Thakur Educational Trust's (Regd.)
THAKUR COLLEGE OF SCIENCE & COMMERCE

PRACTICAL JOURNAL

B.Sc. (Computer Science)
Sem – 3
[2021 –2022]

(TCSCCS0302P)
Programming in Java

Department of Computer Science
Thakur College of Science and Commerce
Thakur Village, Kandivali (East)
Mumbai – 400101

Thakur Educational Trust's (Regd.)

THAKUR COLLEGE OF SCIENCE & COMMERCE

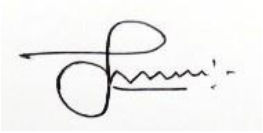
Autonomous College Permanently Affiliated to University of Mumbai (NAAC Accredited with Grade 'A' [3rd Cycle] & ISO 9001:2015 Certified)

Best College Award by University of Mumbai for Year 2018-2019

CERTIFICATE

This is here to certify that Mr./Mast. **Kaysan R. Shaikh**, Roll Number of **4334** B.Sc. Computer Science, has satisfactorily completed the required number of experiments prescribed by the college during the academic year 2021– 2022.

Mumbai



Teacher In-Charge Head of Department

External Examiner

Index

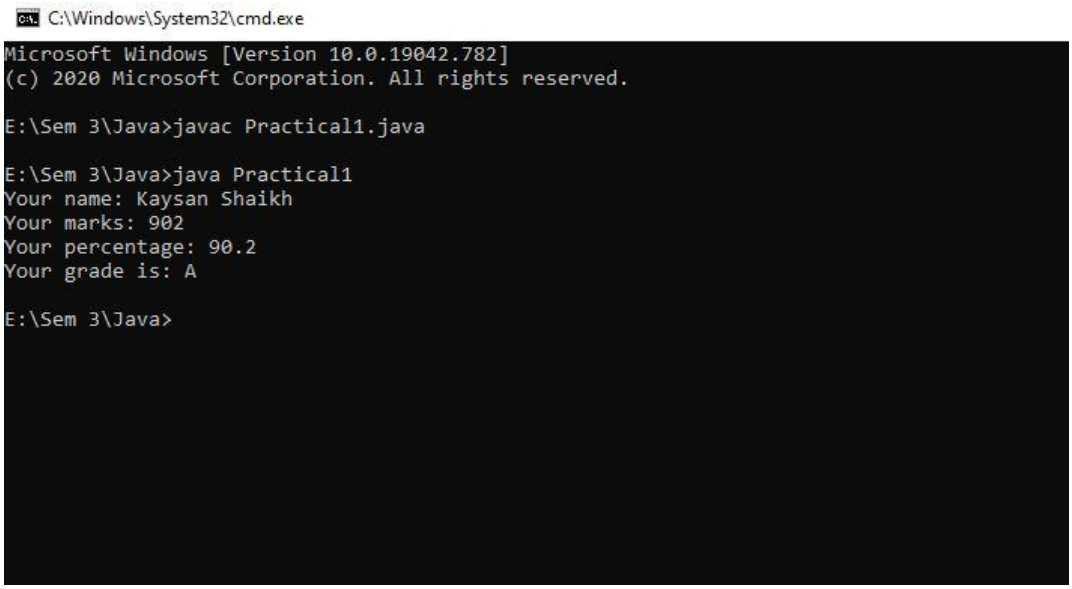
Practical 1

Aim: Programs to demonstrate the use of various Data types and Variables by creating CUI Application.

