CS 161
Intro to CS I

More Conditional Statements
and Begin Repetition

Oregon State University

Odds and Ends…

- Last week to demo Assignment #1!!!
- Demo Assignment #2
- Assignment #3 is posted
  - **Design Due on Canvas** by Sunday, 10/15

Nested for Loops

```
for(x = 0; x < 10; x++) {
    for(y = 0; y < 10; y++) {
        cout << “hello world\n”;
    }
}
```

- How many times is Hello World printed?

Reuse Variables

```
for(x = 0; x < 10; x++) {
    cout << “The value of x is: ” << x << endl;
}
for(x = 0; x < 10; x++) {
    cout << “The value of x is: ” << x << endl;
}
```

Variables with same name

```
int x;
for(x = 0; x < 10; x++) {
  for(x = 0; x < 10; x++) {
    cout << “The value of x is: ” << x << endl;
  }
}
```

- What is the output from this nested loop?

Infinite Loops

```
int x;
for(x = 0; x < 10; x++) {
  for(x = 0; x < 5; x++) {
    cout << “The value of x is: ” << x << endl;
  }
}
```

Infinite Loops

```
int x, y;
for(x = 0; x < 10; x++) {
  for(y = 0; y < 5; x++) {
     cout << “The value of x is: ” << x << endl;
   }
}
```

Infinite Loops

```
int x, y;
for(x = 0; x < 10; x++) {
   for(y = 0; x < 5; y++) {
      cout << “The value of x is: ” << x << endl;
   }
}
```

Infinite Loops

```
int x;
for(x = 1; x <= 10; x++) {
    cout << “The value of x is: ” << --x << endl;
}
```

Why is this good/bad?

```
for(int x = 0; x < 10; x++) {
    for(int y = 0; y < 10; y++) {
         cout << “hello world” << endl;
    }
}
```

- Where can we access x and y?

Looping Recap…

- for loops
  - Repeat for specific number of times
  - Example?
- while loops
  - Repeat while a condition is being met
  - Example?
- do while loops
  - Always do once, and repeat while condition is met
  - Example?

Infinite Loop Example…

How do we read a string of chars?

- User-defined type in string library

```
#include <string>
```

- Declare/Create type

```
string mssg;
```

- Read with cin or getline

```
cin >> mssg;  //get a word
getline(cin,mssg); //get a line of txt
```

Demo…

Finish C++ String Demo

- What does cin do when reading…
  - Int/Floats
  - Strings
- What does getline do?

Demo…

More about **break**, **exit**, and **return**

- **break** – used with switch and loops, breaking out of the closest associated case or loop(for, while, or do while). **This statement can only occur in a loop or case**, otherwise the compiler yells!

- **return** – leave the current function, which exits the program when in the main() function. You can put this **anywhere inside any function**, otherwise the compiler yells!

- **exit()** – exit the entire program, no matter where this is encountered. You can put this **anywhere inside any function, as long as you include <cstdlib>**, otherwise the compiler yells!