CS 161
Intro to CS I

Conditional Statements
Odds and Ends...

- Assignment #2 due Sunday, 11:59pm
- Recitation Quiz #2 emailed Monday, 11:59pm
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
  
  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)
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...

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?

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 ==?

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;
}
Demo...
We can use a switch...

```plaintext
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
”;
            break;
    case 2:
        std::cout << “X is two
”;
        break;
    default:
        std::cout << “You have entered an invalid number!!!
”;
}
```
Demo...
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++?
  ```cpp
  <expression1> ? <expression2> : <expression3>
  ```