

CS 161

Intro to CS I

More Conditional Statements
and Begin Repetition

Odds and Ends...

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

int x;
int y;

Nested for Loops

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

← 10

←



X ≠ Y

100

- How many times is Hello World printed?

int x;

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;  
  }  
}
```

Handwritten annotations in red:

- ① above `x = 0` in the first `for` loop.
- ② above `x < 10` in the first `for` loop.
- ③ above `x = 0` in the second `for` loop.
- A red arrow points from the text "ctrl c" to the first `for` loop.
- A red arrow points from the number "4" to the `x++` in the first `for` loop.
- A red arrow points from the number "4" to the `x++` in the second `for` loop.
- A red arrow points from the number "0-4" to the `x` in the `cout` statement.

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;  
    }  
}
```

0 - 2 billionish
- 2 billionish ←

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;  
    }  
}
```

0
:
0

Infinite Loops

```
int x;
```

```
for(x = 1; x <= 10; x++) {
```

```
    cout << "The value of x is: " << --x << endl;
```

```
}
```

10 times
1-10

*x-1
better*

Why is this good/bad?

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

Initialization in loop

scope y

← you don't have access to y

- Where can we access x and y?

no access to x

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?

0 or more

1 or more

How do we read a string of chars?

- User-defined type in string library

#include <string>

C++ strings

- Declare/Create type

std::string mssg;

- Read with cin or getline

cin >> mssg; //get a word

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

Finish C++ String Demo

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

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int x;
6     float f;
7
8     //Not a good way to handle errors because 4.6, 4t, etc. will work
9     do {
10         if(cin.fail()) {
11             cin.clear(); //reset failbit
12             cin.ignore(256, '\n'); //ignore 256 chars or until newline
13         }
14         cout << "enter int: ";
15         //cin ignores leading whitespace and reads until char not of
16         //the type specified or whitespace (newline, space, tab, etc.)
17         cin >> x;
18         cout << "value of x: " << x << endl;
19     } while(cin.fail()); //fail bit set if it doesn't get type expected
20
21     cout << "enter a float: ";
22     cin >> f;
23     cout << f << endl;
24
25     return 0;
26 }
```