Programming 1 Honors

Final Study Guide

April 25, 2017

The test consists of

1. 15 multiple choice questions - 30 points
2. 4 “find the output” questions - 20 points
3. 3 code writing questions - 10 + 15 + 15 = 40 points
4. 1 code debugging question - 10 points
5. 4 short answer questions - 20 points

General details:

- You will have an opportunity to earn 20 extra credit points.
- Please try and attempt all questions. You get points for trying.
- The test is cumulative and would include questions from pretty much everything covered in the course.
- You won’t be asked to write code on anything covered after Classes (emuns, C concepts and binary arithmetic)
- Anything from the homeworks / quizzes / in class examples / exercises / slides is fair game. You don’t need to look for more material.
- Code debugging is mostly syntax based (missing brackets, semicolons, etc.)
- The code writing questions will be heavily based on the homeworks and class examples and exercises, with some modifications.
- The multiple choice and the debugging questions will test your familiarity with the C++ language and syntax. The code writing questions will test your knowledge of programming.
- Making me laugh might gain you points (depends on the quality of the joke).

Topics to study

- Basic C++ Syntax
  - Writing a basic C++ program stub: including required libraries, adding namespaces and writing the main function.
  - Simple statements - syntax.
  - Comments.
  - Reserved words, literals and escape sequences.
  - Style guidelines.

- Primitive Data types, variables, operators, and sequential execution.
  - Naming, declaring and initializing variables.
  - Primitive data types.
  - Type Conversions - implicit and explicit.
  - Arithmetic Operators and operator precedence.
- I/O - printing and reading values from the user.
  - cout statements - printing literals and variables.
  - Precision for floating point variables.
  - Using cin to read data of different kinds.
- Selection statements and loops
  - Relational and logical operators.
  - Writing simple, multiple and nested if statements.
  - switch - case statements.
  - while, do-while and for loops
  - break and continue statements.
- Writing functions in C++.
  - Writing simple functions.
  - Passing arguments and returning values. Pass by value and pass by reference.
  - Scope of local and global variables.
  - Function overloading and default parameters.
- Arrays
  - Declaring and initializing an array.
  - Reading in array values from the user and printing arrays.
  - Basic array operations - looking for a number, math with array elements, etc.
  - Passing arrays as parameters to functions.
  - Multi dimensional arrays.
- Strings
  - Cstrings - arrays of characters.
  - string objects
  - Reading strings with getline - both options
  - cstring, cctype and string libraries
  - Basic string operations - counting different kinds of characters, finding and replacing substrings, etc.
  - Declaring, initializing, comparing and concatenating strings
  - Arrays of strings.
- Pointers
  - Declaring, initializing and dereferencing pointers.

  - null pointer, reinterpret cast and pointer arithmetic.
  - Pass by address, arrays with pointers, cstrings with pointers.
  - new and delete operators, dynamic memory allocation.

- Structures
  - Creating structures, declaring variables of the new type, creating nested structures, arrays of structures, pointers to structures.
  - The dot, arrow and assignment operators.
  - Structures and functions - passing and returning.
- File Operations
  - Text Files and Streams
  - How to read from a text file, how to write to a text file.
  - Objects of ifstream and ofstream and some associated member functions, like eof().
  - Character I/O - get(), put() and ignore()
  - Streams and functions
- Object Orieneted Programming
  - Classes and Objects - Declaring, defining and using
  - Data Hiding and Protection levels
  - Constructors and initialization of objects
  - Accessors and Mutators
  - Aggregation and embedded objects
  - Multi File Compilations
- Enumerations
- Binary Arithmetic
  - Storing integers and flating point numbers in memory
  - Adding binary numbers
  - Bitwise operators - AND, OR, negation, shifting
- You don't need to study from outside sources. The test is made entirely from the notes, quizzes and assignments.

Sample Questions

1. Which of the following is a C++ reserved word?
   (a) file
   (b) ifstream
   (c) ofstream
   (d) enum

2. Which of the following strings will be placed before ”banana” in lexicographical order?
   (a) Peach

(b) Strawberry
(c) 2 green apples
(d) All of the above

3. Consider the following function declaration:
```
int count (char *st)
```
If you had a for loop (loop variable i) iterating through the Cstring in the function, which of the following lines would result in an error?

   (a) cout<<st[i];
   (b) st[i++] = ’X’;
   (c) cout<<*st[i];
   (d) cout<<*(++st);

4. A perfect number is defined as a positive integer that is equal to the sum of its factors (not including itself). For example, 6 is a perfect number, because the sum of the factors of 6 = 1 + 2 + 3 = 6. Write a C++ program to print all the perfect numbers between 0 and 1000
Sample Run:
```
6
28
496
```

5. Write a C++ program to create an array of 200 integers. Read in the values from a file called `input.txt`. Then, print the most frequent element in the array.

6. Write a C++ program with the following requirements:

   - Write a class called `Drink`. The class has an double attribute called `size` that holds the size of the drink in ounces. The class also has a parametrized constructor and a function called `print` that prints the size.
   - Write a class called `Coffee`. The class has a Drink attribute and an integer attribute called `numShots` that denotes number of espresso shots in the drink. The class also has a parametrized constructor and a function called `print` that prints both the size and the number of shots.
   - In the main function, create an array of Coffee of size 4. Give the drinks values of your choice. Then invoke the `print` function for each of them.

   Sample Run:
```
Drink 1:
Size: 16 Ounces
NumShots: 4
Drink 2:
Size: 8 Ounces
NumShots: 3
Drink 3:
Size: 12 Ounces
NumShots: 2
Drink 4:
Size: 5.5 Ounces
NumShots: 3
```

7. What is data hiding? How does having different access levels help in implementing data hiding?