Inheritance (IS-A Relation) :

```
public class Developer1
{
  public void sum(int x, int y)
  {
  }
  public void sub(int x, int y)
  {
  }
}
```

```
class Developer2
{
  public void sum(int x, int y)
  {
  }
  public void sub(int x, int y)
  {
  }
  public void mul(int x, int y)
  {
  }
  public void div(int x, int y)
  {
  }
}
```

In the above code written by Developer2, We are having some drawbacks :

1) Code Duplication
2) In order to write duplicate
   [Time consuming]
3) It is time consuming from Processor and Compiler point of view also.

* OOP SAYS WE SHOULD ALWAYS **REUSE OUR EXISTING CODE** RATHER THAN RE-CREATE

How to reuse the existing code :
---
* In order to reuse the existing code, we should use Inheritance concept.
* In order to provide Inheritance concept i.e IS-A relation we should use **extends** keyword.

```
public class Developer1
{
  public void sum(int x, int y)
  {
  }
  public void sub(int x, int y)
  {
  }
}
public class Developer2 extends Developer1
{
  public void mul(int x, int y)
  {
  }
  public void div(int x, int y)
  {
  }
}
```

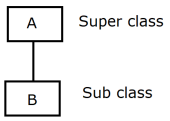
* Here Developer2 is reusing the code, which is written by Developer1.
* Developer2 can directly start from mul() method that menas no need to start from begning onwards.
* In this context :
  Developer1 is Parent class OR Super class
  Developer2 is Child class OR Sub class

Types of Inheritance in java :

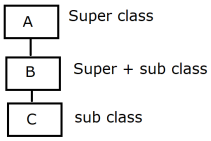
We have 5 types of Inheritance in java :

1) Single Level Inheritance
2) Multilevel Inheritance
3) Hierarchical Inheritance
4) Multiple Inheritance [Not supported by using class]
5) Hybrid Inheritance.

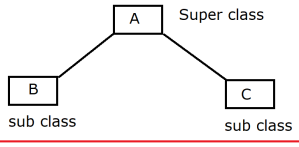
1) Single Level Inheritance



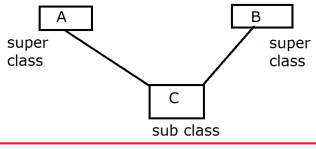
2) Multilevel Inheritance



3) Hierarchical Inheritance



4) Multiple Inheritance (Not Supported)



5) Hybrid Inheritance.