH-CASE

Ex. No. :8
Date: 03.09.2025

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

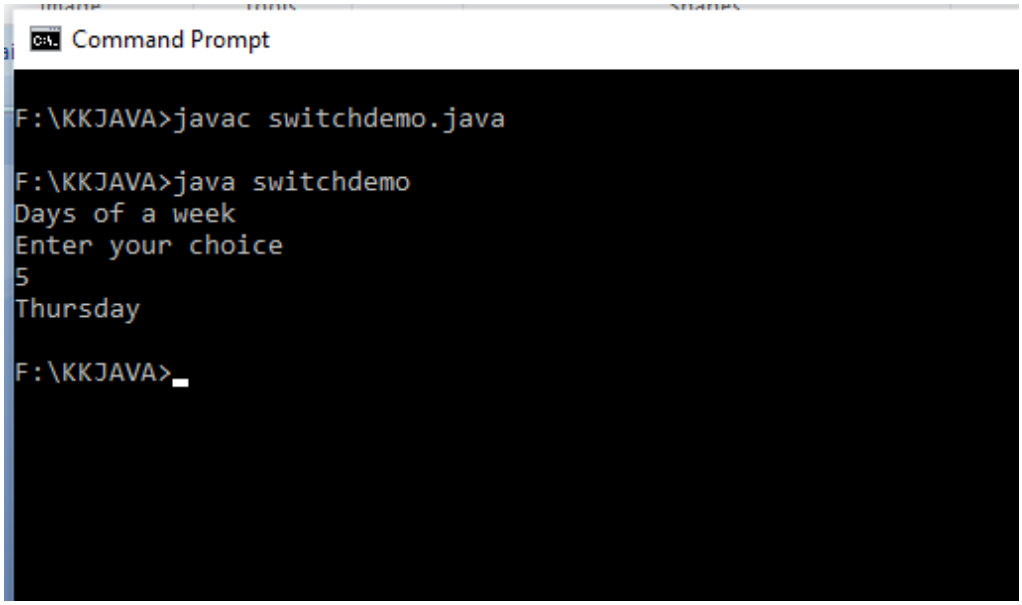
Step 5: Display the day as output Step

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;

        System.out.println("Tuesday"); break;
case 3:
        System.out.println("Wednesday");
        break;
case 4:
        System.out.println("Thursday");
        break;
case 5:
        System.out.println("Friday");  break;

case 6: System.out.println("Saturday");
        break;

case 7:


}
}
}
```

OUTPUT



RESULT

The program runs successfully.

COMMAND LINE ARGUEMENT AND STRING FUNCTIONS

Ex. No. :9
Date: 10.09.2025

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().

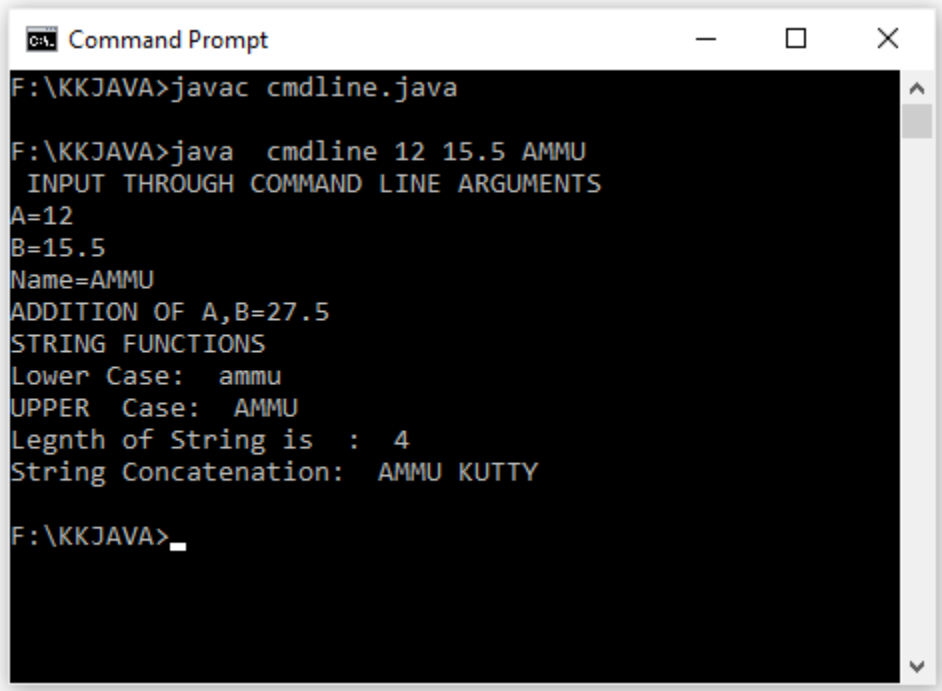
Step 7: stop the program.

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



RESULT

The program takes command line arguments, converts them to respective data types, performs calculations, and implements string functions. The program runs successfully.

INHERITANCE

Ex. No. :10
Date: 17.09.2025

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>
```

RESULT

The program demonstrates inheritance in Java, where the PoojaRoom subclass extends the Room superclass. The PoojaRoom class inherits the area() method from Room and also has its own volume() method. The program calculates and displays the area and volume of a room. The program runs successfully!