Code:

```
class Practical1 {

   public static void main(String[] args) {

        String name="Kaysan Shaikh";

        int m=902;

        float per=90.2f;

        char grade='A';


       System.out.println("Your name: " + name);

       System.out.println("Your marks: " + m);

       System.out.println("Your percentage: " + per);

       System.out.println("Your grade is: " + grade);

   } }}
```

Output:



Practical 2

Aim: Program demonstration on Strings:

a) String Comparisons

b) String Concatenation

c) Check the equality of two strings

d) Reverse a string

e) Finding length of string

a. String Comparisons

Code:

```
class Practical2a
{
public static void main(String args[])
{
String s1="Jack";
String s2="Jack";
System.out.println(s1.equals(s2));
String s3="Zack";
String s4="Zill";
System.out.println(s3==s4);
String s5="R";
String s6="FM";
System.out.println(s5.compareTo(s6));

}
}
```

Output:

```
------------------------< com.mycompany:Project >------------------------
Building Project 1.0-SNAPSHOT
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
true
false
12
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.955 s
Finished at: 2021-10-01T18:39:13+05:30
------------------------------------------------------------------------
```

b. String Concatenation

Code:

```
class Practical2
{
public static void main(String args[])
{
String s1="Far";
String s2="Cry";
String s3=s1.concat(s2);
System.out.println("The concatenated string is \t"
+s3);
String s4= "Assassin" + "Creed";
System.out.println(s4);
}
}
```

Output:

```
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
The concatenated string is      FarCry
AssassinCreed
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.989 s
Finished at: 2021-10-01T18:46:28+05:30
------------------------------------------------------------------------
'cmd' is not recognized as an internal or external command,
```

c. Check the equality of two strings

Code:

```
class Practical2
{
public static void main(String args[])
{
String s1="Red Velvet";
String s2="TWICE";
String s3="TWICE";
boolean a = s1.equals(s3);
System.out.println("s1 is equal to s3\t" +a);
boolean b = s2.equals(s3);
System.out.println("s2 is equal to s3\t" +b);
}
}
```

Output:

```
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
s1 is equal to s3       false
s2 is equal to s3       true
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.953 s
Finished at: 2021-10-01T18:52:02+05:30
```

d. Finding length of string

Code:

```
import java.util.Scanner;

class Practical2

{

public static void main(String[] arg)

{

String str;

Scanner scan=new Scanner(System.in);

System.out.print("Enter a string : \r\n");

str=scan.nextLine();

char[] ch=str.toCharArray();

System.out.println("Reverse of a String is :");

int j=ch.length;

for(int i=j;i>0;i--)

{

System.out.print(ch[i-1]);

}

}
```

}

Output:

```
ut - Run (Project)
Building Project 1.0-SNAPSHOT
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
Enter a string :
Valorant
Reverse of a String is :
tnarolaV
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  17.350 s
Finished at: 2021-10-01T18:57:29+05:30
```

Practical 3

Aim: Create Java Application by using Array:

a) Program to print the largest and smallest element in an array.

b) Program to print the sum of all the items of the array.

c) Program to add two Matrices.

d) Program to multiply two Matrices.

e) Program to sort the elements of an array in ascending order.

a. Program to print the largest and smallest element in an array.

Code:

```
public class Practical3 {
public static void main(String[] args) {
int i;
int arr[]= new int[]{30,50,70,60,80,100};
int smallest = arr[0];
int largest = arr[0];
for(i=1; i<arr.length; i++)
{
if(arr[i]>largest)
{
largest=arr[i];
}
else if(arr[i]<smallest)
{
smallest=arr[i];
}
```

```
}

System.out.println("Largest element in the array is:" +largest);

System.out.println("Smallest element in the array is:" +smallest);

}

}
```

Output:

```
-
 --------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
 Largest element in the array is:100
- Smallest element in the array is:30
 ------------------------------------------------------------------------
 BUILD SUCCESS
 ------------------------------------------------------------------------
 Total time:  1.077 s
 Finished at: 2021-10-01T19:07:27+05:30
 ------------------------------------------------------------------------
 'cmd' is not recognized as an internal or external command,
```

b. Program to print the sum of all the items of the array.

