CS5000: Foundations of Programming

Mingon Kang, PhD
Computer Science, Kennesaw State University
Enumerated Types

- Instead of using integers to represent a set of values, a type with a restricted set of values is used.
- If we define the colors, which are
  - Red
  - Orange
  - Yellow
  - Green
  - Blue
  - Indigo
  - Violet

We might use `final int` to specify the values.
Enumerated Types

- For example,

```java
final int ROCK = 0;
final int PAPER = 1;
final int SCISSORS = 2;

final int MONDAY = 0;
final int TUESDAY = 1;
...
```
Enumerated Types

- Named constants as a group

```c
enum TypeName {VALUE_1, VALUE_2, ..., VALUE_N};

e.g.,
enum WorkDay {MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY};
enum Rainbow{RED, ORANGE, YELLOW, GREEN, BLUE, INDIGO, VIOLET};
enum Weapon {ROCK, PAPER, SCISSORS};
```
Enumerated Types

- **Enum Declaration**

```java
TypeName variableName;
TypeName variableName = TypeName.CONSTANT;

e.g.,
WorkDay meetingDay;
Rainbow color = Rainbow.RED;
WorkDay availableDay = null; // null
```
Enumerated Types

- Enum type can be displayed using print(ln)

```java
System.out.println(color);
Will display the following output:
    RED   (O)
    Color.RED (X)
```

- But, enum is not string.
Enumerated Types

- Comparison
  - Either `equals` method and `==` operator can be used

```java
if (color == Rainbow.RED)
    System.out.println("The color is RED");

Switch (color){
    case RED: STATEMENT1; break;
    case OORAGE: STATEMENT2; break;
    break;
}
```
Enumerated Types

- A static method `values()` returns an array whose elements are the values of the enum type.

```java
Rainbow[] colors = Rainbow.values();
for (int i = 0; i < colors.length; i++)
    System.out.println(colors[i]);
```
Enumerated Types

- An enum variable can be set by using the static method `valueOf()` to convert an input string to a value of the enum type.

```java
Scanner keyboard = new Scanner(System.in);
String answer = keyboard.next();
Rainbow color = Rainbow.valueOf(answer);
```