

Test #2 Overview

Material covered

- Chapters 5 - 6 in Introduction to Programming with C++, 3rd Edition by Liang
- Homework Assignments: Ch5-HW and Ch6-HW

Format (similar to previous tests)

- No books, no notes, no computers
- Types of problems includes:
 - Determining the output of programs on the test
 - Some T/F, multiple-choice, short answer, etc.
 - Writing programs or instructions to accomplish specified tasks.
- Very detail-oriented. Be prepared!

Items provided on the test (also see document on web site)

Tables of ASCII Codes and Operator Precedence

Chapter 5 – Iterative (looping) Structures

While loops

- form of structure
- use of pre-test, so loop variable must be initialized before loop
- 0 or more passes through the loop
- conditional statements

Do while loops

- form of structure
- use of post-test, no initialization of loop variable required before loop
- 1 or more passes through the loop
- conditional statements

For loops

- form of structure
- perhaps best structure for a specific number of iterations

Nested loops (while loops, do while loops, for loops)

Tracing through loops – using a table suggested

Infinite loops

- form
- can be implemented with while loops, do while loops, or for loops
- use **break** statement for exiting loops

Chapter 6 – Functions

Library-defined functions versus user-defined functions

Functions may:

- return no value
- return one value (through the return statement)
- return two or more values (using reference parameters)

Three parts to using/defining functions:

- function declaration (or prototype)
- function call
- function definition

Dummy arguments

- variable names do not need to match between the function call and the function definition
- the number, order, and type of the variables must match

Function arguments – may be value parameters or reference parameters

- Value parameters (or input parameters or copy parameters) – input arguments
- Reference parameters – follow type by & - used for outputs mainly, but can be inputs also

Overloading functions

- two or more versions of the same functions can be defined with different numbers or arguments and/or different types
- C++ determines which version of the function to use

Default inputs – if default values are assigned to arguments in the prototype, the arguments can be omitted in the function call and the default values will be used.

Scope – local versus global variables. Do not use global variables to avoid passing arguments!

No questions on creating custom libraries of functions