CS 161
Intro to CS I
More Decisions/Begin Loops
Odds and Ends...

• Recitation Quiz #2 emailed tonight, 11:59pm
• Assignment 3 posted.

• How many interested in a study social for those feeling lost, behind, not fitting in, etc.?
Logical Operators

• **AND:** if((1>2) && (2<5))
• **OR:** if((1>2) || (2<5))
• **NOT:** if(!(1>2) && (2<5))

• Precedence of Operators: refer to book
C++ If/Else Syntax...

```cpp
if( x > y) {
    cout << “X is greater than Y” << endl;
}
else {
    cout << “X is less than Y” << endl;
}
```

• When does this logic fail?
C++ If/Else...

if( x > y) {
    cout << “X is greater than Y” << endl;
}
else if( x < y) {
    cout << “X is less than Y” << endl;
}
else {
    cout << “X is equal to Y” << endl;
}
What are the curly braces for?

```cpp
if( x > y )
    cout << "X is greater than Y" << endl;
else if( x < y )
    cout << "X is less than Y" << endl;
else
    cout << "X is equal to Y" << endl;
```
What if we are testing for ==?

```cpp
if( x == 0) {
    cout << "X is zero" << endl;
}
else if( x == 1) {
    cout << "X is one" << endl;
}
else if( x == 2) {
    cout << "X is two" << endl;
}
else {
    cout << "Not 0, 1, 2!!!" << endl;
}
```
We can use a switch...

```java
switch( <expression> ) {
    case <const-expr>:
        <statement>;
        ...
    case <const-expr>:
        <statement>;
        ...
    default:
        <statement>;
        ...
}
```

C++ Switch Example

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

• 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?

```cpp
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++?

<expression1> ? <expression2> : <expression3>
Demo...
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...
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

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

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>;
  ...
}

18

OSU Oregon State University
The for Loop

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

• What do you notice about order?
The for Loop

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

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

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

- What do you notice about order?
The for Loop

```java
for(x=1; x <= 365; x++) {
    <statement>;
    <statement>;
    ...
}
Test is False: Execution after loop
```
Demo...
The while loop

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

VS.

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

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

What if we forget this?

What if we forget this?
The do/while loop

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

• Difference b/w while and do/while?