Abstract classes

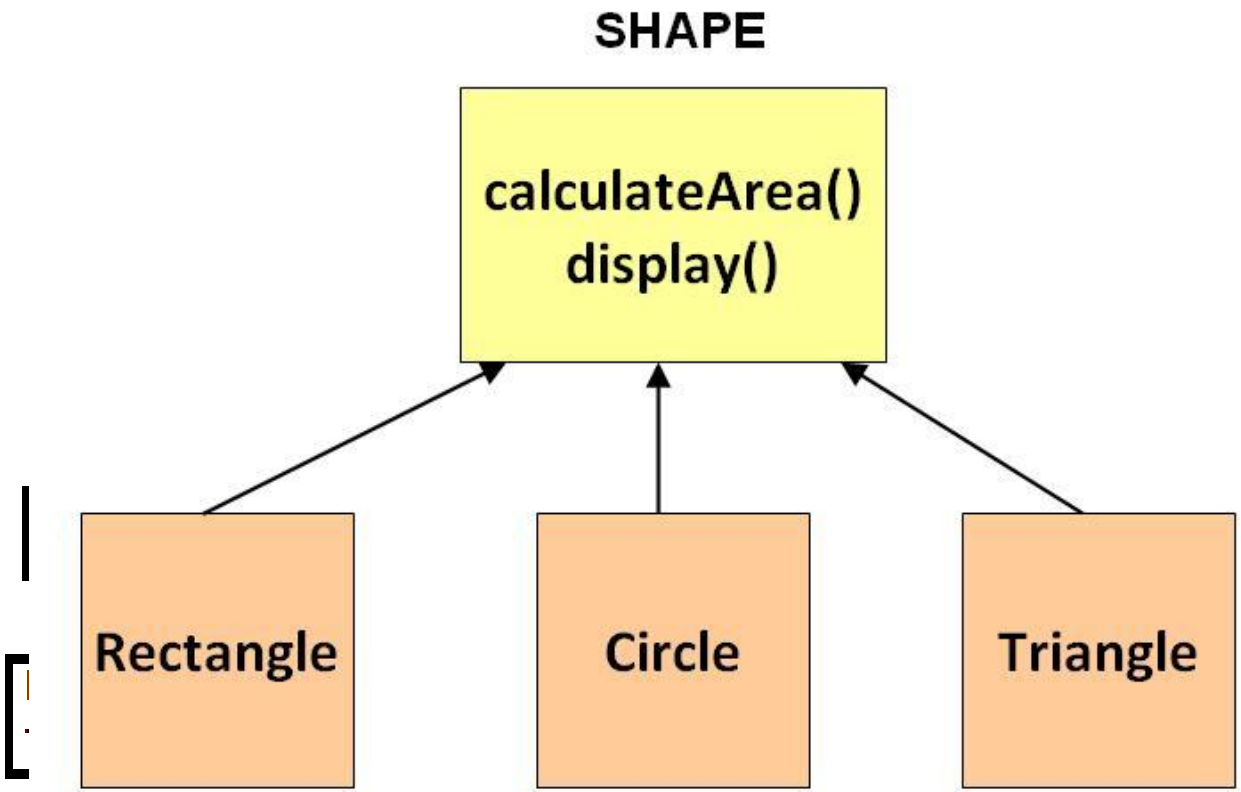
- A class which is not fully described means at least one method definition is not known to the developer at the time of define that class is known as abstract class.

- In other way we can say that a class which can not be instantiated or we can not create an object of a class is known as abstract class.

Abstract class in Java

A class which contains the **abstract** keyword in its declaration is known as abstract class.

- Abstract classes may or may not contain *abstract methods*, i.e., methods without body (public void get();)
- But, if a class has at least one abstract method, then the class **must** be declared abstract.
- If a class is declared abstract, it cannot be instantiated.
- To use an abstract class, you have to inherit it from another class, provide implementations to the abstract methods in it.
- If you inherit an abstract class, you have to provide implementations to all the abstract methods in it.


SHAPE
calculateArea()
display()
Rectangle
Circle
Triangle

```
abstract class Shape
   {
       abstract double calArea();
    }
       class Rectangle extends Shape
   {
      double length;
      double width;
      public Rectangle(double length, double width)
      {
           this.length = length;
           this.width  = width;
      }
     public calArea()
       { return length*width;
       }
   }
   class Circle extends Shape
   {
      double radius;
      public Circle(double r)
       {
          radius = r;
       }
```

```
public double calArea()
    {
        return 3.14*radius*radius;
    }
  }

class TestAbstraction
  {
    public static void main(String args[]){
    Shape s=new Circle(3.5);
    System.out.println(s.calArea());
    s = new Rectangle(3,5);
    System.out.println(s.calArea());
  }
  }
```

- An abstract class can have data member, abstract method, method body, constructor and even main() method.
- Constructor of abstract class is called when an instance of a inherited class is created.
- Abstract classes can also have final methods

