Reminders/Announcements

- MC-04 due tomorrow 6/19
- HW-02 out, due 6/26

DISCUSSION SESSION
WEEK 4

C++ REFERENCES

A reference variable is an alias, that is, another name for an already existing variable.

- A reference variable **must** be initialized at the time of declaration.

```
int a = 5, b = 6;
int& c;   // this is not valid
int& c = a;    // valid!
```

- Once a reference variable has been initialized, it **cannot** be changed to reference another variable.

```
c = b;  //  c is already referencing a
```

References are important in C++, because they give you the ability to manipulate/modify existing data in the computer's memory, rather than making new copies of the data. This helps to reduce code and ultimately improve code performance.

Applications of Reference

There are multiple applications for references in C++. A few of them are:

- Used to modify values of arguments in function calls. That is, you can use reference to get data out of a function without using the **return** keyword.
- Used to prevent/avoid duplicate data.
- Used in Range-based For loop to modify all objects.

```
void func(int& x, int y) {
    x /= 10;
    y -= 10;
}
int main( ) {
    int a = 200, b = 100;
    func(a, b);
    std::cout << a << " " << b << std::endl;
}
```

```
std::string city = "Boston";

for(char letter : city) {
    letter = 'a';
}

std::cout << city << std::endl;
```

```
std::string city = "Boston";

for(char& letter : city) {
    letter = 'a';
}

std::cout << city << std::endl;
```

```
int main( ) {
        int a = 5, b = 10, c = 15;
        int& x = a;
        int& y = b;
        int& z = c;
        x = y + c;
        b = pow(x, 2) / 25 * 2;
        z = y - a;
        x += c;
        y -= a;
        a *= 2;
}
```

What are the final values of **a**, **b** and **z** after this code executes?

```
void intermediateUpdate(int& a, int& b) {
        int& temp = a;
        a = b - temp;
        b += temp;
        temp++;
}
void func1(int& x, int& y) {
        int temp = x * y;
        x += temp;
        y -= temp;
        intermediateUpdate(x, y);
}
void func2(int& x, int y, int& z) {
        z = x * y;
        x *= z;
        intermediateUpdate(y, z);
}
```

```
int main( ) {
        int a = 2, b = 3, c = 4;
        func1(a, b);
        func2(a, b, c);
        std::cout << a << " " << b << " " << c << std::endl;
}
```