2. ⓐ What is Inheritance? Explain various types of Inheritance Supported in Java.

Ans Inheritance is an important feature of object oriented programming. It is a process of deriving a new class from the existing class without modifying it. The new class is called the derived class or subclass or child class and the existing class is called the base class or super class or parent class. The derived class inherits all the features of the base class and you can also add some new features into it.

⇒ Syntax of Java Inheritance

```
class Subclass-name extends Superclass-name
{
    // methods and fields
}
```

The extends keyword indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality.

⇒ Types of Inheritance.

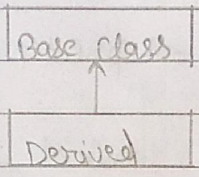
Java supports three types of inheritance

1) Single Inheritance
2) Multilevel Inheritance
3) Hierarchical Inheritance

In Java, multiple Inheritance can't be implemented directly with the help of classes only. Because a class in java can inherits at the most one class. but a class can implement number of interfaces.

1) Single Inheritance

when a subclass is derived simply from its parent class then this mechanism is known as Single Inheritance. In case of Single Inheritance there is only sub class and its parent class. It is also called Simple Inheritance or one level Inheritance. The following figure represents the structure of Single Inheritance.



Single Inheritance

⇒ Syntax

```
Class Base
{
        fields declaration;
        methods definition;
}
class Derived extends Base        // inheriting a class.
{
        fields declaration;
        methods definition;
}
```

SHIVA KUMAR 20210004

```
Class MainClass
{
    public static void main (String args[])
    {
        Create an object of derived class and
        invoke all methods of base class and
        derived class with this object.
    }
}
```

⇒ Program

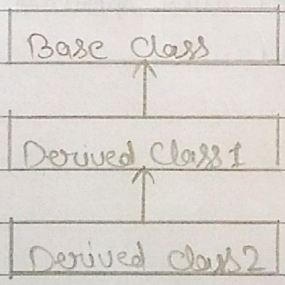
```
Class Animal
{
    void eat()
    { System.out.println ("eating..."); }
}
Class Dog extends Animal
{
    void bark()
    { System.out.println ("barking..."); }
}
Class SingleInheritance
{
    public static void main (String args[])
    {
        Dog d = new Dog();
        d.bark();
        d.eat();
    }
}
```

⇒ Output

barking...
eating...

2) Multilevel Inheritance

It is the enhancement of the concept of inheritance. The derived class is called the subclass or child class for its parents class and this parent class works as the child class for it's just above class. Multilevel Inheritance can go upto any number of levels but one class can extends only one parent class. The lowermost subclass can make use of all its super class member. when a subclass is derived from a derived class then this mechanism is known as multilevel inheritance.



Multilevel Inheritance

⇒ Syntax

```
Class Base
{
    fields declaration;
    methods definitions;
}
```

```
class Derived1 extends Base    // inheriting a Base class
{
    fields declaration;
    methods definitions;
}
class Derived2 extends Derived1    // inheriting a derived class
{
    fields declaration;
    methods definitions;
}
class multilevel Inheritance
{
    public static void main(String args[])
    {
        Creating an object of Derived2 class
        and invoke all methods of Base class
        Derived1 and Derived2 class with this
        object.
    }
}
```

⇒ Program

```
class Animal
{
    void eat()
    { System.out.println("eating..."); }
}
class Dog extends Animal
{
    void bark()
    { System.out.println("barking..."); }
}
```

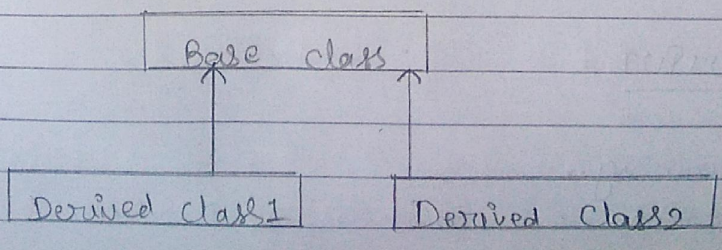
```
class BabyDog extends Dog
{
    void weep()
    { System.out.println(" weeping.. "); }
}
class multilevelInheritance
{
    public static void main(String args[])
    {
        BabyDog d = new BabyDog();
        d.weep();
        d.bark();
        d.eat();
    }
}
```

⇒ OUTPUT

```
weeping...
barking...
eating...
```

⇒ Hierarchical Inheritance

When two or more derived classes from a common base class, is known as hierarchical Inheritance.



Evergreen
Page No. 11
Date : / /201

SHIVA KUMAR 20210004

⇒ Program

```
Class Animal
{
    void eat()
    { System.out.println("eating...."); }
}
Class Dog extends Animal
{
    void bark()
    { System.out.println("barking...."); }
}
Class Cat extends Animal
{
    void meow()
    { System.out.println("meowing...."); }
}
Class HierarchicalInheritance
{
    public static void main(String args[])
    {
        Cat c = new Cat();
        c.meow();
        c.eat();
    }
}
```

⇒ OUTPUT

```
meowing...
eating...
```