Final Exam F24 Final V2

CSCI 13500: Software Analysis and Design 1

Hunter College, City University of New York

Dec 19, 2024, 1:45 PM - 3:45 PM, N118

Exam Rules

- Show all your work. Your grade will be based on the work shown.
- The exam is closed book and closed notes with the exception of a provided cheat sheet.
- When taking the exam, you may bring pens and pencils.
- Scratch paper is provided. For your convenience, you may take the scratch paper and cheat sheet off. But make sure **not** to put solutions to the scratch paper.
- You may not use a computer, calculator, tablet, phone, earbuds, or other electronic device.
- **Do not open this exam until instructed to do so.**

Hunter College regards acts of academic dishonesty (e.g., plagiarism, cheating on examinations, obtaining unfair advantage, and falsification of records and official documents) as serious offenses against the values of intellectual honesty. The College is committed to enforcing the CUNY Policy on Academic Integrity and will pursue cases of academic dishonesty according to the Hunter College Academic Integrity Procedures.

I understand that all cases of academic dishonesty will be reported to the Dean of Students and will result in sanctions.
Name:
EmpID: □□□□□□□□
Email:
Signature:

1 (30 points) Answer the following questions.

(1) Given `string groceries[] = {"milk", "apple", "green onion"}`, what is groceries[2].substr(2, 5)?

(2) Given a declaration `std::vector<int> v(5, 6);`, what is the value of v.size()?

(3) What possible numbers does code `2 + rand() % 6` generate?

(4) Given `string numStr = std::to_string(1) + "35";`, where `to_string` converts an integer to a string. What is the value for numStr?

(5) What is the value of `2 + 7 % 4 / 2` in C++?

(6) Write **header** of a function called max, given an array of characters (type char) with *size* many elements, return the largest ASCII code of all the elements in the array.

(7) Declare class Coord as follows.

```
class Coord {
public:
    double x;
    double y;
};
```

Declare a Coord object `point` and initialize its x as 7 and y as 8.

(8) Given `int grades[] = {73, 100, 99, 62};` What is the value of `*(grades + 2)`?

(9) Given the following code segment.

```
void foo(int *pm, int *pn);

int main() {
    int m = 1;
    int n = 2;

    //TODO: write a statement to call foo using appropriate attributes of m and n.

    return 0;
}
```

(10) Suppose we have main function defined as follows. And calling `foo(s, t)`, the values of `s` and `t` are swapped. That is, `s` becomes “hi” and `t` becomes “hello”.

```
int main() {
   string s = "hello";
   string t = "hi";
   foo(s, t);
   return 0;
}
```

What is the **header** of function foo? Suppose its return type is void.

(11) What is output for the following code?

```
    int a = 3;
    int* p = &a;
    a += 2;
    cout << *p << endl;
```

(12) What is the output for the following code?

```
    vector<int> nums = {1, 2, 0};

    int count = 0;
    for (int i = 0; i < nums.size(); i++)
        if (nums[i] % 2 != 0)
           count++;

    cout << count << endl;
```

(13) What the output of the following code?

