

(Re)Introduction to C++

Dr. Robert Lowe

Division of Mathematics and Computer Science
Maryville College

Outline

- 1 C++ Basics
- 2 Operations and Decisions

Structure of a C++ Program

```
// File: boilerplate.cpp
// Purpose: Sample C++ Program
// Author: Your Name Here
#include <iostream>

using namespace std;

int main()
{

}
```

Compiling C++ Programs

- `g++ program.cpp -o program`
- `make program`

Variables

- Variables are named areas of memory.
- In C++, variables must be declared before they can be used.
- A variable declaration has the form:
`type identifier`
- Identifiers consist of letters, numbers, and underscores.
- Identifiers must begin with a letter or underscore.
- Words in variable names are usually separated by underscores, or via camelCasing.

Data Types

Primitive Data Types

`char` - Character, single letter/symbol

`bool` - Boolean true or false value

`int` - Integer (whole number)

`float` - Single precision floating point number (never use!)

Common Complex Types

`string` - A string of characters

`ifstream` - An input file stream (for reading)

`ofstream` - An output file stream (for writing)

`vector<type>` - A list of variables of `type`.

Input and Output

- Input is accomplished via the extraction operator:
`cin » x;`
- Output is accomplished via the insertion operator:
`cout « "Hello, World" « endl;`
`cout « x « endl;`

Programming Project P1.2

From the end of chapter 1 in *Big C++*:

Programming Problem P1.2

Write a program that prints out a message “Hello, my name is Hal!” Then, on a new line, the program should print the message “What is your name?” As in Exercise P1.1, just use the following lines of code:

```
string user_name;  
getline(cin, user_name);
```

Finally, the program should print the message “Hello, user name. I am glad to meet you!”

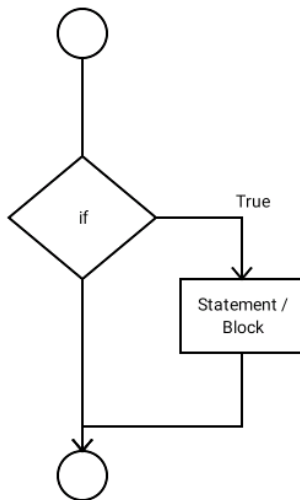
Operator Precedence (thus far)

Operator	Description	Associativity
a++, a--	Postfix increment and decrement	Left-to-Right
not, !	Logical Not	Right-to-Left
++a, --a	Prefix increment and decrement	
a*b, a/b, a%b	Multiply, Divide, Modulus	Left-to-Right
a+b, a-b	Addition and Subtraction	Left-to-Right
«, »	Insertion and Extraction	Left-to-Right
<, <= >, >=	Relational Operators	Left-to-Right
==, !=	Equality Operators	Left-to-Right
and, &&	Logical And	Left-to-Right
or,	Logical Or	Left-to-Right
=, +=, -= *=/= %=	Assignment and Assignment	Right-to-Left

If Statement

```
if ( condition )  
    statement/block
```

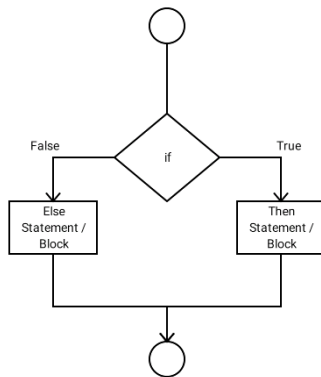
- If the *condition* is true, the *statement/block* will be executed.
- If the *condition* is false, the *statement/block* will be skipped.



If Else Statement

```
if ( condition )  
    then statement/block  
else  
    else statement/block
```

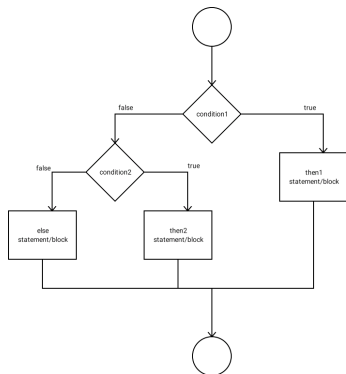
- If the *condition* is true, the *then statement/block* will be executed.
- If the *condition* is false, the *else statement/block* will be executed.



Multi-Way Branching: If-Then-Else-If

```
if ( condition )  
    then statement/block  
else if ( condition )  
    then statement/block  
else  
    else statement/block
```

- The first *then statement/block* with a true condition executes.
- If no matches are found, the (optional) *else statement/block* executes.



Programming Project P3.3

From the end of Chapter 3 in *Big C++*:

Programming Problem P3.3

Write a program that takes user input describing a playing card in the following shorthand notation:

A Ace

2 ... 10 Card values

J Jack

Q Queen

K King

D Diamonds

H Hearts

S Spades

C Clubs

Your program should print the full description of the card.