EXCEPTIONS

Ex. No. :11
Date: 24.09.2025

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>
```

RESULT:

Thus the program executed and the output obtained successfully.

MULTI- THREADING

Ex. No. :12
Date: 01.10.2025

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
```

RESULT:

Thus the program executed and the output obtained succesfully

PACKAGE

Ex. No. :13
Date: 01.10.2025

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>_
```

RESULT

The program successfully demonstrates package usage in Java by calculating rectangle area and performing addition. It prints area and addition results, showcasing package implementation.

INTERFACE

Ex. No. :14
Date: 08.10.2025

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.

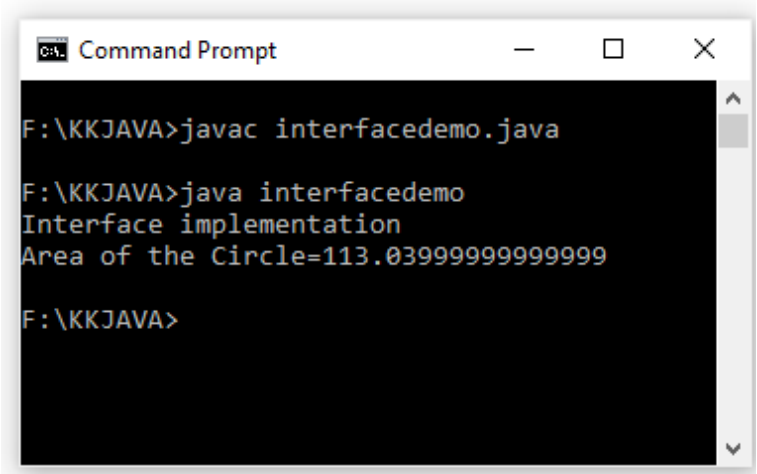
Step 6: Print the results. Step

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



RESULT

The program implements an interface I1 with a variable r, which is used by class C1 to calculate and print the area of a circle. The output is Area of the Circle=113.04.

LAYOUT MANAGER (FLOW LAYOUT)

Ex. No.: 15

Date: 08.10.2025

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()

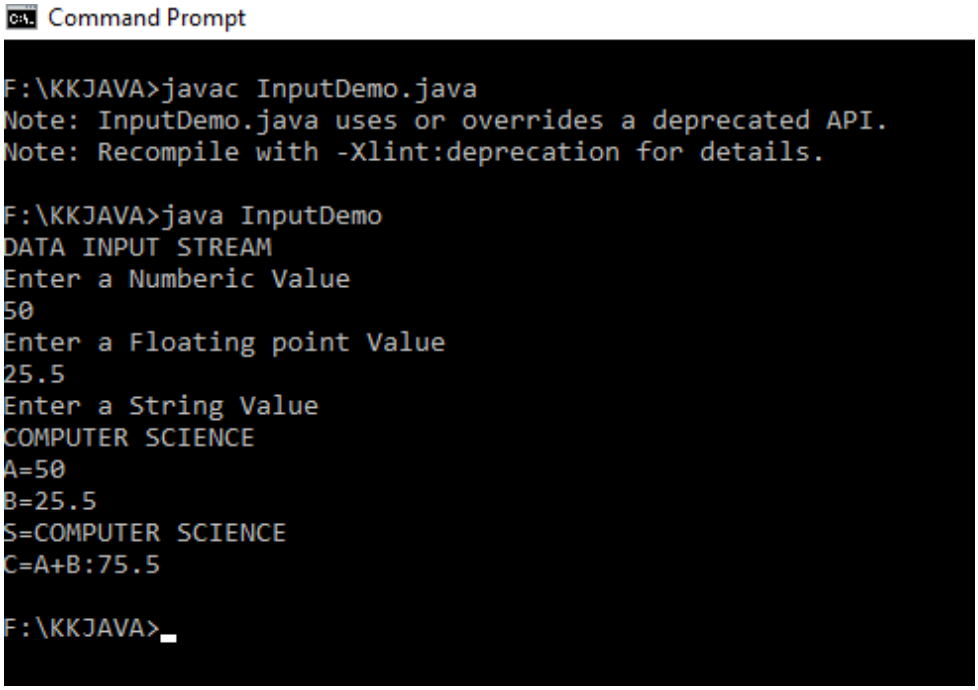
Step 6: Create a main class and create an object for Frame class and show layout. Step 7:

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



RESULT

The Java program creates a FlowLayout with buttons labeled "ONE" to "FIVE" using AWT. The buttons are arranged horizontally in a flow layout within an applet window.