Oregon State University

COLLEGE OF ENGINEERING | School of Electrical Engineering and Computer Science

CS 161
Intro to CS I

Finish Conditionals/Begin Loops

Odds and Ends

- Assignment 3 Design due Sunday
- Back Evaluations due Sunday

Fun Friday...

Oregon State University
College of Engineering

- What if we didn't have a break in the case of a switch statement?
- How could you write an if/else to match not having breaks in this switch?

```
switch( x ) {
      case 0:
      case 1:
            std::cout << "X is zero or one\n";
      case 2:
            std::cout << "X is two\n";
      default:
            std::cout << "You have entered an invalid number!!!\n";
}
```

if (x==0 || x==1) {
all 3 cout }
else if (x==2) {
last 2 couts }
else {
}

if (exp1)
exp2
else
exp3

- What do you think this does in C++?

<expression1> ? <expression2> : <expression3>

T

F

ternary

(x==1 || x==0) ? cout .. : cout ...

The for Loop Examples

Oregon State University
College of Engineering

```
for(x=1; x <= 1; x++) {
   cout << “hello world\n”;
}
```

1

```
for(x=1; x < 1; x++) {
   cout << “hello world\n”;
}
```

0

0 or more times

int x;

The while loop

Oregon State University
College of Engineering

1 one time 2 4

```
for(x=1; x <= 100; x++)
    cout << "hello world\n"; 3
```

for(; ;)

vs.

0 or more

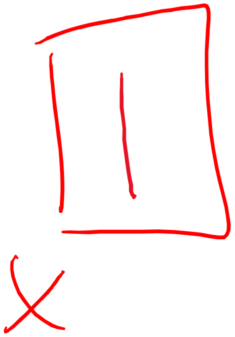
```
1 int x=1;  2
while(x<=100) {
   cout << "hello world\n"; 3
    x++; 4
}
```

Common Mistakes

```
int x=1;
while(x<=100) {
   cout << “hello world\n”;
   ~~x++;~~
}
```

What if we forget this?

What if we forget this?



Ctrl + C

The do/while loop

— 1 or more

```
int x=1;
do {
   cout << “hello world\n”;
   x++;
} while(x<=100);
```

play game

—then ask if again

- Difference b/w while and do/while?

Nested for Loops

Oregon State University
College of Engineering

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

- How many times is Hello World printed?

10 times the outer loop is exec.

*

times the inner loop exe.

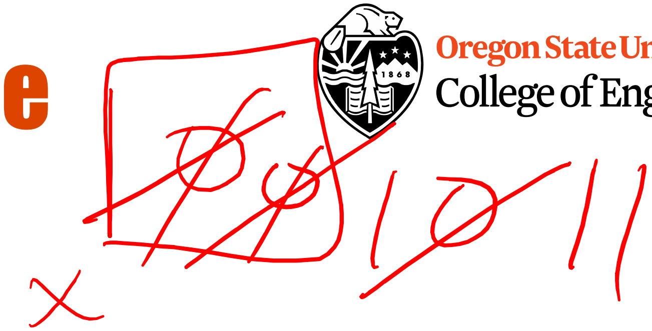
10

100

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?

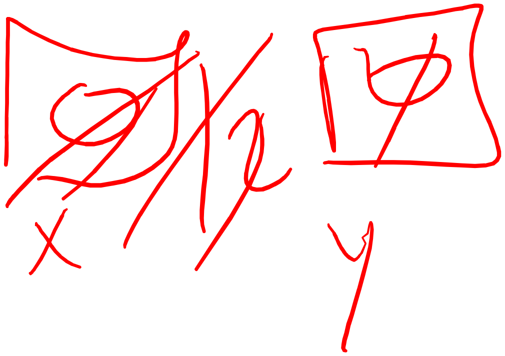
Infinite Loops

~~0~~ ~~0~~ ~~5~~ ~~6~~ 0

x

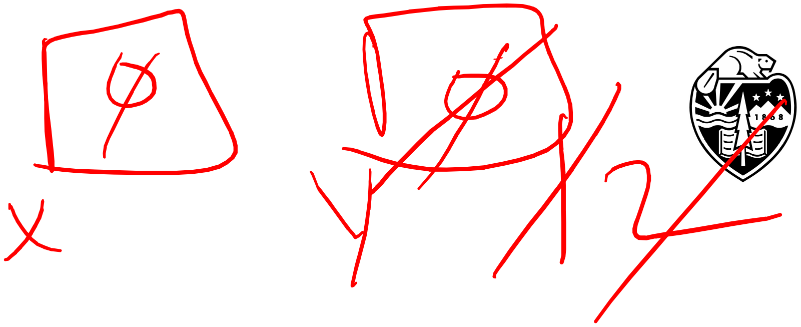
```
int x;
for(x = 0; x < 10; x++) {
  for(x = 0; x < 5; x++) {
    cout << "The value of x is: " << x << endl;
  }
}
```

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

~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~

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

x-1

int x; declarative statement

Why is this good/bad?

Oregon State University
College of Engineering

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

y does not exist here

x does not exist here

old C, can't have declarative statements after non-declarative statement

- 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?

More about break, exit, and return

Oregon State University
College of Engineering

- **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!

Oregon State University
College of Engineering

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
```