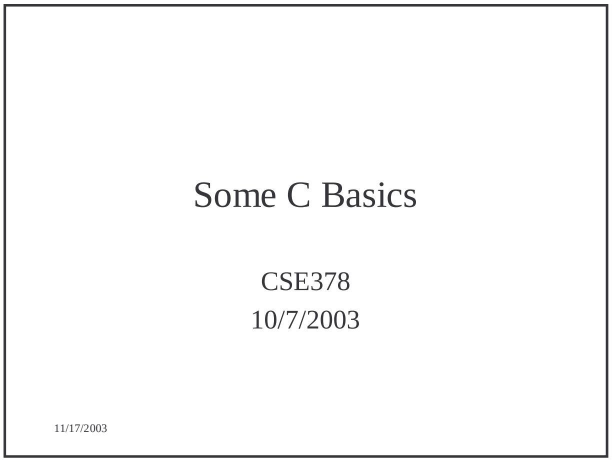
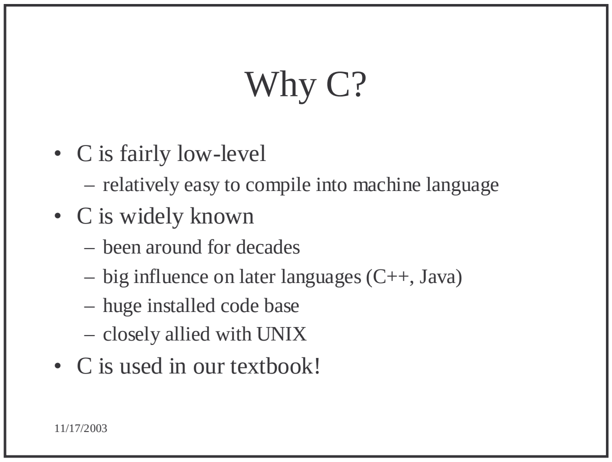


Some C Basics

CSE378
10/7/2003

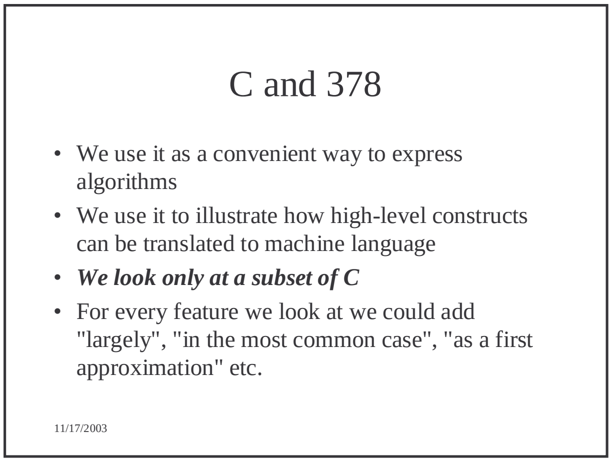
11/17/2003



Why C?

- C is fairly low-level
  - relatively easy to compile into machine language
- C is widely known
  - been around for decades
  - big influence on later languages (C++, Java)
  - huge installed code base
  - closely allied with UNIX
- C is used in our textbook!

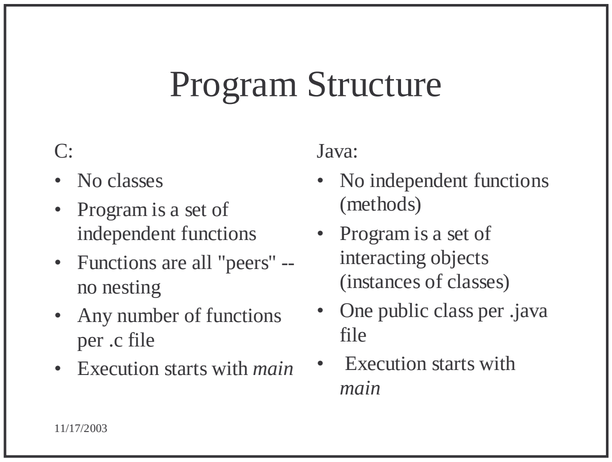
11/17/2003



C and 378

- We use it as a convenient way to express algorithms
- We use it to illustrate how high-level constructs can be translated to machine language
- ***We look only at a subset of C***
- For every feature we look at we could add "largely", "in the most common case", "as a first approximation" etc.