Code:

```
public class Array {

public static void main(String[] args) {

int i,sum=0;

int arr[]= new int[]{100,10,20,30};

for(i=0; i<arr.length; i++)

{

sum = sum + arr[i];

}

System.out.println("Sum of all elements in the array is:" +sum);

}
```

}

Output:

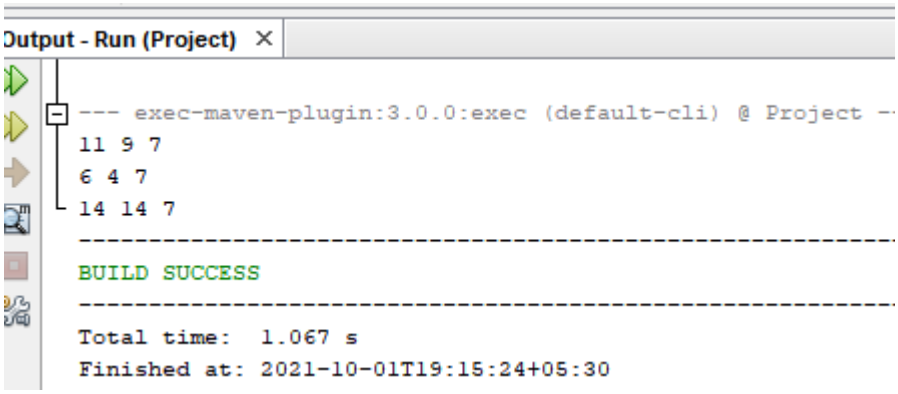
```
--------------------------------[ jar ]-------------------
] --- exec-maven-plugin:3.0.0:exec (default-cli) @ Project
- Sum of all elements in the array is:160
------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------
Total time:  1.057 s
Finished at: 2021-10-01T19:12:08+05:30
```

c. **Program to add two Matrices.**

Code:

```
public class Array{
public static void main(String args[]){
int a[][] = {{2,2,4},{4,1,2},{5,8,4}};
int b[][] = {{9,7,3},{2,3,5},{9,6,3}};
int c[][] = new int[3][3]; //3 rows and 3 columns
for(int i=0; i<3; i++)
{
for(int j=0; j<3; j++)
{
c[i][j] = a[i][j] + b[i][j];
System.out.print(c[i][j]+" ");
}
System.out.println();
}
}
}
```

Output:



d. Program to multiply two Matrices.

Code:

```
public class Array{
public static void main(String args[]){
int a[][]={{1,4,5},{3,2,7},{4,3,4}};
int b[][]={{1,8,3},{2,6,3},{7,3,2}};
int c[][]=new int[3][3];
for(int i=0;i<3;i++)
{
for(int j=0;j<3;j++)
{
c[i][j]=0;
for(int k=0;k<3;k++)
{
c[i][j]+=a[i][k]*b[k][j];
}
System.out.print(c[i][j]+" ");
}
```

```
System.out.println();

}

}

}
```

Output:

```
] --- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
  44 47 25
  56 57 29
- 38 62 29
  ------------------------------------------------------------------------
  BUILD SUCCESS
  ------------------------------------------------------------------------
  Total time:  1.070 s
  Finished at: 2021-10-01T19:20:07+05:30
  ------------------------------------------------------------------------
```

e. **Program to sort the elements of an array in ascending order.**

Code:

```
public class Array {

public static void main(String[] args) {

int [] arr = new int [] {3, 9, 7, 44, 11, 32};

int temp = 0;

System.out.println("Elements of original array: ");

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

{

System.out.print(arr[i] + " ");

}

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

{

for (int j = i+1; j < arr.length; j++)

{
```

```
if(arr[i] > arr[j])
{
temp = arr[i];
arr[i] = arr[j];
arr[j] = temp;
}
}
}
System.out.println();
System.out.println("Elements of array sorted in ascending order: ");
for (int i = 0; i < arr.length; i++)
{
System.out.print(arr[i] + " ");
}
}
}
```

Output:

