

# Enumerated Output

The names of the enumerated values **are not** strings; you cannot print them:

```
Month m1{Month::jan};  
cout << m1 << endl; // does not compile
```

Since enumerations are constant integral scalar values, you **can** use **enum** variables as **switch** selectors. Thus you could write a **to\_string()** function like this:

```
string to_string(Month m)  
{  
    switch (m)  
    {  
        case Month::jan: return "January";  
        case Month::feb: return "February";  
        case Month::mar: return "March";  
        . . .  
        default: return "INVALID MONTH";  
    }  
}
```

This function converts a **Month** variable to a **string** so you can print it or concatenated it.

- Enumerated types are internally just integers: **pass them by value**.
- Each **case** label must use the **fully qualified enumeration literal**.
- The **default** case returns an error if **m** does not match any **Month**. You may want to use an assertion instead, since it is definitely a programming error if the **default** is ever reached.



This course content is offered under a CC Attribution Non-Commercial license. Content in this course can be considered under this license unless otherwise noted.