CIS 251 (C++ Programming)
Exam 2 Study Guide
Jefferson State Community College • Instructor: T. Battles

Format

- Twelve multiple-choice questions, including terminology, concepts, and syntax
- A hand-tracing problem: given a complete program, indicate the output the program generates
- Coding: write the code to solve a specific problem

Source Material: Chapters 4 – 6 in the textbook, with emphasis on concepts in the lectures (these chapters depend on material from the earlier chapters)

Exam aides: One double-sided or two single-sided printed pages up to letter-sized (8 ½" x 11") with any material (notes, sample code) that would be helpful for any section of the exam. You may use a basic or scientific calculator (standalone, not as an app), but otherwise **no interaction with computers, cell phones, or other electronic devices is allowed.**

Key Elements to Study

General

- The `include` statements required for the various functionalities described in these chapters

Chapter 4

- Predefined functions
    - Math functions
    - Random numbers
        - Getting a random number in a specific range
        - Seeding the random number generator with the current time
    - Type casting
- Defining functions
    - Function prototypes / declarations (syntax, placement)
    - Function headers
    - Function bodies (including the `return` statement)
- Calling functions
    - Arguments (matching the parameters)
    - Dealing with return values
- General terminology: black box, information hiding, procedural abstraction
- Functions and variables
    - Local variables
    - Global variables
    - Global constants
- Overloading
    - Using the same function name with different parameter lists
    - Calling different definitions

Chapter 5

- `void` functions
  - Writing a `void` function prototype and definition
  - Calling a `void` function
  - Writing a `return` statement in a `void` function
- Reference parameters
  - Syntax (in the prototype and the definition header)
  - Allowed arguments
  - Behavior
  - Mixture of value and reference parameters
- Procedural abstraction: preconditions, postconditions
- Program testing
  - Boundary values and ranges
  - Stubs and drivers
- Debugging techniques
  - Adding extra output statements
  - Using the `assert` macro

Chapter 6

- Data types for file streams
- Opening a file
  - Checking for success / failure when opening an input file
  - Overwriting versus appending to an output file
  - Letting the user specify the file name (data type of the file name variable)
- Performing reads / writes
  - Syntax compared to console input / output
  - Reading to the end of an input file
- Closing a file
  - Syntax
  - Why this is necessary
- Output formatting
  - Applied to file output
  - Flags for `setf` / `unsetf`
  - `precision` and `width`
  - Manipulator functions
- Streams as arguments to functions
  - Passing by reference
  - Writing a function that can handle either console or file input or output
- Character I/O
  - `get`, `put`, and `putback`
  - Clearing unwanted input to the end of the line
  - Issues with mixing character I/O and normal I/O
- Default arguments
  - Syntax in the function header
  - Calling a function with varying argument list lengths
- Character functions
  - Conversion: using the value returned
  - Evaluation: return data type