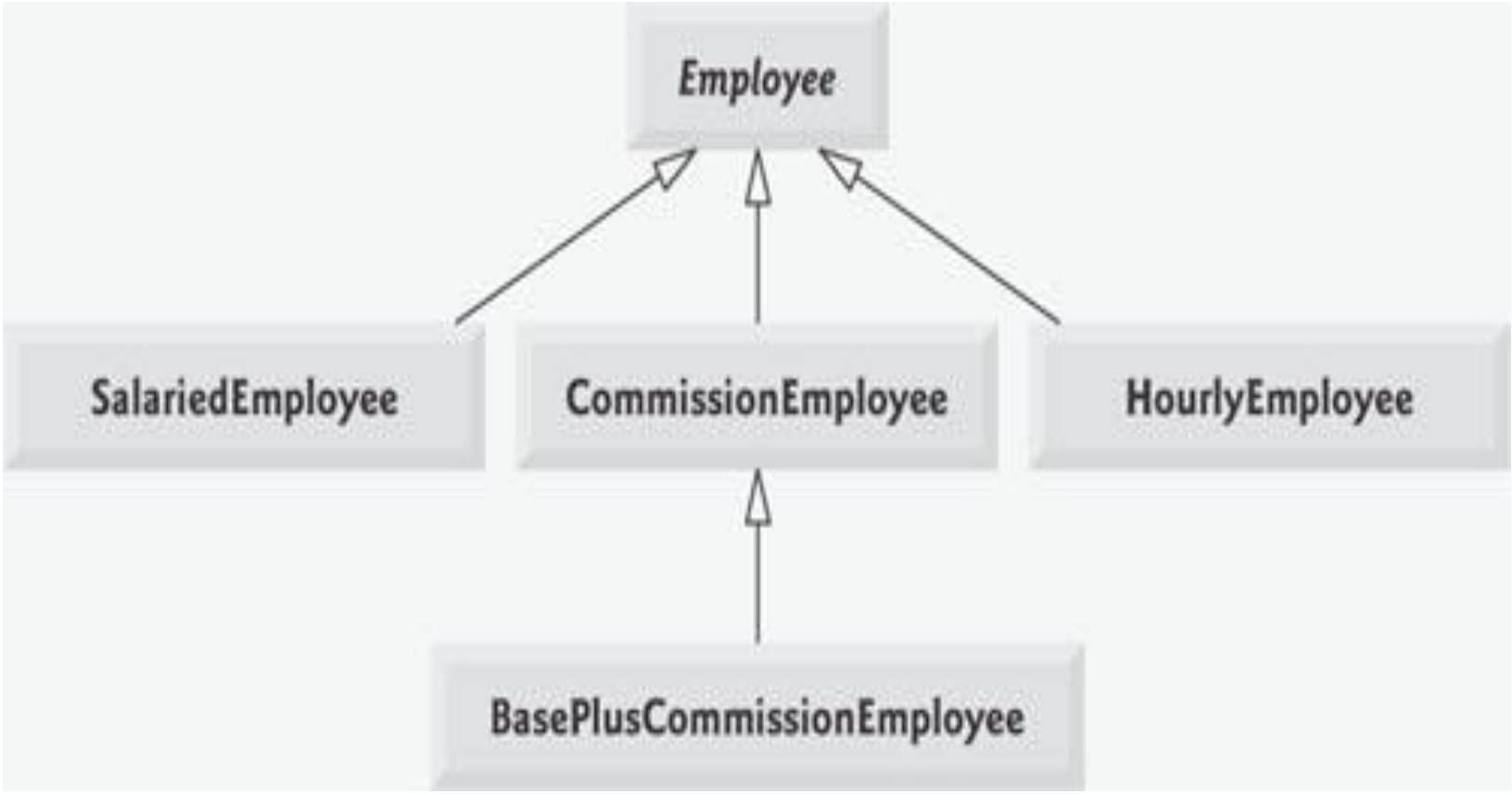
- Is it possible to create abstract and final class in Java?
- Is it possible to have an abstract method in a final class?
- Is it possible to inherit from multiple abstract classes in Java?

Abstract Methods in Java

- If you want a class to contain a particular method but you want the actual implementation of that method by child classes, you can declare the method in the parent class as an abstract.
- **abstract** keyword is used to declare the method as abstract.
- Declaring a method as abstract has two consequences −
  - The class containing it must be declared as abstract.
  - Any class inheriting the current class must either override the abstract method or declare itself as abstract.

A shopping mall has four types of employees and their pay structure is given below:

- Salaried employees are paid a fixed weekly salary regardless of the number of hours worked. Hourly employees are paid by the hour and receive overtime pay for all hours worked in excess of 40 hours. Commission employees are paid a percentage of their sales. Salaried-commission employees receive a base salary plus a percentage of their sales.
- For the current pay period, the shopping mall has decided to reward salaried-commission employees by adding 10% to their base salaries. Develop a Java class hierarchy for the given case study:


Employee
SalariedEmployee
CommissionEmployee
HourlyEmployee
BasePlusCommissionEmployee