# 

#### Principles of Computer Science II Nadeem Abdul Hamid

CSC121A - Spring 2005

Lecture Slides 11 - C Functions and Structured Programming



- Relational, equality, and logical expressions evaluate to int values 1 (true) or 0 (false)
- Expressions are parsed according to precedence and associativity rules
  - Rules of parsing C are standardized; order of evaluation is not (except for, ? && || operators)
- Statement forms
   sequence, empty, assignment, compound (block), conditional (if/if-else/switch), looping (for/while/dowhile), goto, continue, break

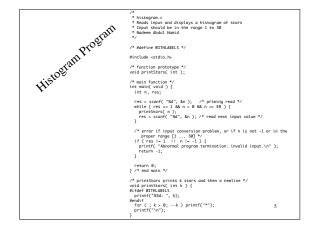
2

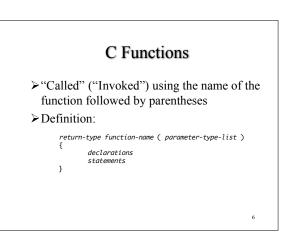
### Structured Programming

- A problem-solving strategy and programming methodology
  - Flow of control be as simple as possible
  - Program construction using top-down design
- Top-down design (stepwise refinement)
  - Repeatedly decompose problem into smaller problems, until you have a collection of small problems or tasks which can be invidually coded very easily
- Code for this decomposition is written using the C *function* mechanism (similar to methods in Java)

## Histogram Program

➢ Write a program that displays a histogram (bar chart of \*s) based on input read from a file. The file contains an arbitrarily-long list of numbers between 1 and 30. The last number in the list is followed by a -1.

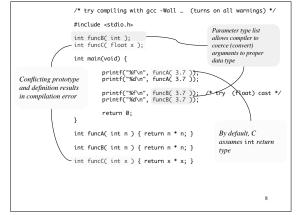






return-type function-name ( parameter-type-list ) ;

- Used to declare functions before they are used
  Identifiers in the type list are for documentation only
  - ignored by compiler
    Variable number of arguments specified using ... (printf function)
- One of most important improvements of ANSI C over traditional C
  - Allow compiler to validate function calls
  - Values passed to functions are coerced, as necessary



#### Compiler's View of Functions

- Function declarations generated in different ways by the compiler
  - Function call
  - Compiler assumes default declaration, returning int and no assumptions about parameters
  - Function definition
    - ANSI C style gives return type and parameter types

      Exercise mototume
  - Function prototype
     Special case of function declaration
    - Header files mostly contain these prototypes

#### Declarations, Prototypes, Definitions

- Function declaration specifies interface between function and rest of world (return type, argument types)
- Function prototype is an ANSI-style function declaration
- Function definition gives same info as declaration with names for arguments and block of code

10

12

## Other Features of C Functions

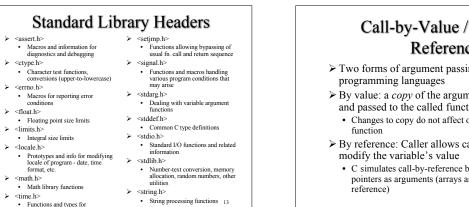
- ➢ If no return type is specified for a function, compiler assumes int
  - but it is better style to always indicate the return type

11

• Similarly, if a parameter's type is not specified, the compiler assumes int

C Standard Library

- Be familiar with the functions in the library (Appendix A)
- > Whenever possible, reuse functions from the C library
  - Reduces development time
  - · Increases program portability





## Call-by-Value / Call-by-Reference

- > Two forms of argument passing common in
- > By value: a *copy* of the argument's value is made and passed to the called function
- · Changes to copy do not affect original value in calling
- > By reference: Caller allows called function to
  - · C simulates call-by-reference by passing addresses and pointers as arguments (arrays are always passed by

14

