



CS 31 Introduction to Computer Science I



Howard A. Stahl



Agenda

- Welcome
- Review Syllabus
- My Philosophy
- Let's Get Started...

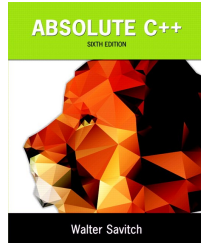


Let's Get Started...

- The Textbook
- Computer Languages
- The History of C++
- The Compilation Process
- Developing Programs With C++
- HelloWorld.cpp
- Variables and Datatypes

The Textbook

- Absolute C++ By Walter Savitch
 - Any Edition Will Do...
 - Readable And Useful



Computer Languages

- Computer Languages Have Evolved Over Time
- Initially, Programmers Wrote Code In Machine Language
01010110 0001 0001000
- Eventually, Assemblers Were Made To Hide Machine Language Behind Mnemonic Instructions
ADD R1, 8

High-Level Languages

- C++ Is A "High-Level" Language
- With High-Level Languages, Programmers Write Programs In A Structure Quite Different From What The Machine Actually Executes
- Languages Are Interpreted Or Compiled
 - C++ Is A Compiled Language

The Original Creator Of C++

- Bjarne Stroustrup
 - Formerly of AT&T, Now With Morgan Stanley

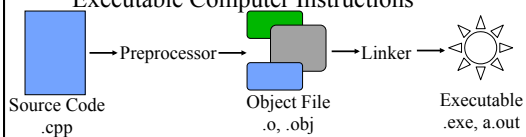


The History of C++

- Authored by Bjarne Stroustrup, AT&T
- Extended the C Language
 - Supports Object-Oriented Programming
- C++ is considered a superset of C
- Language Is Now An International Standard
 - C++03 : ISO Standard from 2003 (VS 2012)
 - C++11 : ISO Standard from 2011 (Xcode 6/7, VS 2015)
 - C++14 : ISO Standard from 2014

Compiled Languages

- Compiled Languages Must Be Turned Into Executable Computer Instructions



- Errors Can Occur At Each Step!
 - compile-time, linkage, run-time

Time For Our First Demo!

- HelloWorld.cpp

(See Handout For Example 1)

Summarizing Our First Demo!

- A Preprocessor handles lines with #
- All C++ statements end with a ;
- The main() function starts the program
- Opening and Closing Braces define code blocks { }
- cout & cin come from <iostream>
- Comments can be single-line with // or extend over multiple lines with /* */

Input and Output in C++

- C++ input statement: `cin >> number;`
 - a numerical value is extracted from the keyboard (cin) and is placed into the variable called "number".
- C++ output statement: `cout << "Hello";`
 - send information from program to terminal screen (cout)
 - double quotes "..." delimit a string
 - `\n` sends a new-line-character

Variables and Datatypes

- Most Programs Manipulate Variables
- Variables Are Named Memory Locations
- Variables Must Be Declared

Datatype	Description
int, short, long	Whole numbers
double	Decimal numbers
string	Characters
bool	true or false
char	A Single Character

Data Types:

Display 1.2 Simple Types (1 of 2)

Display 1.2 Simple Types

TYPE NAME	MEMORY USED	SIZE RANGE	PRECISION
short (also called short int)	2 bytes	−32,768 to 32,767	Not applicable
int	4 bytes	−2,147,483,648 to 2,147,483,647	Not applicable
long (also called long int)	4 bytes	−2,147,483,648 to 2,147,483,647	Not applicable
float	4 bytes	approximately 10^{-38} to 10^{38}	7 digits
double	8 bytes	approximately 10^{-308} to 10^{308}	15 digits

Copyright © 2016 Pearson Inc.

Data Types:

Display 1.2 Simple Types (2 of 2)

long double	10 bytes	approximately 10^{-4932} to 10^{4932}	19 digits
char	1 byte	All ASCII characters (Can also be used as an integer type, although we do not recommend doing so.)	Not applicable
bool	1 byte	true, false	Not applicable

The values listed here are only sample values to give you a general idea of how the types differ. The values for any of these entries may be different on your system. *Precision* refers to the number of meaningful digits, including digits in front of the decimal point. The ranges for the types **float**, **double**, and **long double** are the ranges for positive numbers. Negative numbers have a similar range, but with a negative sign in front of each number.

Copyright © 2016 Pearson Inc.

C++11 Fixed Width Integer

TYPE NAME	MEMORY USED	SIZE RANGE
int8_t	1 byte	−128 to 127
uint8_t	1 byte	0 to 255
int16_t	2 bytes	−32,768 to 32,767
uint16_t	2 bytes	0 to 65,535
int32_t	4 bytes	−2,147,483,648 to 2,147,483,647
uint32_t	4 bytes	0 to 4,294,967,295
int64_t	8 bytes	−9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
uint64_t	8 bytes	0 to 18,446,744,073,709,551,615
long long	At least 8 bytes	

Avoids problem of variable integer sizes for different CPU architectures

Copyright © 2016 Pearson Inc.

Variables and Datatypes

- Variables Are Known By Name
- Identifiers Must
 - begin with a-z, A-Z, or _
 - followed by a-z, A-Z, 0-9 or _
- Identifier Names Are Case-Sensitive
- It is always good practice to initialize variable values when they are declared

Reserved Keywords

- You can't use these lowercase names
- Full List in Appendix 1, page 9115
 - break case char
 - const default do
 - double else extern
 - float for if
 - int long return
- We'll be learning about these over time...

Literals

- A fixed, static value used in a program
- Called “literals” because you “literally typed” them into your program!
- Three basic types in C++
 - numeric 5.1E+3 3.14159 -70
 - character ‘a’ ‘7’ ‘*’
 - string “Hello World!”

Literals and Variables Compared

- A literal is a fixed value that never changes
- A variable is a container for values
 - a named value that may change
 - assignment statement is one way
 - there are many others
 - A variable can only hold one value at a time
 - A variable loses its old value when a new one
 - All variables must be declared before used

Variable Declarations

- Every variable in C++ must be declared
 - normally occurs at the start of a main program
 - associates a name with a datatype
- Syntax: type_name variable_name;
- Examples: int i; double d;
- Your book focuses solely on int and double

Assignment Statement

- Common form: `variable = expression;`
 - Causes expression to be evaluated and the result assigned as the new value of the variable
- Examples: `i=5; i=i+1; y=m*x+b;`
- Can't Do's: `1=7; 5=x; "U"="2";`

Time For Our Next Demo!

- Grader.cpp

(See Handout For Example 2)

Summarizing Our Second Demo!

- Programs Can Get Quite Long!!!
- Use `#include <string>` for string
- Variables hold a single value at a time
- Const holds only a single value ever
- Arithmetic Operations: `+` `-` `*`
- Notation Shorthand: `++` `--` `+=` `-=` `*=` `/=`
- Special Characters: `\t` `\n` `\"` `\'` `\\` `\\?` `\\`

Variable Initialization

- A variable has no meaningful value unless assigned
- Rule: Set each variable before its value is used! GARBAGE, otherwise! (The container has no predictable value)
- One way of avoiding uninitialized variables: initialize at the time of declaration

```
- int your_sum = 20;  
- double rate(0.1), balance(0.00);
```

Time For Our Next Demo!

- Datatypes.cpp

(See Handout For Example 3)

Summarizing Our Third Demo!

- Variables Are Typed Memory Locations
 - datatype determines size requirements
- When Choosing Datatypes, Be Mindful Of Their Valid Values

Summary

- Computer Languages
- The Compilation Process
- The History of C++
- Developing Programs With C++
- HelloWorld.cpp
- Variables and Datatypes
