Week 5

Exercises:

Loops: for and while

Nested loops

New topic: LOOPS

- What is a loop?

New topic: LOOPS

- What is a loop?
  - A block of code that is executed **repeatedly**
  - Until a certain condition is met

- We need to be familiar with a few more operators for loops…

Compound Assignment Operators

+= **-=** **/=**

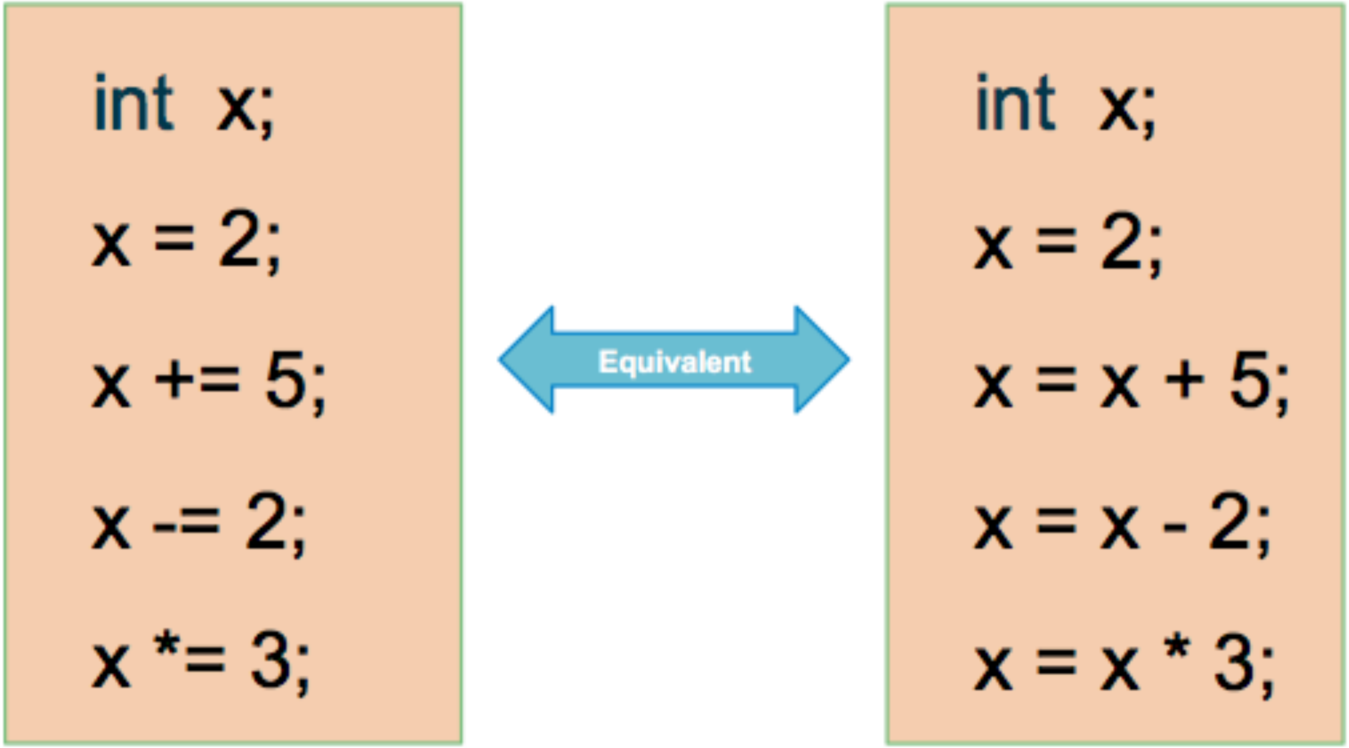
***=** **%=**

Variable = variable + (expression)

~~ is the same as~~

Variable += expression

Compound Assignment



```
int  x;
x = 2;
x += 5;
x -= 2;
x *= 3;
```

```
int  x;
x = 2;
x = x + 5;
x = x - 2;
x = x * 3;
```

++i and i++

- These are called the increment operators
- They increment i by one

- i++ happens after i does its job
  - This is called the post-increment operator

- ++i happens before i does its job
  - This is called the pre-increment operator

--i and i--

- These are called the decrement operators
- They decrement i by one

- i-- happens after i does its job
  - This is called the post-decrement operator

- --i happens before i does its job
  - This is called the pre-increment operator

While loops

- How would we write a while loop that prints the integers from 0 to 4?