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
Elements of original array:
3 9 7 44 11 32
Elements of array sorted in ascending order:
3 7 9 11 32 44
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.064 s
Finished at: 2021-10-01T19:32:39+05:30
------------------------------------------------------------------------
```

Practical 4

Aim: Write an appropriate Java program to demonstrate the use of different types of operators.

a) Arithmetic operators

b) Relational operators

c) Logical operators

d) Assignment operators

e) Increment and Decrement operators

f) Conditional operators

g) Bitwise operators

h) Special operators

a. Arithmetic operators

Code:

```
class Operators {
public static void main(String[] args) {
int a=5, b=10;

//Addition Operation
int sum = a + b;
System.out.println("Addition is: " + sum);

//Subtraction Operation
int diff = a - b;
System.out.println("Subtraction is : " + diff);
```

```
//Multiplication Operation

int multi = a * b;

System.out.println("Multiplication is : " + multi);


//Division Operation

int div = a / b;

System.out.println("Division is : " + div);


//Modulus Operation

int rem = a % b;

System.out.println("Modulus is : " + rem);

}

}
```

Output:

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
Addition is: 15
Subtraction is : -5
Multiplication is : 50
Division is : 0
Modulus is : 5
------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------
Total time:  1.069 s
Finished at: 2021-10-01T19:38:22+05:30
------------------------------------------------------------
```

b. Relational operators

Code:

```
class Operators {
public static void main(String[] args) {
int a=5, b=10;

//is equal to
System.out.println("a == b is " + (a == b) );

//is not equal to
System.out.println("a != b is " + (a != b) );

//Greater than
System.out.println("a > b is " + (a > b) );

//Less than
System.out.println("a < b is " + (a < b) );

//Greater than or equal to
System.out.println("a >= b is " + (a >= b) );

//Less than or equal to
System.out.println("a <= b is " + (a <= b) );
}
}
```

Output:

```
] --- exec-maven-plugin:3.0.0:exec (default-cli) @ Project
 a == b is false
 a != b is true
 a > b is false
 a < b is true
 a >= b is false
 a <= b is true
 ------------------------------------------------------------
 BUILD SUCCESS
 ------------------------------------------------------------
 Total time:  1.079 s
```

c. Logical operators

Code:

```
class Operators {
public static void main(String[] args) {
boolean a=true, b=false;

//Logical AND
System.out.println("a AND b is " + (a && b));

//Logical OR
System.out.println("a OR b is " + (a | b) );

//Logical Not
System.out.println("Reverse(a AND b) is " + !(a && b));
}
}
```

Output:

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
a AND b is false
a OR b is true
Reverse(a AND b) is true
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.061 s
Finished at: 2021-10-01T19:44:44+05:30
------------------------------------------------------------------------
```

d. Assignment operators

Code:

```
class Operators {

public static void main(String[] args) {

int a=5, b=10, c;


//Simple assignment operator

System.out.println("c = a+b is " + (c=a+b));


//Add AND assignment operator

System.out.println("c += a is " + (c+=a));


//Subtract AND assignment operator

System.out.println("c -= b is " + (c-=b));


//Multiply AND assignment operator

System.out.println("c *= a is " + (c*=a));
```

```
//Divide AND assignment operator
System.out.println("c /= a is " + (c/=a));

//Modulus AND assignment operator
System.out.println("c %= a is " + (c%=a));

int x=10;
//Left shift AND assignment operator
System.out.println("x <<= 1 is " + (x<<=1));

int y=5;
//Right shift AND assignment operator
System.out.println("y >>= 1 is " + (y>>=1));

int z=4;
//Bitwise AND assignment operator
System.out.println("z &= 2 is " + (z&=2));

int w=2;
//Bitwise exclusive OR assignment operator
System.out.println("w ^= 2 is " + (w^=2));

int k=9;
//Bitwise inclusive OR assignment operator

System.out.println("k |= 5 is " + (k|=5));
}
```

}

Output:

