**LAB enum.1**

*Write an enum for days-of-week (Sunday to Saturday), say EnumDay; now in EnumPractice’s main method use for loop and EnumClass.value() method to print all the values of EnumDay*

**Steps:**

* Write a enum – EnumDay, having days-of-weeks as values
* Write a EnumPractice class, having p.s.v.main method
* Inside main method write for loop to display all enum type values of EnumDay

**ENUM**

**package** cg.javaflp.practice;

/\*\*

\* An enum type is a type whose fields consist of a fixed set of constants

\*/

**public** **enum** EnumDay {

*MONDAY*, *TUESDAY*, *WEDNESDAY*, *THURSDAY*, *FRIDAY*, *SATURDAY*, *SUNDAY*

}

**ENUM PRACTICE CLASS**

**package** cg.javaflp.practice;

**public** **class** EnumPractice {

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

**for** (EnumDay d : EnumDay.*values*() ){

System.*out*.println(d);

}

}

}

**Learning’s:**

* How to write enum
* Use of for loop in case of enum

**LAB enum.2**

*Adding to Lab x.1; now in EnumPractice main method call a method which tell about a day (this method uses switch block to do so) – pass enum type as key to the switch block*

**Steps:**

* In EnumPractice class, write a method tellAboutDay, which takes enum object as parameter
* Write a switch-case block, and use this enum type as key of switch

**package** cg.javaflp.practice;

**public** **class** EnumPractice {

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

EnumPractice practice = **new** EnumPractice();

// calling tellAboutDay method and passing EnumDay object to it

practice.tellAboutDay(EnumDay.*WEDNESDAY*);

}

**private** **void** tellAboutDay(EnumDay day) {

**switch** (day) {

**case** *MONDAY*:

System.*out*.println("Monday is bad day");

**break**;

**case** *TUESDAY*:

System.*out*.println("Tuesday is waste day");

**break**;

**case** *WEDNESDAY*:

System.*out*.println("Wednesday is work day");

**break**;

**case** *THURSDAY*:

System.*out*.println("Monday is thrust day");

**break**;

**case** *FRIDAY*:

System.*out*.println("Friday is fun day");

**break**;

**case** *SATURDAY*:

System.*out*.println("Saturday night full tight");

**break**;

**case** *SUNDAY*:

System.*out*.println("Sunday is sleep day");

**break**;

**default**:

System.*out*.println("Not a day");

**break**;

}

}

}

**Learning’s:**

* How to use enum type in case of switch

**LAB enum.3**

*Write a method inside an enum and call this method from main class.*

**ENUM**

**package** cg.javaflp.practice;

**public** **enum** EnumDay {

*MONDAY*, *TUESDAY*, *WEDNESDAY*, *THURSDAY*, *FRIDAY*, *SATURDAY*, *SUNDAY*;

**public** **void** display(){

System.*out*.println("MUAHAHAHAHAH....");

}

}

**ENUM PRACTICE**

**public** **class** EnumPractice {

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

EnumDay obj = EnumDay.*MONDAY*;

obj.display();

}

}

**Learning’s:**

* How to call method of enum

**LAB enum.4**

*Write an enum inside a class, use this enum from outside this class*

**Steps:**

* Write a class (InnerEnum), then write a enum (Direction) inside it
* Create object of this enum and use it

**A CLASS HAVING AN INNER ENUM**

**package** cg.javaflp.practice;

**public** **class** InnerEnum {

**public** **enum** Direction{

*NORTH*, *SOUTH*, *EAST*, *WEST*

}

}

**ENUM PRACTICE**

**public** **class** EnumPractice {

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

/\* FOR FOURTH ASSIGNMENT \*/

InnerEnum.Direction direction = InnerEnum.Direction.*EAST*;

}

}

**Learning’s:**

* Understand how to create, initialize, populate, and use a multidimensional array

**LAB enum.5**

*Write an enum inside a method, and use this enum inside this method only*

CODE WILL SHOW COMPILE TIME ERROR

**Learning’s:**

* Enum can be inner but cannot be local