11/17/2003



Program Structure

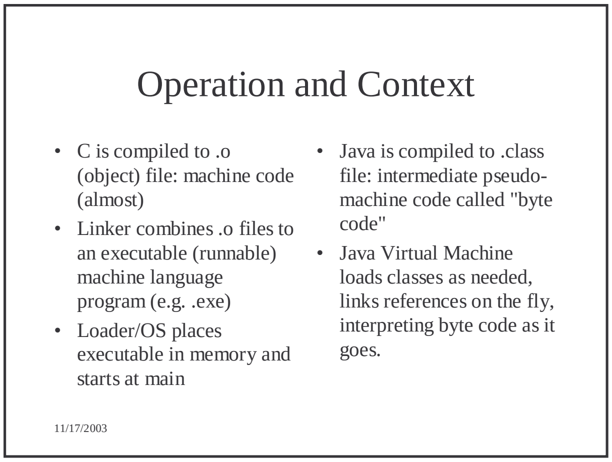
C:

- No classes
- Program is a set of independent functions
- Functions are all "peers" -- no nesting
- Any number of functions per .c file
- Execution starts with *main*

Java:

- No independent functions (methods)
- Program is a set of interacting objects (instances of classes)
- One public class per .java file
- Execution starts with *main*

11/17/2003

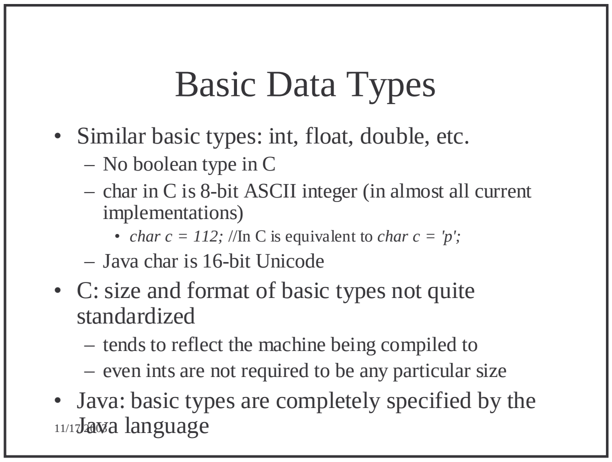


Operation and Context

- C is compiled to .o (object) file: machine code (almost)
- Linker combines .o files to an executable (runnable) machine language program (e.g. .exe)
- Loader/OS places executable in memory and starts at main

- Java is compiled to .class file: intermediate pseudo-machine code called "byte code"
- Java Virtual Machine loads classes as needed, links references on the fly, interpreting byte code as it goes.

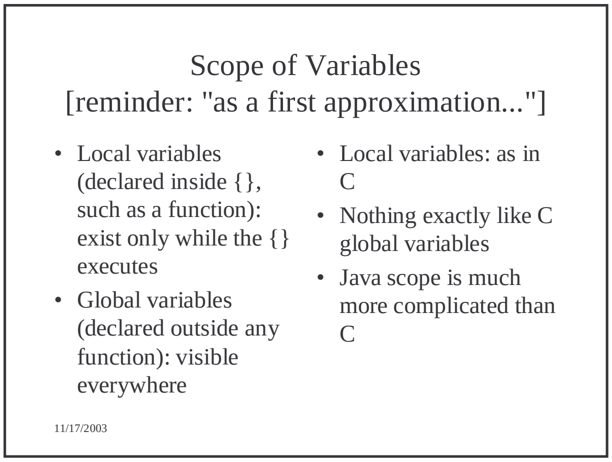
11/17/2003



Basic Data Types

- Similar basic types: int, float, double, etc.
  - No boolean type in C
  - char in C is 8-bit ASCII integer (in almost all current implementations)
    - *char c = 112;* //In C is equivalent to *char c = 'p';*
  - Java char is 16-bit Unicode
- C: size and format of basic types not quite standardized
  - tends to reflect the machine being compiled to
  - even ints are not required to be any particular size
- Java: basic types are completely specified by the Java language

