An enum is a special "class" that represents a group of **constants** like final.

An enum is just like class only difference that enum  cannot be used to create objects(no use **new** keyword), and it cannot extend other classes (but it can implement interfaces).

1.The **Enum in Java** is a user defined data type which contains a fixed set of constants.

2.Enums are used to create our own data type like classes.

3.Unlike C/C++, enum in Java is more powerful. Here, **we can define an enum either inside the class or outside the class.**

**enum** Season { WINTER, SPRING, SUMMER, FALL }

**Or**,

**enum** Season { WINTER, SPRING, SUMMER, FALL; }  //; is optional

It can be used for days of the week (SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, and SATURDAY) , directions (NORTH, SOUTH, EAST, and WEST), season (SPRING, SUMMER, WINTER, and AUTUMN or FALL), colors (RED, YELLOW, BLUE, GREEN, WHITE, and BLACK) etc.

Example:1

**class** EnumExample3{

**enum** Season

{ WINTER, SPRING, SUMMER, FALL; }//semicolon(;) is optional here

**public** **static** **void** main(String[] args)

 {

Season s=Season.WINTER;//enum type is required to access WINTER

System.out.println(s);

}

}

Example:2

**class** EnumExample4

{

**enum** Season

{

WINTER(5), SPRING(10), SUMMER(15), FALL(20);

**private** **int** value;

**private** Season(**int** value)

{

**this**.value=value;

}

}

**public** **static** **void** main(String args[]){

**for** (Season s : Season.values())

System.out.println(s+" "+s.value);

}}

Output:

WINTER 5

SPRING 10

SUMMER 15

FALL 20

If you put main() method inside the enum, you can run the enum directly.

**enum** Season {

WINTER, SPRING, SUMMER, FALL;

**public** **static** **void** main(String[] args) {

Season s=Season.WINTER;

System.out.println(s);

}

}

Output:

WINTER