CS 3460

Introduction to Enumerations

Enumerations

- Enumeration
  - Ability to define legal values for a type
- Two types of enumerations; very similar to Java
  - Unscoped Enumeration (legacy – don’t use)
  - Scoped Enumeration (new C++ goodness)

Unscoped Enumeration (legacy)

```
enum City
{
    NewYork,
    Chicago,
    Denver,
    LosAngeles
};
```

- Specify name & values
- Note the switch statement directly uses the values, no *scoping* to the type
- Underlying type is integral

```
int main()
{
    City thisCity = Chicago;

    switch (thisCity)
    {
        case NewYork:    std::cout << "New York" << std::endl; break;
        case Chicago:    std::cout << "Chicago" << std::endl; break;
        case Denver:     std::cout << "Denver" << std::endl; break;
        case LosAngeles: std::cout << "Los Angeles" << std::endl; break;
    }

    return 0;
}
```

Unscoped Enumeration (legacy)

```
enum City
{
    NewYork         = 0,
    Chicago         = 1,
    Denver          = 2,
    LosAngeles      = 3
};
```

- Can specify the integer for the values
- Uninitialized values are set by the compiler

```
int main()
{
    City thisCity = Chicago;

    switch (thisCity)
    {
        case NewYork:    std::cout << "New York" << std::endl; break;
        case Chicago:    std::cout << "Chicago" << std::endl; break;
        case Denver:     std::cout << "Denver" << std::endl; break;
        case LosAngeles: std::cout << "Los Angeles" << std::endl; break;
    }

    return 0;
}
```

Scoped Enumeration

```
enum class State
{
    Alabama,
    Arizona,
    Hawaii,
    Kansas
};
```

- Specify name & values
- Note the switch statement is *scoped* to the type.
- Underlying type is integral

```
int main()
{
    State thisState = State::Alabama;

    switch (thisState)
    {
        case State::Alabama:    std::cout << "Alabama" << std::endl; break;
        case State::Arizona:    std::cout << "Arizona" << std::endl; break;
        case State::Hawaii:     std::cout << "Hawaii" << std::endl; break;
        case State::Kansas:     std::cout << "Kansas" << std::endl; break;
    }


    return 0;
}
```

Scoped Enumeration

```
enum class State : unsigned char
{
    Alabama  = 0,
    Arizona  = 1,
    Hawaii   = 2,
    Kansas   = 3
};
```

- Can specify the type and integer for the values
- Uninitialized values are set by the compiler

```
int main()
{
    State thisState = State::Alabama;

    switch (thisState)
    {
        case State::Alabama:    std::cout << "Alabama" << std::endl; break;
        case State::Arizona:    std::cout << "Arizona" << std::endl; break;
        case State::Hawaii:     std::cout << "Hawaii" << std::endl; break;
        case State::Kansas:     std::cout << "Kansas" << std::endl; break;
    }


    return 0;
}
```