Regular Increment/Decrement

```cpp
int value { 5 };

//Increment by one
value = value + 1;
std::cout << "The value is : " << value << std::endl;

value = 5; // Reset value to 5

//Decrement by one
value = value - 1;
std::cout << "The value is : " << value << std::endl;
```

```cpp
value = 5;

std::cout << "The value is (incrementing) : " << value++ << std::endl; //5
std::cout << "The value is : " << value << std::endl; //6


std::cout << std::endl;

//Decrement with postfix

//Reset value to 5
value = 5;

std::cout << "The value is (decrementing) : " << value-- << std::endl;//5
std::cout << "The value is : " << value << std::endl; //4
```

Prefix Increment/Decrement

```cpp
value = 5;

++value;
std::cout << "The value is (prefix++) : " << value << std::endl; //6

//Reset value to 5
value = 5;
std::cout << "The value is (prefix++ in place) : " << ++value << std::endl;//6

std::cout << std::endl;

//Prefix : Decrementing

//Reset value to 5;
value = 5;
--value;
std::cout << "The value is (prefix--) : " << value << std::endl;//4

//Reset value to 5;
value = 5;
std::cout << "The value is (prefix-- in place) : " << --value << std::endl;//4
```

- Prefix and postfix increment/decrement operators are cool.

- But they only increment by one, what if we need to increment by a value other than 1, say 5 or 7. We'll see a way to do that in the next lecture

- There is no   value** or **value , or //value or value//. They don't make sense, these prefix/postfix operators are only available for + and -, again because that's where they make sense