```
Building Project 1.0-SNAPSHOT
--------------------------------[ jar ]-------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
c = a+b is 15
c += a is 20
c -= b is 10
c *= a is 50
c /= a is 10
c %= a is 0
x <<= 1 is 20
y >>= 1 is 2
z &= 2 is 0
w ^= 2 is 0
k |= 5 is 13
------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------
```

e. Increment and Decrement operators

Code:

```
class Operators {

public static void main(String[] args) {

int a = 18;

System.out.println("a = " + a++);

System.out.println("a = " + a);

int b = 20;

System.out.println("b = " + b--);

System.out.println("b = " + b);

int x = 56;

System.out.println("x = " + ++x);

int y = 88;

System.out.println("y = " + --y);
```

```
}

}
```

Output:

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Proj
a = 18
a = 19
b = 20
b = 19
x = 57
y = 87
------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------
Total time:  1.060 s
Finished at: 2021-10-01T19:51:32+05:30
------------------------------------------------------
```

f. Conditional operators

Code:

```
class Operators {

public static void main(String[] args) {

String z;

int x=77, y=34;

z = x==y ? "Yes":"No";

System.out.println("Is X and Y value equal " +z);

}

}
```

Output:

```
] --- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
· Is X and Y value equal No
 ------------------------------------------------------------
 BUILD SUCCESS
 ------------------------------------------------------------
 Total time:  0.943 s
 Finished at: 2021-10-01T19:55:41+05:30
```

g. Bitwise operators

Code:

```
class Operators {

public static void main(String[] args) {

int x=4, y=7, z;


//Bitwise AND Operator

System.out.println("x & y = " + (x & y));


//Bitwise OR Operator

System.out.println("x | y = " + (x | y));


//Bitwise XOR Operator

System.out.println("x ^ y = " + (x ^ y));


//Binary Complement Operator

System.out.println("~y = " + ~y );


//Binary Left Shift Operator

z = x << 2;
```

```
System.out.println("x << 2 = " + z );


//Binary Right Shift Operator

z = x >> 2;

System.out.println("x >> 2 = " + z );


//Binary zero fill right shift operator

z = x >>> 2;

System.out.println("x >>> 2 = " + z );

}

}
```

Output:

```
 ------------------------< com.mycompany:Project >--------
] Building Project 1.0-SNAPSHOT
 --------------------------------[ jar ]-----------------

] --- exec-maven-plugin:3.0.0:exec (default-cli) @ Project
 x & y = 4
 x | y = 7
 x ^ y = 3
 ~y = -8
 x << 2 = 16
 x >> 2 = 1
- x >>> 2 = 1
 ------------------------------------------------------------
 BUILD SUCCESS
 ------------------------------------------------------------
 Total time:  1.057 s
```

Practical 5

Aim: Write a suitable Java Program using following selection statements:

a) if and if-else

b) nested-if

c) if-else-if

d) switch-case

a. if and if-else

1. if

Code:

```
package Practical5;

public class ConditonalStatements

{

public static void main(String[] args){

float marks=76f; //defining an 'marks' variable

if(marks>=75) //checking the marks

{

System.out.print("The Obtained Grade Is A");

}

}

}
```

Output:

```
--- exec-maven-plugin:3.0.0:exec (default
The Obtained Grade Is A
------------------------------------------
BUILD SUCCESS
------------------------------------------
Total time:  0.946 s
Finished at: 2021-10-02T09:57:27+05:30
```

2. **if else**

Code:

```
package Practical5;

public class ConditonalStatements

{

public static void main(String[] args) {

int a=10, b=10;

if (a>b)

{

System.out.print("a is greater than b");

}

else

{

System.out.print("a is equal to b");

}

}

}
```

