

```
enum in java :  
-----  
* enum is a keyword in java which is introduced from JDK 1.5V.  
  
* It is similar to a class because .class file will be generated by the java compiler.  
  
* An enum is used to declare universal constants, all the universal constants must be separated by ; (comma) and ; (semicolon) is optional at the end.  
Example :  
public enum Color  
{  
    RED, BLUE, BLACK //public + static + final  
}  
  
* All the enum constants are by default public, static and final.
```

* The compiler generated code for an enum :

Color.java

```
-----  
public enum Color  
{  
    RED, BLUE, BLACK  
}
```

Color.class

```
-----  
public final class Color extends java.lang.Enum  
{  
    public static final Color RED = new Color();  
    public static final Color BLUE = new Color();  
    public static final Color BLACK = new Color();  
}
```

* An enum we can define inside a class, outside of the class as well as inside the method also.

* In the compiler generated .class file the java compiler **automatically add** a static method called **values()** to fetch all the enum constants, return type of this method is enum [] (enum array).
public static enum[] values();

* Every enum object contains "**order position**" which starts from 0, we can find this order position by using a predefined non static method called **ordinal()**
public final int ordinal();

* Every enum by default extends from **java.lang.Enum** class so an enum can't extend another class but It can implement multiple interfaces.

* Every enum is implicitly **final** so a class cannot extend an enum.

* Inside an enum we can write static variable, non static variable, static method, non static method, instance block, constructor and static block but all these members must be declared after enum constant because THE FIRST LINE OF AN ENUM IS RESERVED FOR ENUM CONSTANT ONLY and in this case ; is compulsory.

* Enum constructor is always private or default only.

* At the time of loading the enum constants which are by default public, static and final, appropriate constructor will be automatically executed.

* Every enum constant is by default object of type enum.

* We can pass enum in switch statement.

* In order to compare enum constant we can use == operator OR equals() method which is final so we cannot override, actually It is internally using == operator

* In **java.lang.Enum** class , name(String) and ordinal(int) both are blank final field so initialized through constructor. name(), ordinal() both are final so we cannot override that means once we set the name and order position of an enum constant then It is not modifiable.

Enum.java

```
-----  
public abstract class Enum<E extends Enum<E>> implements Comparable<E>, Serializable  
{  
    private final String name;  
    private final int ordinal;  
  
    protected Enum(String name, int ordinal)  
    {  
        this.name = name;  
        this.ordinal = ordinal;  
    }  
  
    public final String name()  
    {  
        return name;  
    }  
  
    public final int ordinal()  
    {  
        return ordinal;  
    }  
}
```