Operators

if/else statements revisited

```
if (conditional1) {
    // Execute everything in here if
    // conditional1 was evaluated to true
}
else if (conditional2) {
    // Execute everything in here if
    // conditional1 was evaluated to false AND
    // conditional2 was evaluated to true
}
else {
    // Execute everything in here if
    // conditional1 was evaluated to false AND
    // conditional2 was evaluated to false
}
```

Does the following compile?

```
#include <iostream>
using namespace std;

int main() {
    int x = 30;

    if (x == 10)
        cout << "Test 1! ";
        cout << "x is 10" << endl;
    else if (x == 20)
        cout << "Test 1! ";
        cout << "x is 10" << endl;
    else
        cout << "Test 1! ";
        cout << "x is 10" << endl;
}
```

T or F?

The if statement evaluates a **conditional** statement between the parenthese

- Uses **operators** like < or >
  - **Operators** are symbolic keywords used to compute values between operands, or the arguments to the operators
- Evaluates to true or false

The number 0 equates to false

```
if (0)
    cout << "Never gets printed :(" << endl;
else
    cout << "Always printed :)" << endl;
```

Conditional operators

Operator	Meaning
==	Equal to
!=	Not Equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to

Checking multiple conditions

What if I wanted to check multiple conditions in an if statement?

- X Is between 0 **and** 100
- X is less than 0 **or** greater than hundred

There are special operators to do that

&&	AND
||	OR

```
// Check between 0 (non inclusive) and 100
if (x > 0 && x <= 100)
cout << "Just right" << endl;

// Check if less than or equal to 0 and greater than 100
if (x <= 0 || x > 100)
cout << "So many numbers!!" << endl;
```

Mathematical operators

Operator	Meaning
+	Plus
-	Minus
*	Multiply
/	Divide
%	Modulo (gets remainder)
++	Increment
--	Decrement

Operators have Precedence

What does that even mean??

What if I gave you the following set of instructions: "Wash the car, dry the car, and do your homework."

In what order will you perform the tasks?

Chances are you'll do one of the following:

- Wash the car, then dry the car, and then do your homework.
- Do your homework, then wash the car, and then dry the car.
- Some of you anarchists might want to wash the car, then do your homework, then dry the car because the man doesn't tell you how to clean.

Its all about the order of operations

Some operators have higher priority

- are evaluated before others

http://en.cppreference.com/w/cpp/language/operator_precedence

PEMDAS to the next level

Associativity varies!

- The direction in which operators evaluate their operands; can be left to right (e.g. 5 + 3 + 2) or right to left (e.g. i = 5;)

Manually set precedence

Use () to control the order of operations

Statement in () is evaluated before it is used

Some examples on precedence

⚠ Will the following code compile? If so, what will it print out?
Note precedence levels: (=, 15), (+, -, 6), (*, 5)

```
#include <iostream>
using namespace std;

int main () {
    int a = 5,
        b = a = a + 5 * 2; // "Parallel Assignment"

    cout << a << endl;
    cout << b << endl;
}
```

⚠ Will the following code compile? If so, what will it print out?
Note precedence levels: (=, 15), (+, -, 6), (*, 5)

```
#include <iostream>
using namespace std;

int main () {
    int x = -3,
        y = 5,
        z = 8 * (x = y - 4);

    cout << x << endl;
    cout << z << endl;
}
```

Strings

Strings objects that represent a sequence of characters

- Surrounded by “ ”

You have been using strings with cout.

To use strings as variables need to **#include <string>**

Strings can be **concatenated**

The + operator behaves differently with strings

- It combines strings into new string

Be careful!

- You cannot concatenate a string and an int

What does the program print?

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string me = "Sakib",
    blank = "",
    space = " ",
    newLine = "\n",
    result,
    nothing;

    result = me + "iscool";
    cout << result << endl;
    cout << "-----" << endl;

    result = me + blank + blank + "=reallycool";
    cout << result << endl;
    cout << "-----" << endl;
}
```

Special characters

What if I want to print Enter or Tab?

- ‘\n’ is a new line
- ‘\t’ is a tab
- These are treated as ONE character

How long is a string

Use the .length() method of a string to find out how many characters are in a string

```cpp
#include <iostream>
#include <string>
using namespace std;

int main() {
    string me = "Sakib",
    blank = "",
    space = " ",
    newLine = "\n",
    result;

    cout << me << ": " << me.length() << endl;
    cout << "blank" << ": " << blank.length() << endl;
    cout << "space" << ": " << space.length() << endl;
    // What will newLine.length() return?
    cout << "newLine" << ": " << newLine.length() << endl;
    cout << "result" << ": " << result.length() << endl;
}
```