Output:

```
------------------------< com.mycompany:Project >------------------------
Building Project 1.0-SNAPSHOT
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
a is equal to b
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.940 s
Finished at: 2021-10-02T10:03:54+05:30
------------------------------------------------------------------------
```

b. nested if

Code:

```
package Practical5;

public class ConditonalStatements

{

public static void main(String[] args){

int a=18;

if (a%2 == 0)

{

if (a%3 == 0)

{

System.out.println("a is divisible by both 2 & 3");

}

}

}

}
```

Output:

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
a is divisible by both 2 & 3
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.952 s
```

c. **if else if**

Code:

```
package Practical5;

import java.util.Scanner;

public class ConditonalStatements

{

public static void main(String[] args){

int a;

Scanner scan = new Scanner(System.in);

System.out.println("ENTER NUMBER :");

a = scan.nextInt();

if (a>0)

{

System.out.println("A is Posistive");

}

else if(a<0)

{

System.out.println("A is Negative");

}
```

```
else
{
System.out.println("A is Zero");
}
}
}
```

Output:

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Pr
ENTER NUMBER :
-39
A is Negative
------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------
Total time:  5.203 s
```

d. Switch-Case

Code:

```
package Practical5;
import java.util.Scanner;
public class ConditonalStatements
{
public static void main(String[] args) {
                //defining a variable
                int num;
Scanner scan = new Scanner(System.in);
System.out.println("Select any one:");
num = scan.nextInt();
                //switch expression
                switch(num) {
```

```
            //case statements
            case 10: System.out.println("The Selected Number Is 10 from this case");
            break;
            case 20: System.out.println("The Selected Number Is 20 from this case");
            break;
            case 30: System.out.println("The Selected Number Is 30 from this case");
            break;

            //default case
            default:
            System.out.println("The Number is not valid for the case");

            }
        }
}
```

Output:

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
Select any one:
10
The Selected Number Is 10 from this case
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
```

Practical 6

Aim: Implement Java programs using following loops:

a) for loop

b) for-each loop

c) while loop

d) do-while loop

a. for loop

Code:

```
package Practical6;

public class Loops {

public static void main(String[] args){

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

{

System.out.println(a);

}

}

}
```

Output:

```
1
2
3
4
5
6
7
8
9
10
------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------
```

b. for-each loop

Code:

```
package Practical6;

import java.util.*;

public class Loops {

public static void main(String[] args) {

ArrayList<String> list = new ArrayList<String>();

list.add("!Games");

list.add("Minecraft");

list.add("FarCry 6");

list.add("Valorant");


//traversing the list of elements using for-each loop

for(String s:list) {

System.out.println(s);

                }

        }
```

}

Output:

```
] --- exec-maven-plugin:3.0.0:exec (default-cli) @ Proje
 !Games
 Minecraft
 FarCry 6
 Valorant
 ------------------------------------------------------------
 BUILD SUCCESS
 ------------------------------------------------------------
 Total time:  0.952 s
 Finished at: 2021-10-02T10:51:10+05:30
 ------------------------------------------------------------
```

c. **while loop**

Code:

```
package Practical6;

public class Loops {

public static void main(String[] args){

int i=0;

while(i<=10)

{

System.out.println(i);

i++;

}

}

}
```

Output:

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
0
1
2
3
4
5
6
7
8
9
10
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
```

d. do-while loop

Code:

```
package Practical6;

public class Loops {

public static void main(String[] args){

int i=0;

do

{

System.out.println(i);

i++;

}

while(i<=5);

}

}
```

Output:

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
0
1
2
3
4
5
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
```

Practical 7

Aim: Write a suitable Java Program to demonstrate following types of Inheritance:

a) Single Inheritance

b) Multilevel Inheritance

c) Hierarchical Inheritance

d) Multiple Inheritance

e) Hybrid Inheritance

a. Single Inheritence

Code:

