Senthamizhselvan.K.
192321143.
CSA0-0555

Inheritance:

Inheritance is a fundamental concept in Object Oriented Program [OOP] that allows a new class to inherit fields and Methods from an existing class.

Single inheritance

Single inheritance is a type of inheritance in which a subclass inherits from only one Superclass.

Examp:

Class A
↓
Class B

Program:

```
Class A {
    Int a;
    Void display A() {
        System.out.Println("a" +a);
    }
}
```

```
class B extends A
{
int b;
void display B()
{
System.out.Println("b="+b);
}
}
Public class Main{
Public static void main (String[] args)
{
B.obj = new B();
obj.a = 10;
obj.b = 20;
obj.display A();
obj.display B();
}
}
```

10

20

Multiple Inheritance In Java.

Multiple inheritance refers to the feature in some object oriented programming language where a class can inherit characteristics from more than one Parent class.

Example:

class A

class B

class C

Program:

```
class A {
    Public void method A ()
    {
    system. out. Println (" Inside the class");
    }
}
Class B extends A {
    Public void method B()
    {
    system. out. Println (" Inside class B");
    }
}
```

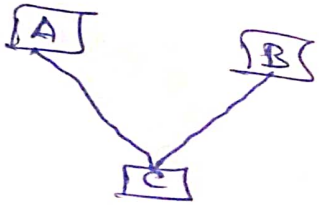
```
class C extends B{
    Public void Method ()
    {
      System.out.Println ("Inside class C");
    }
}
Public class Main
  {
    Public static void main (String [] args)
    {
      C obj = new C();
        obj c = method A();
        obj c = method B();
        obj c . method C();
    }
}
```

Multiple Inheritance

Multilevel

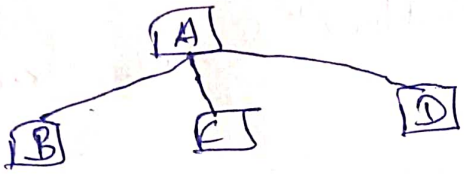
Multilevel inheritance is a type of inheritance in java where a class is derived from a class that is also derived from another class.

example



```
B. obj B = new B();
   obj B. display B();
   obj c . display c();
  }
}
```

Hierarchical Inheritance



It occurs when multiple subclass inherit from a single subclass. this means that a single Parent class can have multiple classes.

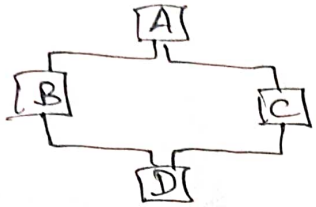
```
class A {
   A()
  { system . out. Println ("Inside class A");
  }
}
class B extends A {
```

```
Class A{
Void display A ()
 {
  system.out.Println ("Inside class A");
 }
}

Class B{
   Void display B ()
    {
     system.out.Println (" Inside class B");
    }
}
class c extends A (){
      Void display A ()
       {
        super.display A ();
        system.out.Println (" Inside class c");
       }
      void display ()
       {
        system.out.Println (" Method of class c");
       }
}
Public class Main
 {
 Public static void main (String [] args).
    {
      Cobj c = new C ();
       obj c.display A ();
```

Hybrid

Combine any Inheritance.




```
class A {
    Public void display A()
    {
    system.out.Println("Inside class A");
    }

class B {
    Public void display B()
    {
    system.out.Println("Inside class B");
    }
class D extends class c{
    Public. display
    {
     System.out.Println("Inside class D");
    }
Public class
    {
    Public static void main (String[] args)
    {
     C obj= new c();
     c obj D = new (C);
      obj c.display ();
```

```
Obj c.display();
obj c.displayc();
System.out.println();
    obj D.displayA();
    obj D.displayC();
    obj D.displayD();
}
}
```

Exception handling

An exception is an error during the execution of a Program.

Key Component of exception Handling:

try.
Catch,
throw,
Finally,
throws.

Try:

Try is a block of code to test the error being executed.

Catch:

Catch statement that block of code to be executed if an error occurs in try block.

Finally:

Code that always executed.

The finally block is a session of the code that is executed regardless of whether an exception is thrown or not.