

# CS 161

## Intro to CS I

More Conditional Statements

# Recap C++ If/Else...

```
if( x > y) {  
    std::cout << "X is greater than Y\n";  
}  
else if( x < y) {  
    std::cout << "X is less than Y\n";  
}  
else {  
    std::cout << "X is equal to Y\n";  
}
```

# What are the curly braces for?

```
if( x > y)
    std::cout << "X is greater than Y\n";
else if( x < y)
    std::cout << "X is less than Y\n";
else
    std::cout << "X is equal to Y\n";
```

# What if we are testing for ==?

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

# Demo...

# We can use a switch...

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

# C++ Switch Example

```
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

```
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";  
}
```



# 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...