The test consists of

1. 15 multiple choice questions - 30 points
2. 3 “find the output” questions - 15 points
3. 2 code writing questions - $15 + 25 = 40$ points
4. 1 code debugging question - 15 points
5. 3 short answer questions - 15 points

General details:

- You will have an opportunity to earn 15 extra credit points.
- Please try and attempt all questions. You get points for trying.
- While the test is cumulative, the questions will focus on the topics after the first midterm (functions and forward).
- 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 - simple statements, comments, reserved words, literals, etc.
- Primitive Data types, variables, operators, and sequential execution.
- I/O - printing and reading values from the user.
- Selection statements and loops
- 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, ar-
    rays of structures, pointers to structures.
  – The dot, arrow and assignment operators.
  – Structures and functions - passing and returning.

• Studying the topics listed above will be enough to pass the test. To get a 100, you would be
  required to study everything on the notes.

• 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 NOT a C++ reserved word?
   (a) new
   (b) pointer
   (c) struct
   (d) return
2. Given an array of doubles starting at address 5000, what is the starting address of element 7 of the array?
   (a) 5000
   (b) 5048
   (c) 5056
   (d) 5064

3. Which of the following functions will return 0 (false) for the character ‘A’?
   (a) isalnum
   (b) isprint
   (c) islower
   (d) islalpha

4. Write a C++ function that accepts a string as a parameter and returns the number of sentences in the string. You don’t have to read in the input or print the output to match the sample run. Just write the function.
   Sample Run:
   Input:
   Never gonna give you up. Never gonna let you down. Never gonna run around and desert you.
   Output:
   The string has 3 sentences.

5. Write a C++ function that accepts a double array and its size. This array represents the side lengths of a bunch of squares. Dynamically create an array of doubles of equal size. Calculate areas of squares and return the array. Once again, you don’t have to read or print anything.
   Sample Run:
   Input array:
   2.5, 12, 5.9, 14.82, 0.8
   Output array:
   6.25, 144, 34.81, 219.6324, 0.64

6. Write a C++ program that creates a structure with 3 integer parts - whole, numerator and denominator. Write a function called print, that accepts one structure variable as a parameter and prints the fraction as an improper fraction. In the main function, create a variable of type Fraction, read in the 3 values, and the call the print function.
   Sample Run:
   Enter the whole number: 3
   Enter the numerator: 7
   Enter the denominator: 11
   The fraction is: 40/7

7. Figure out the output generated by the following code snippet:
   ```cpp
   int mat[3][3] = {{1,5,19},{6,-2,10},{12,8,5}};
   int sum=0;
   for(int i=0;i<3;i++)
       sum += mat[i][i];
   cout<<"The sum is "<><<endl;
   ```

8. What is dereferencing a pointer? Explain with an example.