EECS 161
Intro to Programming I

Conditional Statements
Chap. 2.1
Decisions in Life

- What is a decision?
- When do we make decisions?
- How do we make decisions?

If it is sunny today, then I’ll go to the beach and fly a kite.
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.
Decisions within Decisions

• What happens if there is no wind at the beach?

• How does this change our decisions?
  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
  if it is not windy at the beach
  then I’ll walk on the shore
Flow chart for decisions

- Is it sunny?
  - Yes: Go to beach
  - No: Is it raining?
    - Yes: Read book
    - No: Go outside

- Is it windy?
  - Yes: Fly kite
  - No: Walk on beach
Decisions in our programs

- Use an if/else

```java
if (<expression>) {
    <statement>;
    ...
    <statement>;
} else {
    <statement>;
    ...
}
```
What is the <expression>?

Could be a relational expression:

<expression> <relational op> <expression>

- Relational Ops
  - == - equal to
  - != - not equal to
  - < - less than
  - > - greater than
  - <= - less than or equal to
  - >= - greater than or equal to
Examples

- if(2 + 1) //expression
- if(2 - 4) //expression
- if(2 - 2) //expression
- if(4 == 4) //expression relational op expression
- if((2+1) == 4) //expression relational op expression
- if(4.1 != 4) //expression relational op expression
- if(3 <= 4) //...
- if(4 >= 4)
- if(3.5 > 4)
- if(4 < 4)
- if(3+2*2 > 9)
- if((3+2)*2 > 9)
C++ If/Else Syntax...

```cpp
int x, y;
x = 12;
y = 10;
if (x > y) {
    cout << "X is greater than Y" << endl;
}
else {
    cout << "X is less than Y" << endl;
}
```

• When does this logic fail?
```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 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;
```

(always assoc 1st stmt under w/ decision)
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 << “You have entered an invalid number!!!” << endl;
}
```
Logical Operators

• **AND**: if((1>2) && (2<5))

• **OR**: if((1>2) || (2<5)) — if either is true then true

• **NOT**: if(!(1>2) && (2<5))

• **Precedence of Operators**: pg. 51 - 52

\[ 7 < x < 10 \]
We can use a switch...

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

The switch statement tests for equality, or a range of equality.
C++ Switch Example

```cpp
switch( x ) {
    case 0:
        cout << "X is zero" << endl;
        break;
    case 1:
        cout << "X is one" << endl;
        break;
    case 2:
        cout << "X is two" << endl;
        break;
    default:
        cout << "You have entered an invalid number!!!" << endl;
}
```
C++ Switch Example

```cpp
switch( x ) {
    case 0:
        std::cout << "X is zero or one\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";
}
```
More conditionals...

- What if we didn’t have a break in the case of a switch statement?
- What do you think this does in C/C++?

<expression1> ? <expression2> : <expression3>