11/17/2003

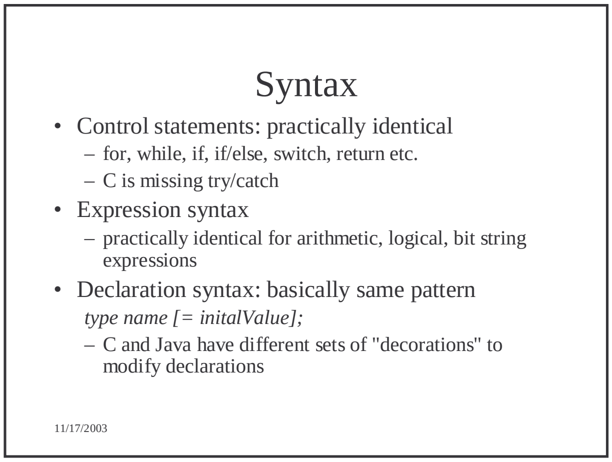


Scope of Variables [reminder: "as a first approximation..."]

- Local variables (declared inside {}, such as a function): exist only while the {} executes
- Global variables (declared outside any function): visible everywhere

- Local variables: as in C
- Nothing exactly like C global variables
- Java scope is much more complicated than C

11/17/2003



Syntax

- Control statements: practically identical
  - for, while, if, if/else, switch, return etc.
  - C is missing try/catch
- Expression syntax
  - practically identical for arithmetic, logical, bit string expressions
- Declaration syntax: basically same pattern

  type name [= initalValue];
  - C and Java have different sets of "decorations" to modify declarations

11/17/2003

Function Syntax

- Function (method) declaration: basically same pattern

  returnType funcName (paramList) {

  statements

  }

  One difference: all declarations must precede all statements (at least until very recent versions of C)

Pointers vs References

- A pointer in C is a variable which contains the address of (reference to) another variable
  - Special syntax used when declaring and using pointers
  - if *ptr* is a pointer, **ptr* is what it points to
- In Java, objects are always accessed through references.
  - The language syntax does not distinguish between an object and the reference (pointer) to it
  - References are strictly typed
  - Memory allocation and deallocation are automatic
- It's a big deal. Pointer and memory management errors are frequent and serious in C programs.

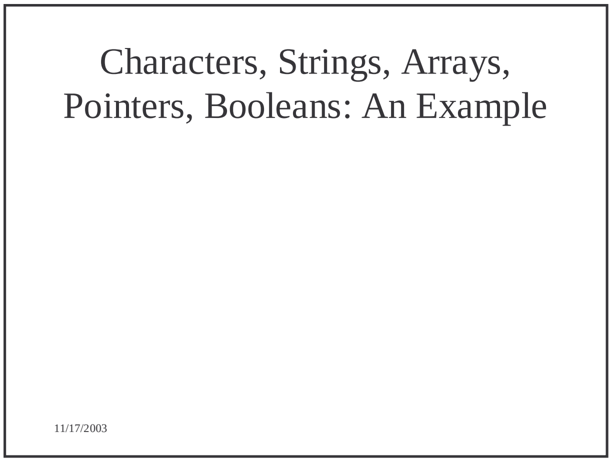
Arrays

- Similar in the two languages when [] notation is used
  - One big difference: in C, the size of the array is not encapsulated in it in any way
- In C, an additional view of an array is possible (and common):
  - The array name is viewed as a pointer, and array elements are offsets from this pointer.
  - lends itself to translation into machine code

Strings

- No string type in C
- Strings are represented by a universally-acknowledged convention: array of characters terminated by a null
- Unsafe but efficient

- String is a class in Java, in the basic java.lang package.
- Safe but not always efficient



Characters, Strings, Arrays, Pointers, Booleans: An Example

11/17/2003