



Chapter 5

Logic; Got Any?



Flow of Control

- Flow of Control
 - The order in which the computer executes statements in a program
- Control Structure
 - A statement used to alter the normally sequential flow of control
 - Selection (Chapter 5)
 - Iteration (Chapter 6)



Selection

- Choosing between one or more options
- Example:
 - If statements
 - Switch statements



Boolean Data

- We state an assertion and check it's validity
 - If the assertion is true, we do one thing.
 - Else we do some other thing.
- We use bool data types
 - Two values
 - true and false



Logical Expressions

- o Boolean Variables and Constants
 - `eventFound = false;`
- o Relational Operators
 - `lessThan = (i < j);`



Relational Operators

- | o Operator | o Relationship Tested |
|--------------------|--------------------------|
| <code>==</code> | Equality |
| <code>!=</code> | Inequality |
| <code>></code> | Greater Than |
| <code><</code> | Less Than |
| <code>>=</code> | Greater Than or Equal To |
| <code><=</code> | Less Than or Equal To |



Logical Operators

o Three operators

- And `&&`
 - Both relationship must be true for the and to be true
- Or `||`
 - One relationship must be true for the or to be true
- Not `!`
 - Inverts the relationships truth value



And Truth Table

Value of x	Value of y	Value of x && y
true	true	true
true	false	false
false	true	false
false	false	false



Or Truth Table

Value of x	Value of y	Value of x y
true	true	true
true	false	true
false	true	true
false	false	false



Not Truth Table

Value of x	Value of !x
true	false
false	true



Short Circuit

- If the value of a statement can be determined before the entire statement must be evaluated, than do it



Precedence of Operators

! Unary + Unary -	Highest Precedence ↓ Lowest Precedence
* / %	
+ -	
< <= > >=	
== !=	
&&	
=	



The if Statement

- If-Then-Else Form

 - if (Expression)

 - Statement A

 - else

 - Statement B



Expressions and Statements

- An expression is a statement that evaluates to a logic value

- A statement is what you want the computer to do

○ ● ● | Example

```
if ( hours <= 40.0 )
    pay = rate * hours;
else
    pay = rate * ( 40.0 + ( hours -40.0 ) * 1.5);
cout << pay;
```

○ ● ● | Compound Statements

- Not limited to one statement
- Use { and } to indicate statements that are to be grouped together



Example

```
if ( divisor != 0 )
{
    result = dividend / divisor;
    cout << "Division performed.\n";
}
else
{
    cout << "Division by zero is not allowed\n";
    result = 9999;
}
```



If-Then Form

- Sometimes there is only one choice
- Use:
if (Expression)
Statement



Nested If's

- o You can stack ifs as deep as you want

```
if ( month == 1 )
    cout << "January";
else if ( month == 2 )
    cout << "February";
...
else
    cout << "December";
```



Dangling Else

- o Where does the else go?

```
if ( average < 70.0 )
    if ( average < 60.0 )
        cout << "FAILING";
    else
        cout << "Barely Passing ☹";
```

○ ● ● | How about this one

```
if ( average >= 60.0 )
    if ( average < 70.0 )
        cout << "Barely Passing ☹️";
else
    cout << "FAILING";
```

○ ● ● | Correct Version

```
if ( average >= 60.0 )
{
    if ( average < 70.0 )
        cout << "Barely Passing ☹️";
}
else
    cout << "FAILING";
```

○ ● ● | Testing streams

- You can test whether the last operation performed on a stream failed or not
 - `if (cin)`
 - `if (cin.fail())`
 - `if (!inFile)`
 - `if (!inFile.fail())`