

# Class Inheritance and Type Casting

Date: April 8, 2008

Instructor : Dr. Rajat Moona

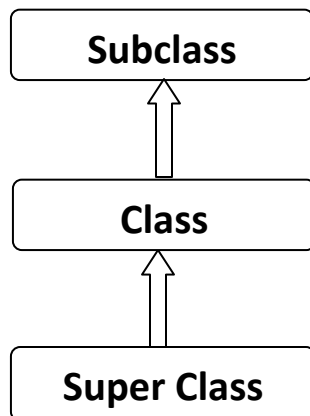
In previous class we learnt that, in object-oriented programming, **Inheritance** is a way to form new classes using classes that have already been defined. The new classes, known as *derived classes* inherit attributes and behavior of the pre-existing classes, which are referred to as *base classes*.

**Class Inheritance:** An object can belong to multiple types. Object of derived class is also an object of base class.

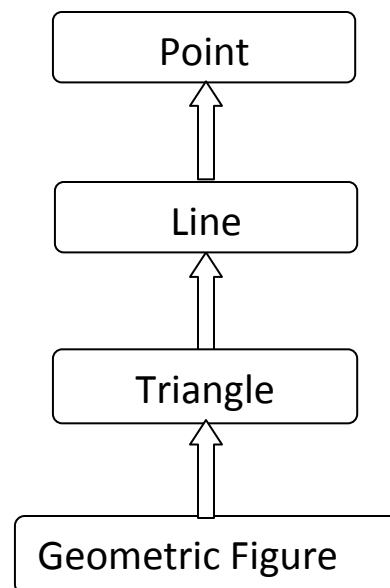
```
Class classname1 extends classname2 {  
... // Attributes and behavior specific to classname1  
}
```

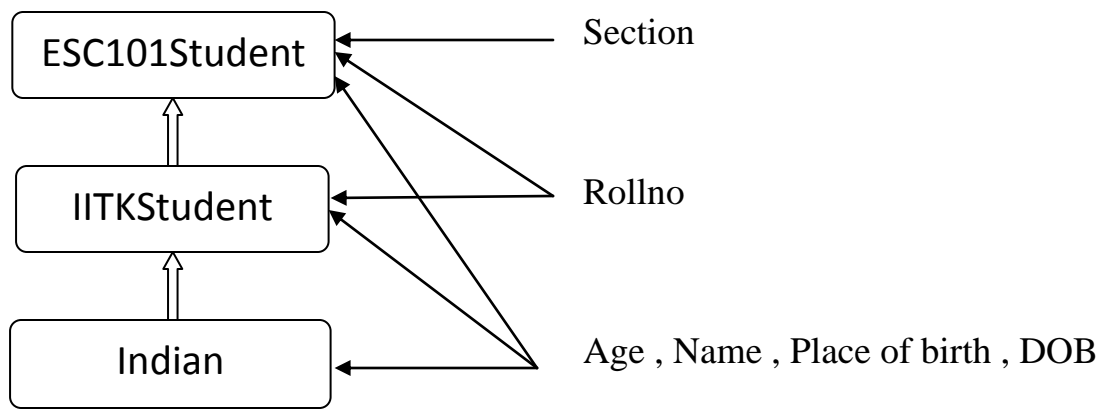
Here an object of classname1 is also an object of classname2.

Class Diagram



Example





Example code is as follows:

```

Class Indian{
    String name;
    int age;
    Indian(){
        ...
    }
}
Class IITKStudent extends Indian{
    int Rollno;
    IITKStudent(){
        ...
    }
}
Class ESC101Student extends IITKStudent{
    String Section;
    ESC101Student(){
        ...
    }
}
  
```

Now lets take an object of class Indian.

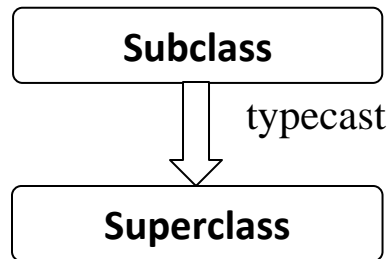
```

Indian O;
O = new Indian();
O = new IITKStudent();
O = new ESC101Student();
  
```

All above statements are correct.

## Type casting:

Subclass object is type cast to superclass object.



```
o = new Indian();
o.Name;    //Correct
o.Section  //Incorrect
o.Rollno;  //Incorrect
ESC101Student e;
e = o;     // if o is not object of ESC101Student then this results in error
o = e;     //Correct for any object type of o because of type cast
```

$L = R \Rightarrow L = (\text{type of } L)R.$

This means evaluate R and do an implicit type cast of L and then store evaluated R to L. Thus order of evaluation is from right to left.

For Example ,

```
Indian o = new IITKStudent();
```

Here 3 operations are performed.

- Memory allocation for new object
- type cast to class 'Indian'
- call to constructor

Implicit type cast is not done if there is loss of information.

e.g.

```
int x = 3.2/2.0;    //incorrect
int x = (int)(3.2/2.0); //correct
```

An Object of subclass can be implicitly converted to that of superclass.

```
Indian o = new IITKStudent(); // implicit type cast to 'Indian'  
o.RollNo                       //Incorrect  
((IITKStudent)o).RollNo;       //Correct
```

Now Consider following example.

```
O = new ESC101Student();  
((IITKStudent)o).RollNo;       //Correct  
((IITKStudent)o).Scetion;      //Incorrect  
((ESC101Student)o).Section;    //Correct
```