CS 161
Intro to CS I
Assignment #1 & Decisions
Chap. 2.1-2.2
#include <iostream>
#include <climits>
#include <cmath>

using std::cout;
using std::endl;
using std::pow;

int main(void) {
    unsigned short us_num;
    us_num = USHRT_MAX;
    // us_num = us_num + 1;
    // us_num += 1;
    cout << ++us_num << endl;

    long test = (long) pow(2, 62) - 1;
    long test2 = pow(2, 62);
    cout << test << endl << test2 << endl;

    cout << ((int) pow(2, (sizeof(int) * 8 - 1) - 1) * 2 - 1) << endl;
    cout << ((int) pow(2, (sizeof(int) * 8 - 1) - 1) * 2 + 1) << endl;

    cout << (int) pow(2, (sizeof(int) * 8 - 1)) - 1 << endl;
    cout << (unsigned int) pow(2, (sizeof(int) * 8 - 1)) - 1 << endl;

    cout << (int) (pow(2, (sizeof(int) * 8 - 1)) - 1) << endl;
    return 0;
}
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)
• if(2 – 4)
• if(2 – 2)
• if(4 == 4)
• if((2+1) == 4)
• if(4.1 != 4)
• 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
if( x > y) {
    std::cout << "X is greater than Y\n";
}

else {
    std::cout << "X is less than Y\n";
}

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

```cpp
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?

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

```cpp
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";
}
```
Logical Operators

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

- **Precedence of Operators**: pg. 51 - 52
We can use a switch...

```java
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

```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";
}
```
Reading/Assignments

• Read Chap. 2.3
• No class on Monday, 1/21
Quiz #2

• 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>`
```cpp
#include <iostream>

int main()
{
    int x = 0;

    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";
        break;
    }

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

    (x > 0) ? std::cout << "hello\n" : std::cout << "hello2\n";
    return 0;
}
```