```
package Practical7;
class Games{
void Ubisoft(){System.out.println("Watch_Dogs...Farcry_6...");}
}
class WatchDogs extends Games{
void Play(){System.out.println("Playing Now...");}
}
class TestInheritance{
public static void main(String args[]){
WatchDogs w=new WatchDogs();
w.Play();
w.Ubisoft();
}
}
```

Output:

```
--------------------------------[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cl
Playing Now...
Watch_Dogs...Farcry_6...
------------------------------------------
BUILD SUCCESS
------------------------------------------
Total time:  0.995 s
```

b. Multilevel Inheritance

Code:

```
package Practical7;

class Games{

void Ubisoft(){System.out.println("Watch_Dogs...Farcry_6..");}

}

class WatchDogs extends Games{

void Play(){System.out.println("Playing...");}

}

class Streaming extends WatchDogs{

void Stream(){System.out.println("Streaming...");}

}

class TestInheritance{

public static void main(String args[]){

Streaming s=new Streaming();

s.Stream();

s.Play();

s.Ubisoft();

}

}
```

Output:

```
--------------------------------[ jar ]----------

--- exec-maven-plugin:3.0.0:exec (default-cli) @
Streaming...
Playing...
Watch_Dogs...Farcry_6..
------------------------------------------------
BUILD SUCCESS
------------------------------------------------
```

C. Hierarchical Inheritance

Code:

```
package Practical7;

class Games{

void Ubisoft(){System.out.println("WatchDogs...Farcry_6..");

}

}

class WatchDogs extends Games{

void MainCharcter(){System.out.println("Aiden Pearce");}

}

class FarCry extends Games{

void MainVillan(){System.out.println("Antón Castillo");}

}

class TestInheritance{

public static void main(String args[]){

FarCry f=new FarCry();

f.MainVillan();

f.Ubisoft();

WatchDogs w=new WatchDogs();

w.MainCharcter();
```

```
w.Ubisoft();

}}
```

Output:

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ P
Antón Castillo
WatchDogs...Farcry_6..
Aiden Pearce
WatchDogs...Farcry_6..
------------------------------------------------
BUILD SUCCESS
------------------------------------------------
Total time:  0.944 s
```

d. Multiple Inheritance

Multiple level Inheritance is not supported in java As one child cannot have two fathers

Code:

```
class A{

void msg(){System.out.println("Hello");}

}

class B{

void msg(){System.out.println("Welcome");}

}

class C extends A,B{//suppose if it were


 Public Static void main(String args[]){

   C obj=new C();

   obj.msg();//Now which msg() method would be invoked?

}

}
```

Output:

Compile by: javac C.java

```
.68.79.179/C.java:7: error: '{' expected
class C extends A,B{//suppose if it were
 ^
.68.79.179/C.java:9: error: ';' expected
 Public Static void main(String args[]){
 ^
2 errors
```

d. Hybrid Inheritance

Code:

```
package Practical7;

//parent class

class GrandFather

{

public void show()

{

System.out.println("I am grandfather.");

}

}

//inherits GrandFather properties

class Father extends GrandFather

{

public void show()

{

System.out.println("I am father.");

}

}
```

```
//inherits Father properties

class Daughter extends Father

{

public void show()

{

System.out.println("I am Daughter.");

}

}

//inherits Father properties

public class Son extends Father

{

public void show()

{

System.out.println("I am a Son.");

}

public static void main(String args[])

{

Son obj = new Son();

obj.show();

}

}
```

Output:

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @
I am a Son.
------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------
Total time:  0.994 s
```

Practical 8

Aim: Programs to demonstrate polymorphism:

a) Method over-riding

b) Method overloading

a. Method Over-riding

Code:

```
package Practical8;

//Java Program to demonstrate the real scenario of Java Method Overriding

//where three classes are overriding the method of a parent class.

