TOPIC - INHERITANCE

Inheritance is one the most important features of OOP. Inheritance is the classification of class where main class divided in different sub class. Main class is known as parent class or base class and their sub class is known as child class or derives class. Derive class is inherit of base class.

Inheritance allows a class to use the properties and methods of another class while adding its own functionality through additional properties and methods. Inheritance thus helps creating classes from existing classes through an IS-A relationship. When members are inherited, they can be used in the derived class that inherits them provided the base class permits the access data and method to do so.

There are three types of member used in inheritance-

i. Private – That member which is used only in defining class is known as private member.

ii. Protected- That member which is used in defining class and also their sub class is known as protected member.

iii. Public – That member which is used globally means in defining class, their sub class and also under main() is known as public member

Types of Inheritance

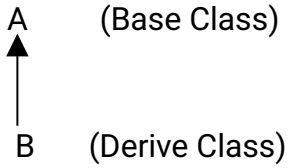
In Java there are three types of Inheritance-

i. Single Inheritance

ii. Multilevel Inheritance

iii. Multiple Inheritance

Single Inheritance- In this inheritance a base class has only one derive class.



EX-

Q. WAP in java to solve problem through inheritance concept where available base class member and derive class member then display all information by using derive class object.

Base Class Derive Class

Roll Fdate

Sname Fee

Course

Sol.

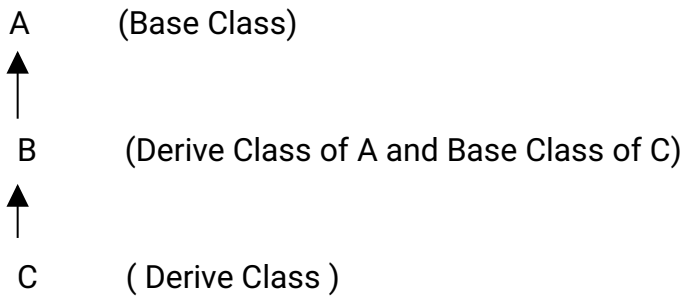
```
import java.util.*;
class  ak
{
private int roll;
private String sname,course;
void get()
{
Scanner sc=new Scanner(System.in);
System.out.println("Give Roll No.");
roll=sc.nextInt();
System.out.println("Give Student Name");
sname=sc.next();
System.out.println("Give Course");
course=sc.next();
}
void display()
{
System.out.println("Roll :"+roll);
System.out.println("Sname :"+sname);
System.out.println("Course :"+course);
}
}
class mk extends ak
```

```
{
private String fdate;
private int fee;
void get1()
{
Scanner sc=new Scanner(System.in);
super.get();
System.out.println(“Give Fdate”);
fdate=sc.next();
System.out.println(“Give fee”);
fee=sc.nextInt();
}
Void display1()
{
super.display();
System.out.println(“Fdate :”+fdate);
System.out.println(“Fee :”+fee);
}
}
class sk
{
public static void main(String ar[])
{
mk m=new mk();
m.get1();
m.display1();
}
```

```
}
```

[Note – In java by default member is public, if it is not defined. Super key word accesses to base class members. It is an important to technique that how to connect derive class with base class and all data access through derive class object.]

Multilevel Inheritance – In this inheritance one base class has only one derive class and that derive class becomes base class for other derive class and so on is known as multilevel inheritance. Derive class can accesses their base class members.



EX.

Q. Q. WAP in java to solve problem through inheritance concept where available base class member and derive class member then display all information by using derive class object.

Base Class 1	Base Class 2	Derive Class
Roll	Fdate	Semester
Sname	Fee	Examdate
Course		Result

Sol.

```
import java.util.*;
class  ak
{
private int roll;
private String sname,course;
void get()
{
Scanner sc=new Scanner(System.in);
```

```
System.out.println("Give Roll No.");
roll=sc.nextInt();
System.out.println("Give Student Name");
sname=sc.next();
System.out.println("Give Course");
course=sc.next();
}
void display()
{
System.out.println("Roll :"+roll);
System.out.println("Sname :"+sname);
System.out.println("Course :"+course);
}
}
class mk extends ak
{
private String fdate;
private int fee;
void get1()
{
Scanner sc=new Scanner(System.in);
super.get();
System.out.println("Give Fdate");
fdate=sc.next();
System.out.println("Give fee");
fee=sc.nextInt();
}
```

```
void display1()
{
super.display();
System.out.println("Fdate :"+fdate);
System.out.println("Fee :"+fee);
}
}
class rk extends mk
{
private String semester,examdate,result;
void get2()
{
Scanner sc=new Scanner(System.in);
super.get1();
System.out.println("Give Semester Name");
semester=sc.next();
System.out.println("Give Exam date");
examdate=sc.next();
System.out.println("Give Result");
result=sc.next();
}
Void display2()
{
super.display1();
System.out.println("Semester :"+semester);
System.out.println("Exam Date :"+examdate);
System.out.println("Result :"+result);
```

```
}
}


class sk
{
public static void main(String ar[])
{
rk r=new rk();
r.get2();
r.display2();
}
}
```

TOPIC – LIBRARY CLASSES OR LIBRARY METHODS OR FUNCTIONS:-

In java pre-defined methods are available in pre-define classes is called library classes and methods are known as library methods or function.

