Enum Lab

```
public class Problem1 {
    public static void main(String[] args) {
        MyEnums[] myEnums = new MyEnums[4];
        for (int i = 0; i < myEnums.length; i++)
        {
            System.out.println(myEnums[i]);
        }
    }
}
enum MyEnums
{
    FIRST("FIRST"), SECOND("SECOND"), THIRD("THIRD"), FOURTH("FOURTH");
    private final String name;
    MyEnums(String name){
        this.name = name;
    }
    public String getName() {
        return name;
    }
}
================================================================================
=======================================
public class Problem2 {
 public static void main (String args[]) {
 Set<Day> mySet2 = new TreeSet<Day>();
 mySet2.add(Day.SATURDAY);
 mySet2.add(Day.WEDNESDAY);
 mySet2.add(Day.FRIDAY);
 mySet2.add(Day.WEDNESDAY);
 for(Day d: mySet2){
 System.out.println(d);
 }
 /*Set<Day> mySet1 = new HashSet<Day>();
 mySet1.add(Day.SATURDAY);
 mySet1.add(Day.WEDNESDAY);
 mySet1.add(Day.FRIDAY);
 mySet1.add(Day.WEDNESDAY);
 for(Day d: mySet1){
 System.out.println(d);
 }*/
 }
}
================================================================================
=======================================
public class Problem3 {
 public static void main(String[] args) {
 //write code here to print CAT, LION, SNAKE,
 // and " DOG makes sound: bark "
 for (Animal animal : EnumSet.range(Animal.CAT, Animal.SNAKE)) {
 System.out.println(animal + ",");
 }
 String str = "DOG";
 Animal animal1 = Animal.valueOf(Animal.class, str);
 System.out.println(animal1 + " makes sound: " + animal1.sound());
 }
}
enum Animal {
 DOG { String sound() { return "bark"; } },
 CAT { String sound() { return "meow"; } },
 LION { String sound() { return "roar"; } },
 SNAKE { String sound() { return "hiss"; } };
```

```
 abstract String sound();
}
================================================================================
=======================================
// First Problem

enum EnumOne
{
   GREEN,
   YELLOW{
      public void info(){
         System.out.println("Warning Color");
      }
   },
   RED;

   public void info(){
      System.out.println("Signal Colors");
   }
}

public class NewClass
{
   public static void main(String[] args)
   {
      EnumOne[] eo = EnumOne.values();
      for (EnumOne e : eo){
        e.info();
      }
   }
}

/**
 * Singal Color
 * Warning Color
 * Signal Color
 *  */



// Second Problem

enum Levels
{
   TOP, MEDIEUM(10), BOTTOM(20, 30);
   int i, j;
   private Levels()
   {
   }
   private Levels(int i)
   {
      this.i = i;
   }
```

```
   private Levels(int i, int j)
   {
      this.i = i;
      this.j = j;
   }
}
public class MainClass
{
   public static void main(String[] args)
   {
      int[] val = new int[6];
      val[0] = Levels.TOP.i;
      val[1] = Levels.TOP.j;
      val[2] = Levels.MEDIEUM.j;
      val[3] = Levels.MEDIEUM.i;
      val[4] = Levels.BOTTOM.i;
      val[5] = Levels.BOTTOM.j;

      for (int n : val){
         System.out.println(n);
      }
   }
}

/**
 * 0
 * 0
 * 0
 * 10
 * 20
 * 30
 * */
```

```
package Lab_22_03_18;

/*
* enum can contain constructor and it is executed separately for each enum
constant
* We can’t create enum objects explicitly and hence
*enum can contain concrete methods only i.e. no any abstract method.
* */

enum carManufactural {

    TOYOTA, TESLA, BMW, HUNDAYUI;

    // write constructors

    private carManufactural() {
        System.out.println("Constructor called for : " +
                this.toString());
    }

    public void carManufacturalInfo() {
        System.out.println("Universal popular car");
```

```
    }
}

public class labOne {
    // write a method for print output.
    // print TOYOTA, TESLA, BMW, HUNDAYUI
    //TESLA
    //Universal popular car
    public static void main(String[] args) {
        carManufactural c1 = carManufactural.TESLA;
        System.out.println(c1);
        c1.carManufacturalInfo();


    }


}
```

```
package Lab_22_03_18;

//use enum valueOf() function and convert a String to an Enum object.
        enum Day {
    SUNDAY, MONDAY, TUESDAY, WEDNESDAY,
    THURSDAY, FRIDAY, SATURDAY;
}
public class LabTwo {
        public static void main(String[] args){
            String day = "SUNDAY";
            Day dayEnum = Day.valueOf(day);
            System.out.println(dayEnum);

            //Print day index number
            Day arr[] = Day.values();
            for (Day d : arr){
                System.out.println(d + " at index " + d.ordinal());
            }

        }

}
```

```
package Lab_22_03_18;

public class Lab1 {
    enum Day{ sunday, monday, tuesday, wednesday, thusday, friday, saterday}
    public static void main(String args[]){
        Day day=Day.monday;
        switch(day){
            case sunday:
                System.out.println("sunday");
                break;
            case monday:
                System.out.println("monday");
                break;
            default:
                System.out.println("other day");
```

```
        }
    }
}
```

==========enum Color

```
{
    RED, GREEN, BLUE;
    private Color()
    {
        /*write code without any kind of  loop to  output*/
//                RED
//                GREEN
//                 BLUE
    }
}
public class lab3
{
    // Driver method
    public static void main(String[] args)
    {
        Color c1 = Color.RED;
    }
}
```

Ans System.out.println(this.toString());

```
enum Enums
{
    RED, BLUE ;
    {
        System.out.println("RED");
    }
    static
    {
        System.out.println("BLUE");
    }
    private Enums()
    {
        System.out.println("Green");
    }
}
public class Lab6
{
    public static void main(String[] args)
    {
        Enum en = Enums.BLUE;
    }
    //  RED
}
```

output :

```
RED
Green
RED
Green
BLUE
```

```java
/*  output :
        WINTER 5
        SPRING 10
        SUMMER 15
        FALL 20
*/
public class Lab2 {
    enum Season {
        WINTER(5.68), SPRING(10.87), SUMMER(15.08), FALL(20.98);
        Season(double v) {
            this.value = (int) v;
        }
        public int value;
    }
    public static void main(String args[]){
        for (Season s : Season.values())
            System.out.println(s+" "+s.value);
    }
}
```
