

# C++ Notes

Ch5, Ch6

# Overflow

- Overflow occurs when a data value becomes too large for its data type:
- Consider this program:  

```
unsigned short x = 0;  
x = 50000 + 20000;  
cout << x;
```
- Think about what you would *expect* to see from the cout statement.

# Assignment

- Enter the code on the previous slide
- Write the **result** on a small piece of paper and give it to Mr. O.
- The first correct answer gets full credit, the second correct answer gets slightly less, the third gets slightly less...etc.

# Why the strange answer?

- $50000 + 20000 = 70000$ , so why is that not the answer.
- Overflow:
  - An UNSIGNED SHORT integer can store values between 0 and 65535
  - Once the value becomes more than 65535, the rest of the number (the extra amount) *wraps around* to the other end of the storage queue.



# Underflow

- Same concept as overflow...value becomes too small for the data type, so flows to the other end of the queue
- Try this code:

```
unsigned short x = 0;
```

```
x = x - 15000;
```

```
cout << x;
```

# Notes: Ch6

Strings

# How to use strings in C++

- When using strings, our compiler requires a header file
  - `#include <string.h>` (the book shows `ostring.h` instead)
- Declare a string like a variable
  - Ex: `string first_name;`
- After that, things work similar to QB

# String Operations / Function

- Concatenation
  - `String3 = string1 + string2`
  - Pieces `string2` on the end of `string1`
- Length
  - `X = string1.length();`
  - Finds the length of `string1`, stores in `x`
  - `X` must be integer type (or short)