Defining Enumerated Types

Here is the definition of the `Weekday` type mentioned at the beginning of the lesson. The C++ compiler assigns values to the names. **`sunday`** is assigned **`0`**, **`monday`** is assigned **`1`**, and so on.

```
enum class Weekday
{
    sunday, monday, tuesday, wednesday, thursday,
    friday, saturday
};
```

You may also **explicitly specify** the underlying integer value used to represent any or all of the literals of an enumerated type. For example, the **`Coin`** type represents U.S. coins where each literal is equal to the monetary value of that coin.

```
enum class Coin
{
    penny = 1, nickel = 5, dime = 10,
    quarter = 25, halfDollar = 50, dollar = 100
};
```

If you supply initializers for some values but not others, the compiler will automatically number the remaining literals consecutively after the last.

```
enum class Month
{
    jan = 1, feb, mar, apr, may, jun, jul, aug,
    sep, oct, nov, dec
};
```

Here, **`jan`** has the value **`1`**, **`feb`** has the value **`2`**, and so forth up to **`dec`**, which is **`12`**.

*In previous semesters, the Course Reader used **`UPPER_CASE`** for the enumerated elements, since that is the convention in Java. Looking at several style guides (both Google and the Core Guidelines), we've switched to lower-camelCase in this edition.*