There are different types of library method or pre-define function:-

i. String function
ii. Number function
iii. Date function

etc.

String method / Function

Java string class provides a lot of methods to perform operation on string for string manipulation.

All these string functions are applicable on string variable or literals and return or result types of these functions are either int , char , boolean or String.

String s = “sample”;

Here :-

- s is string variable or object.
- “sample” is string literal .

1. Length() –

   This function counts the number of character (including blank spaces, digits, special characters) present in a string.

   Syntax:-

   < String Variable / Literal>.Length()

   Ex:-

   i) System.out.println(“Welcome”.Length());

   Output : 7

   ii) String s=”5*Star”);

   int l=s.Length();

   System.out.println(l);

   Output : 6

2. charAt() –

   This function extracts a character of the given position from a string.

   Syntax:-

   < String Variable / Literal>.charAt(position)

   [Note – If the given position is invalid then java will give the following runtime error message- “ String IndexOutofBoundsException”]

Ex –

Q. WAP in java to give any string then displays each character in next line.

Sol.

```
import java.util.*;
        class sk
                {
        public static void main(String ar[])
                {
        int i,l;
        String s;
        char ch;
        Scanner sc=new Scanner(System.in);
        System.out.println("Give any String");
        s=sc.next();
        l=s.Length();
        (for i=0;i<l;i++)
        {
        ch=s.charAt(i);
        System.out.println(ch);
        }
        }
        }
```

Input : - RAM

OUTPUT :- R

A

M

3. concat() –

   This function can join only a string with another string.

   Syntax:-

   < String Variable /Literal>.concat(String Variable/Literal)

   Ex-

   String a=”Patna”;

   String b=”Science”;

   String c=”College”;

   String d=a.concat(“ “).concat(b).concat(“ “).concat(c);

   Output :- Patna Science College

4. toUpperCase():-

   This function converts all lower case alphabets into upper case but all other characters remain unchanged.

   Syntax:-

   <String Variable/Literal>.toUpperCase()

   Ex –

   String s=”patna-800001”;

   String a=s.toUpperCase();

   System.out.println(a);

   Output :- PATNA-800001

5. toLowerCase():-

   This function converts all upper case alphabets into lower case but all other characters remain unchanged.

   Syntax:-

<String Variable/Literal>.toLowerCase()

Ex –

String s=”PATNA-800001”;

String a=s.toLowerCase();

System.out.println(a);

Output :- patna-800001

6. startsWith():-

   This function checks whether a string is present at the starting of another string or not.

   Syntax:-

   <String Variable/ Literal>.startsWith(String Variable/literal)

   Ex-

   System.out.println(“First java program”.startsWith(“First”));

   System.out.println(“First java program”.startsWith(“java”));

   Output :-

   True

   False

7. endsWith():-

   This function checks whether a string is present at the starting of another string or not.

   Syntax:-

   <String Variable/ Literal>.startsWith(String Variable/literal)

   Ex-

```
String s="Computer";
System.out.println(s.endsWith("puter"));
System.out.println(s.endsWith("put"));
System.out.println(s.endsWith("Computer"));
System.out.println(s.endsWith("computer));
```

Output :

True

False

True

False

8. replace():-

This function used for replace source string character with any other replacement character.

Syntax :-

<String Variable / Literal>.replace (Original character, replacement character)

Ex-

```
String s="GOOD".replace('O','E');
System.out.println(s);
```

Output :-

GEED

9. trim():-

This function ignores whitespaces from both sides of string and removed.

Syntax:-

<String Variable / Literal>.trim()

Ex-

```
String s="   WELCOME   ".trim();
String s1="   WELCOME  ";
int l=s.Length();
int l1=s1.Length();
System.out.println("Length of s:"+l);
System.out.println("Length of s1:"+l1);
```

Output:-

Length of s: 7
Length of s1:9

10. compareTo():-

This function compares two strings. If both strings are equal then return o. If first string is less than second string then return -1 otherwise 1.

Syntax :-

<int variable>=<String1>.compareTo(String2)

Ex-

```
String a="Amit";
String b="Binit";
Int c=a.compareTo(b);
System.out.println(c);
```

Output :-

-1