CS 161
Intro to CS I
More Conditionals and Begin Repition
```cpp
#include <iostream>

using namespace std;

int main( ) {
    int num;

    //variable stuff
    cout << "enter num 0-15: "; //give prompt for reading input
    cin >> num;   //what if it breaks at this point
    //might want to read in as string and check the characters...

    if(num>=0 && num<=15) {
        cout << num/8 << endl; //int arithmetic, 0 or 1 if there aren't/are 8s
        num=num-(num/8)*8; //now don't take or take those 8s out of num
    }

    return 0;
}
```
Demo...

```bash
flip2 ~/cs161/private 113% g++ hello.cpp
flip2 ~/cs161/private 114% a.out
enter num 0-15: t
0
flip2 ~/cs161/private 115% a.out
enter num 0-15: hello
0
flip2 ~/cs161/private 116% a.out
enter num 0-15: -15
flip2 ~/cs161/private 117% a.out
enter num 0-15: 16
flip2 ~/cs161/private 118% a.out
enter num 0-15: 10
1
flip2 ~/cs161/private 119%
```
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
Our Assignment #2

• How are we going to use strings to help us?
Demo...
Switch Statements...

```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” << endl;
        break;
    case 1:
        std::cout <<“X is one ” << endl;
        break;
    case 2:
        std::cout <<“X is two ” << endl;
        break;
    default:
        std::cout <<“You have entered an invalid number!!! ” << endl;
}
```
C++ Switch Example

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

• What if you have the statement `var = var + 1;`
  `var += 1; /*Add operand on right to var*/`
  `var++; /*Increment var by one*/`

• What if you have the statement `var = var - 5;`
  OR `var = var + var;`
  `var -= 5;`
  `var += var;`

• Pre vs. Post increment: `++var` vs. `var++`
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
The for Loop

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

Ending point: Continuation Test
The for Loop

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

• What do you notice about order?

Test is True: Execution Block
The for Loop

```java
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

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

Test is False: Execution after loop
The while loop

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

VS.

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

What if we forget this?

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

What if we forget this?
The do/while loop

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

• Difference b/w while and do/while?
 Looping ...

• for loops
  – Repeat for specific number of times
  – Example?

• while loops
  – Repeat while a condition is being met
  – Example?

• do while loops
  – Always do once, and repeat while condition is met
  – Example?