While loops

```
int x = 0; //initialize x

while (x < 5){
        cout << x << endl;
        ++x;
}
```

While loops

```
while (x < 5){
        cout << x << endl;
        ++x;
}
```

- **The (x < 5) is the…**

While loops

```
while (x < 5){
        cout << x << endl;
        ++x;
}
```

- **The (x < 5) is the…** condition

While loops

```
while (x < 5){
        cout << x << endl;
        ++x;
}
```

- The (x < 5) is the… condition
- **The cout << x << endl is the…**

While loops

```
while (x < 5){
        cout << x << endl;
        ++x;
}
```

- The (x < 5) is the… condition
- **The cout << x << endl is the…** loop body

While loops

```
while (x < 5){
        cout << x << endl;
        ++x;
}
```

- The (x < 5) is the… condition
- The cout << x << endl is the… loop body
- **The ++x is the…**

While loops

```
while (x < 5){
        cout << x << endl;
        ++x;
}
```

- The (x < 5) is the… condition
- The cout << x << endl is the… loop body
- **The ++x is the…** update

While loops

```
int x = 0; //initialize x

while (x < 5){
        cout << ++x << endl;
        ~~++x;~~
}
```

What would happen if the increment happened as above?

While loops

```
int x = 0; //initialize x

while (x < 5){
        cout << ++x << endl;
        ~~++x;~~
}
```

What would happen if the increment happened as above?

increment x, and then print this new value of x

While loops

```
int x = 0; //initialize x

while (x < 5){
        cout << x++ << endl;
        ~~++x;~~
}
```

What would happen if the increment happened this way instead?

print x is it is, and then **increment** x

In groups: While loop exercises

- Print only evens from 0 – 100
- Print only odds from 0 – 100
- Count to 1000, printing only the numbers divisible by 50
- Print a 3 x 3 square of stars (*)
  - Hint: you need 3 loops, or nested loops if you've looked ahead!

String Literals

- Remember what a string made up of?
  - **A collection of chars**

- We can access each one of these chars

```
string name = “Jim”;
cout << name[0] << endl; //prints: J
```

Accessing characters of strings

```
string name = “Jim”;
cout << name[0] << endl; //prints: J
```

Strings are what we call: **0 indexed**

- The first char of the string is at **index 0**
- The second char of the string is at **index 1**
- **And so on…**

The index is inside the brackets: [1]

Adding Characters to Strings

- This is called concatenating

```
string name = “Jim”;
cout << name[0] << endl; //prints J

name += ‘m’; //adds m to the end of the string
name += ‘y’; //adds y to the end of the string

cout << name << endl; //prints Jimmy
```

Nested Loops

- Nested loops means one loop is inside another
- There is an outer loop, and then inner loops (can be many)
- Need to keep track of how the inner loops are controlled by the changes to the outer loop
- The inner loops will run many times, whereas the outer loop will run once through to completion
- Can you have a **for loop** inside a **while loop** and vice-versa?

Nested Loops

- Nested loops means one loop is inside another
- There is an outer loop, and then inner loops (can be many)
- Need to keep track of how the inner loops are controlled by the changes to the outer loop
- The inner loops will run many times, whereas the outer loop will run once through to completion
- Can you have a **for loop** inside a **while loop** and vice-versa? **YES**

In groups: Nested Loops Exercise

- Print a 3x3 square of * (star characters)
- **First with 3 loops (you can choose)**
- **Then with nested loops**

Let’s code some loops!

- **Pick some of the following to try out, and write each using a *while* loop and a *for* loop**
- Printing only odd or only even numbers from 0 to 100
- Count to 1000 by twos, and only print the numbers divisible by 50
- Count to 50 by 2s, then finish to 100 by 5s (two loops needed here)
- Print each letter of a string on a separate line until the string is done (tricky – feel free to ask)
- Any other fun ones you can think of!