

CSI33 DATA STRUCTURES

Department of Mathematics and Computer Science
Bronx Community College

October 23, 2017



OUTLINE

1 CHAPTER 8: A C++ INTRODUCTION FOR PYTHON PROGRAMMERS

- Expressions and Operator Precedence
- Decision Statements
- Type Conversions
- Looping Statements
- Arrays



OUTLINE

1 CHAPTER 8: A C++ INTRODUCTION FOR PYTHON PROGRAMMERS

- Expressions and Operator Precedence
- Decision Statements
- Type Conversions
- Looping Statements
- Arrays



LOCAL VARIABLES

- Must be declared giving their datatypes, so values are guaranteed to fit in the memory locations reserved by the compiler.
- May optionally be initialized when declared.
- Local variables declared in a function (not formal parameters) are called **automatic** variables. They are given memory locations without putting values in those locations. So automatic variables must be initialized before use (to get predictable value).
- The memory for a variable holds an actual value, not a reference (as in Python).
- Exception: References (**pointer** variables) are specially denoted by *, the **dereference** operation.



EXPRESSIONS

EXPRESSIONS

- Just as in any programming language, expressions have values.
- The simplest expressions are constants or variable names.
- Just as in any programming language, expressions are constructed by connecting smaller sub-expressions with operator symbols. These values are calculated by performing the operations, with the highest precedence operation first.
- Expressions can also be formed by making function calls which return values.



EXPRESSIONS

OPERATORS ARE SIMILAR TO THOSE IN PYTHON

Exceptions:

- **&&** for and.
- **||** for or.
- **increment** operators: $x++$ or $++x$ for $x += 1$
- **decrement** operators: $x--$ or $--x$ for $x -= 1$

EXPRESSIONS

OPERATOR PRECEDENCE

The order of operations in C++ follows the same standard rules as Python:

$(a+b*c)$ means multiply b times c , then add a .



IF STATEMENT

INDENTATION IS MEANINGLESS

Since C++ ignores indentation, multiple statements in an `if` statement must be enclosed in braces to form a block.



IF STATEMENT

C++ HAS NO ELIF

Multiple conditional branches in C++ require nested if statements.



DATA TYPES

IMPLICIT CONVERSION

Like Python, C++ automatically converts ints to floats or doubles when adding an integer to a float value.



DATA TYPES

EXPLICIT CONVERSION

C++ also performs explicit conversion with the `static_cast` keyword.

LOOPS

FOR LOOPS OVER DIFFERENT VALUES OF A SINGLE VARIABLE

In Python, a for loop must use a pre-existing sequence of values, like the list returned by the function `range`. In C++, a variable changes its value by incrementing or decrementing its value for each iteration of the loop.



LOOPS

WHILE LOOPS

While loops in C++ are exactly like those in Python. They test for a boolean value to be true before they begin each iteration of the loop. When it is false, the loop is exited.



LOOPS

DO-WHILE LOOPS

Do-while loops in C++ are different than while loops in Python. They only test for the boolean value to be true at the end of each loop iteration. This means that a loop is guaranteed to execute at least once.



SINGLE-DIMENSION ARRAYS

NOT AS SAFE AS PYTHON LISTS

- No range checking is performed when an index is used.
- Can initialize values when declaring an array.



MULTI-DIMENSIONAL ARRAYS

ARRAYS OF ARRAYS

- Any number of dimensions is supported.
- Consecutive addresses are easily found (fast random access via a formula).



ARRAYS OF CHARACTERS

USING ARRAYS OF CHARACTERS IS RISKY

```
char c[20];  
cout << "enter your first name: ";  
// this code is a security risk  
// a buffer overflow occurs if the user enters  
// more than 19 characters  
cin >> c;  
cout << "Hello " << c << endl;
```



ARRAYS OF CHARACTERS

C STRINGS

- char arrays with terminating zero in last position.
- Literal string values (in quotes) are zero-terminated strings in C++.
- Convenient library functions for concatenation (`strcat`) etc.
- `#include <string.h>` for library functions



ARRAYS OF CHARACTERS

BETTER TO USE THE C++ STRING CLASS.

- `#include <string>`
- An object's capacity increases automatically.
- Convenient member functions for concatenation and other operations.
- Easy to initialize a C++ string from a char array.
- Rule-of-thumb: For text input/output (`cin`, `cout`), always use the string class!

