# CS 161
# Intro to CS I

More Conditional Statements

and Begin Repetition

# Odds and Ends...

- Make sure you have at least one depth of 5 on the decision tree (there is a path where the user enters 5 inputs/choices before the program ends!!!).

- Assignment #2 due Sunday, 11:59pm
  - .cpp file (source code) and .pdf file (design)

- Assignment #2 questions?

**OSU** **Oregon State University**

# C++ Switch Example

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

```cpp
 1 #include <iostream>
 2
 3 using namespace std;
 4
 5 int main() {
 6    int x;
 7
 8    cout << "enter 0, 1, or 2: ";
 9    cin >> x;
10
11    //Without a break, then after the match to the case occurs, then all
12    //other lines are executed until a break is encountered or the end of
13    //the switch is encountered.
14    switch(x) {
15       case 0:
16       case 1:
17          cout << "X is zero or one" << endl;
18       case 2:
19          cout << "X is two" << endl;
20       default:
21          cout << "You entered something invalid" << endl;
22
23    }
24
25    return 0;
26 }
~
~
~
~
```

```
-- INSERT --                                    24,1        All
-- INSERT --                                    17,74       All
```

4

# Fun Friday…

- 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";
}
```

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

&lt;expression1&gt; ? &lt;expression2&gt; : &lt;expression3&gt;

# Demo…

# Multiple Decisions

- What if I want to make these same decisions for the whole year?

> If it is sunny today
>> then I'll go to the beach
>> if it is windy at the beach
>>> then I'll fly a kite
>> else if it is not windy at the beach
>>> then I'll walk on the shore
>
> Else if it is raining today
>> then I'll stay inside and read a book
>
> Else if it is snowing
>> then I'll go to the mountains to ski

- Repeat the process for 365 days

# How do we do this?

- Repetition: for loops
  - Semantics
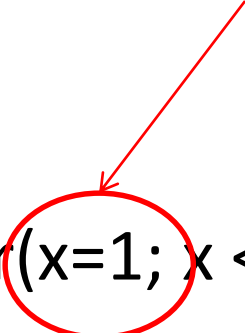    - Repeat for a specific # of iterations w/ starting point, ending point, and an increment
  - Syntax

  ```
  for(x=1; x <= 365; x++) {
      <statement>;
      <statement>;
      …
  }
  ```

# The for Loop

Starting point:
Initialization

```
for(x=1; x <= 365; x++) {
    <statement>;
    <statement>;
    …
}
```

# The for Loop

Ending point:
Continuation Test

```
for(x=1; x <= 365; x++) {
    <statement>;
    <statement>;
    …
}
```
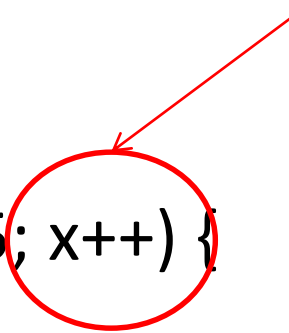
OSU Oregon State University

# The for Loop

Test is True:
Execution Block

```
for(x=1; x <= 365; x++) {
    <statement>;
    <statement>;
    ...
}
```

- What do you notice about order?

# The for Loop

Increment:
Update

```
for(x=1; x <= 365; x++) {
    <statement>;
    <statement>;
    …
}
```

- Same as x = x+1
- What about x = x + 2?
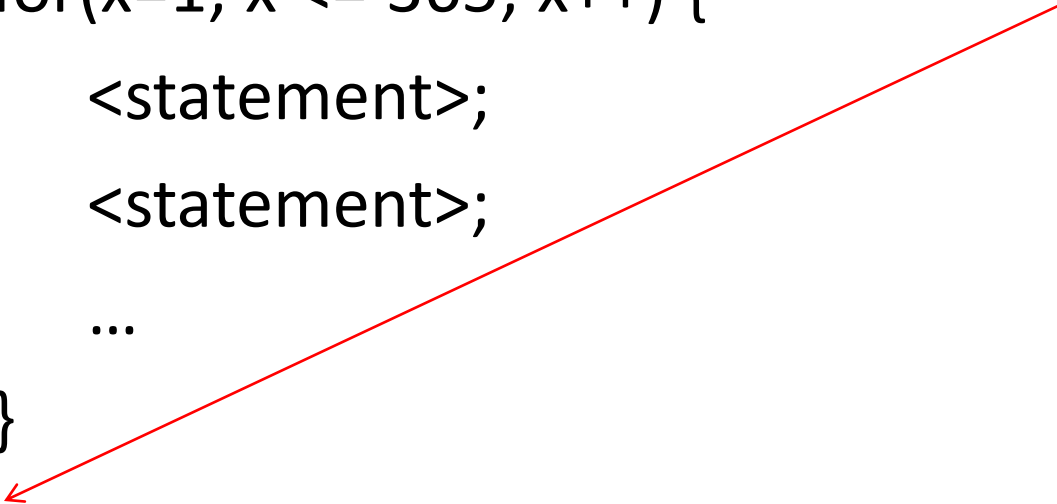
# The for Loop

Ending point:
Continuation Test

for(x=1; x <= 365; x++) {

    <statement>;

    <statement>;

    …

}

- What do you notice about order?

Oregon State University

# The for Loop

```
for(x=1; x <= 365; x++) {

    <statement>;

    <statement>;

    …

}
```

Test is False:
Execution
after loop

14

# The for Loop Examples

```
for(x=-100; x <= 100; x++)
    cout << "hello world\n";
for(x=2+2; x <= 17*3; x++)
    cout << "hello world\n";
for(x=0; x <= 100; x++)
    cout << "hello world\n";
for(x=0; x < 100; x++)
    cout << "hello world\n";
for(x=-100; x <= -1; x++)
    cout << "hello world\n";
```

OSU Oregon State University

# The for Loop Examples

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

- Why is it better to use curly braces?

**OSU** Oregon State University

# The for Loop Pattern

for(<variable> = n; <variable> **<=** p; <variable>**++**) {

    <statement>;

    …

}

for(<variable> = n; <variable> **>=** p; <variable>**--**) {

    <statement>;

    …

}

# The while loop

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

**VS.**

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

# Common Mistakes

int x=1; ← What if we forget this?

while(x<=100) {

  cout << "hello world\n";

  x++; ← What if we forget this?

}

## Why use a while instead of a for?

# The do/while loop

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

- Difference b/w while and do/while?