//Creating a parent class.

class Bank{

int getRateOfInterest(){return 0;}

}

//Creating child classes.

class SBI extends Bank{

int getRateOfInterest(){return 8;}

}


class ICICI extends Bank{

int getRateOfInterest(){return 7;}

}

class AXIS extends Bank{

int getRateOfInterest(){return 9;}

}

//Test class to create objects and call the methods

class Test2{
```

```
public static void main(String args[]){

SBI s=new SBI();

ICICI i=new ICICI();

AXIS a=new AXIS();

System.out.println("SBI Rate of Interest: "+s.getRateOfInterest());

System.out.println("ICICI Rate of Interest: "+i.getRateOfInterest());

System.out.println("AXIS Rate of Interest: "+a.getRateOfInterest());

}

}
```

Output:

```
Building Project 1.0-SNAPSHOT
--------------------------------[ jar ]-------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project
SBI Rate of Interest: 8
ICICI Rate of Interest: 7
AXIS Rate of Interest: 9
------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------
Total time:  1.076 s
Finished at: 2021-10-02T11:54:36+05:30
```

b. Method Overloading

1. Overloading Changing the No. of arguments

Code:

```
package Practical8;

class Adder{

static int add(int a,int b){return a+b;}

static int add(int a,int b,int c){return a+b+c;}

}

class TestOverloading1{

public static void main(String[] args){

System.out.println(Adder.add(34,35));

System.out.println(Adder.add(11,31,21));

}

}
```

Output:

```
] Building Project 1.0-SNAPSHOT
  --------------------------------[ jar ]--------------------

] --- exec-maven-plugin:3.0.0:exec (default-cli) @ Project --
  69
  63
  ------------------------------------------------------------
  BUILD SUCCESS
  ------------------------------------------------------------
  Total time:  1.006 s
```

2. Overloading; changing data type of arguments

Code:

```
package Practical8;

class Adder{

static int add(int a, int b){return a+b;}

static double add(double a, double b){return a+b;}

}

class TestOverloading2{

public static void main(String[] args){

System.out.println(Adder.add(13,19));

System.out.println(Adder.add(72.3,42.6));

}

}
```

Output:

```
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
32
114.9
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.016 s
Finished at: 2021-10-02T12:05:40+05:30
```

Practical 9

Aim: Program to demonstrate all Math class functions.

Code:

```
package Practical9;

class MathClass

{

public static void main(String[] args)

{

double x = 100;

double y = 132;

float z = 78.20f;


// returns the maximum of two numbers

System.out.println("Maximum number of x and y is: "

+Math.max(x, y));


// returns the minimum of two numbers

System.out.println("Minimum number of x and y is: "

+Math.min(x, y));


// returns the square root of y


System.out.println("Square root of y is: " +

Math.sqrt(y));


// returns the cube root of y

System.out.println("Cube root of y is: " + Math.cbrt(y));
```

```
// returns the logarithm of given value
System.out.println("Logarithm of x is: " + Math.log(x));

// returns the trigonometric sine of x
System.out.println("Sine value of x is: " +Math.sin(x));

// returns the trigonometric cosine value of x
System.out.println("Cosine value of x is: "
+Math.cos(x));

// returns the trigonometric tangent value of x
System.out.println("Tangent value of x is: "
+Math.tan(x));

//returns the absolute value
System.out.println("Absolute value is:" +Math.abs(x));

//returns the floor value

System.out.println("Floor value is:" +Math.floor(x));

//returns the ceil value
System.out.println("Ceil value is:" +Math.ceil(x));

//returns the round value
System.out.println("Round value is:" +Math.round(z));
```

```
}

}
```

Output:

```
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Project ---
Maximum number of x and y is: 132.0
Minimum number of x and y is: 100.0
Square root of y is: 11.489125293076057
Cube root of y is: 5.091643369659489
Logarithm of x is: 4.605170185988092
Sine value of x is: -0.5063656411097588
Cosine value of x is: 0.8623188722876839
Tangent value of x is: -0.5872139151569291
Absolute value is:100.0
Floor value is:100.0
Ceil value is:100.0
Round value is:78
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
```