



TERMINOLOGY

Building A Common Vocabulary



STATEMENT

- One complete instruction to the computer to do something
- Like a sentence in a natural language
- Terminated by a semicolon
- Examples
 - `x = a + b;`
 - `y = sqrt(2);`
 - `cout << x << endl;`



EXPRESSION

- A fragment of code that represents or evaluates to a value
- Expression values are temporary, lasting only while the statement runs
- Expressions are part of a statement: can't stand alone and don't end with a semicolon
- Simple expressions
 - Constant: `5`
 - Variable: `counter`
 - Function call: `sqrt(5)`



COMPLEX EXPRESSIONS

- Expressions are formed recursively:
 - E is an expression, so
 - Operation(E) is an expression
- Complex expressions are formed by combining simple expressions
 - `-n`
 - `counter + 5`
 - `angle < 180`
 - `sqrt(pow(a, 2) + pow(b, 2))`



DECLARATION

- Introduces a name or identifier (i.e., a symbol) to the compiler
- Compiler puts the name in the symbol table
- Symbol table is used to generate code



DEFINITION

- Allocates memory
 - To store or hold a variable
 - To hold the machine instructions for a function
- In C++, variables and functions must be defined once before they can be used, but may be declared multiple times
- Sometimes code can be both a declaration and a definition
- Variables may be initialized when defined but it is not required
 - `int counter;`
 - `int counter = 100;`



MULTI-FILE PROGRAMS

- Variables may be defined in one file and used in another
- Must declare the variable in the using file
 - `int counter; // definition`
 - `extern int counter; // declaration`