The website for these SI sessions is `https://github.com/benblazak/2014-fall-si-cpsc120`.

Many of these examples are from `https://github.com/benblazak/2014-spring-si-cpsc120`, which is full of stuff I wrote for a lab last semester. If you're looking for extra practice, this is one of many places you might start.

Quick Note on Identifiers

Of the five identifiers below, which one do you like most?

```
int a = 5;
int bob = 5;
int employee = 5;
int employee_age = 5;
int employees_age_when_they_talked_to_us_at_the_job_fair_last_week = 5;
```

At which point did you figure out what the variable was supposed to represent?

Basic Data Types and Integer Representation

- List all the data types you can think of right now.

- What are they good for?

- Why do we have data types, anyway?

- What is the range of an 8 bit unsigned integer?

- What is the range of an 8 bit signed integer?

- What is the range of a 32 bit signed integer?

What happens when an integer "overflows"?

```
#include <iostream>
using std::cout;
using std::endl;

int main() {
             int a = 2147483647;  // 2^31-1 = 2147483647
    unsigned int b = 4294967295;  // 2^32-1 = 4294967295

    cout << "a   = " << a   << endl;
    cout << "a+1 = " << a+1 << endl;
    cout << "b   = " << b   << endl;
    cout << "b+1 = " << b+1 << endl;

    return 0;
}
```

```
a   = 2147483647
a+1 = -2147483648
b   = 4294967295
b+1 = 0
```

We can input binary numbers directly?

```cpp
#include <iostream>
using std::cout;
using std::endl;

int main() {
    cout << "0b0000 == " << 0b0000 << endl;
    cout << "0b0001 == " << 0b0001 << endl;
    cout << "0b0010 == " << 0b0010 << endl;
    cout << "0b0011 == " << 0b0011 << endl;
    cout << "0b0100 == " << 0b0100 << endl;
    cout << "0b0101 == " << 0b0101 << endl;
    cout << "0b0110 == " << 0b0110 << endl;
    cout << "0b0111 == " << 0b0111 << endl;

    cout << endl;

    cout << "0b00001111 == " << 0b00001111 << endl;
    cout << "0b00011111 == " << 0b00011111 << endl;
    cout << "0b00111111 == " << 0b00111111 << endl;
    cout << "0b01111111 == " << 0b01111111 << endl;
    cout << "0b11111111 == " << 0b11111111 << endl;

    return 0;
}
```

```
0b0000 == 0
0b0001 == 1
0b0010 == 2
0b0011 == 3
0b0100 == 4
0b0101 == 5
0b0110 == 6
0b0111 == 7

0b00001111 == 15
0b00011111 == 31
0b00111111 == 63
0b01111111 == 127
0b11111111 == 255
```

Assignment

What do you think the output of this program will be? (Answer on the last page.)

```
#include <iostream>
using std::cout;
using std::endl;

int main() {
    int x;
    int y;
    int z;

    x = y = z;
    cout << "x = " << x << ", y = " << y << ", z = " << z << endl;
    // be careful with this one

    x = y = z = 5;
    cout << "x = " << x << ", y = " << y << ", z = " << z << endl;

    x = 5;
    y = 7;
    z = 11;
    x = x + 1;
    y = y + 2;
    z = x + y + 3;
    cout << "x = " << x << ", y = " << y << ", z = " << z << endl;

    x = 5;
    y = 7;
    z = 11;
    z = y;
    y = x;
    x = z;
    z = 11;
    cout << "x = " << x << ", y = " << y << ", z = " << z << endl;
}
```

Program Design

Explain, at a high level, an algorithm for checking whether a string is a valid C++ identifier, given that you can only look at one character at a time.

Selected Answers

- Output for the "Assignment" program.

```
x = 1, y = 1, z = 1
x = 5, y = 5, z = 5
x = 6, y = 9, z = 18
x = 7, y = 5, z = 11
```