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    for (int row = 0; row < 3; row++) {
        for (int col = 0; col < 4; col++) {
            if (col >= 2)
               cout << "#";
            else cout << "-";
        }
        cout << endl;
    }
    return 0;
}
```

(14) What is panel after slide left operation?

1	5	3
4		7
2	6	8

(15) Suppose in Project 3, data member `bins` have the following values,

`{{2, 1, 3}, {1, 1, 3}, {2, 2}, {3}}`,

After moving eligible element(s), according to rules listed in Project 3, from the leftmost bin to the rightmost bin, what are the elements in the **leftmost** bin?

2 (15 points) Answer the following questions.

(1) Define function `countSuccessiveFrontElms`, for an given array of integers with its size, return the number of successive (aka consecutive) elements in the front of this array.

For example, call the function with array with values 0, 0, 1, 0, the size of array is 4. There are two zeros residing successively (aka consecutively) in the front of array, the return is 2. Note that the rightmost zero is not adjacent with other zeros in the front, so it is not counted as part of the results.

In main function, write the following statements. No need to write the full definition of main function.

Define an int array with elements 0, 0, 1, 0.

Call and print the number of successive front elements of the above array.

(2) Define function `searchLast`, given an array of integers, its size, and a target (an integer), return a pointer to the **last** occurrence of the target in an array, or nullptr if there is no match.

For example, suppose an array has elements 1, 2, 3, 2, if the target is 2, then the return of the function is a pointer to the rightmost element. if the target is 4, then the return is nullptr.

3 (10 points) Programming exercise on class

1. Define class for representing weight in pounds (also called lb) and ounces. It is reasonable to define it to have two integer fields:

   lb for the number of pounds, and oz for the number of ounces. Note that a pound has 16 ounces, so we need to make sure that oz is in [0, 15].

   ```
   class Weight {
   public:
       int lb;
       int oz; //value in [0, 15]
   };
   ```

   Define Weight minusOzs(Weight curr, int ozVal);

   The function should create and return a weight object that is ozVal ounces fewer than curr. Note that 1 lb = 16 oz. Example:

   ```
   minusOzs({3, 10}, 30) // should return {1, 12}
   ```

   Reason: 3 lbs 10 ounces is 3 * 16 + 10 = 58 ounces. Then 58 - 30 = 28 ounces, which equals 1 lb and 12 ounces.

   For simplicity, we assume that total number ounces for `curr` is larger than or equal to `ozVal`.

2. In main function, write the following statements. No need to define the whole main function.

   - Declare and instantiate `curr` as a Weight object with lb equals 3 and oz equals 10.
   - Declare and instanitate a Weight object called `lighter` that is 30 ounces fewer than `curr`. You may call `minusOzs` with appropriate parameters.

4 (10 points) Write codes of vector

Define a function called `choose`, for a vector `v` of characters (type char), return a vector with all the elements from `v` that are lowercase letters, in the same order. In English, lowercase letters are `'a' - 'z'`

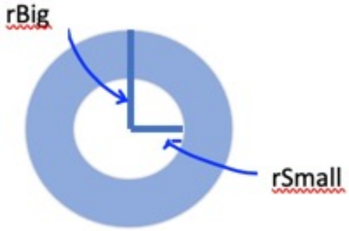
For example, given a vector of characters with elements `'a', 'B', '#', '1', 'c'`, the return is a vector with elements `'a', 'c'`.

Hint: int islower (int c); check if character is lowercase letter. A value different from zero (i.e., true) if indeed c is a lowercase alphabetic letter. Zero (i.e., false) otherwise.

islower is from cctype library. However, you do not need to include library in your code.

5 (15 points) Define class for ring shape.

1. In mathematics, an annulus is the region between two concentric circles. Generally called a ring. It has two parameters:

   (a) radius of the inner or the smaller circle **rSmall**

   (b) radius of the outer or the bigger circle **rBig**

2. **Assume that Ring.hpp is provided** where data members **rSmall** and **rBig** are declared as double types. **Your job** is to define **Ring.cpp** with the following requirement.

3. Define a default constructor, set data members **rSmall** to be 1 and **rBig** to be 2.

4. Define a non-default constructor, which takes formal parameters rSmall and rBig, both are double types.

   (a) If both rSmall and rBig are positive and rBig is larger than rSmall, set data member **rSmall** by given parameter rSmall and set data member **rBig** by given parameter rBig.

   (b) otherwise, set data members **rSmall** to be 1 and **rBig** to be 2.

5. Define method **getArea**, return the value of $\pi(rBig)^2$ - $\pi(rSmall)^2$, where π is defined as `M_PI` in `cmath` library. Note that *rBig* and *rSmall* are data members, not r * Big or r * Small.

6. Define method **getPerimeter**, which returns $2\pi(rSmall) + 2\pi(rBig)$. Note that $rBig$ and $rSmall$ are data members, not r * Big or r * Small.

Define **RingTest.cpp**, do the following:

7. Create a Ring object named **donut** from its default constructor.

8. Find out and print the area of **donut**.

9. Find out and print the perimeter of **donut**.

6 (10 points) function on vectors

Define a function called `compare`, given two vectors of strings, if they have the same number of elements, find out whether the length of **every** element in the first is smaller than that of the same-index element in the second vector, if yes, return true, otherwise, return false. If these vectors do not have the same number of elements, return false.

For example, if the first vector is {”hello”, ”hi”} and the second vector is {”abcdef”, ”abc”, ”123”}, the return is false. Reason: the two vectors have different number of elements.

If the first vector is {”hello”, ”hi”, ”how”} and the second vector is {”hellooo”, ”hey”, ”abcd”}, return true. Reason: both vectors have the same number of elements. Furthermore, the length of `"hello"` is smaller than that of `"hellooo"`, the length of `"hi"` is smaller than the length of `"hey"`, and the length of `"how"` is smaller than the length of `"abcd"`.

If the first vector is {”hello”, ”hi”} and the second vector is {”abcdef”, ”ab”}, the return is false. Reason: even though the number of elements of the two vectors are the same, the length of the second element ”hi” of the first vector is not smaller than the length of the second element ”ab” of the second vector.

7 (10 points) Define recursive function

Define a recursive function `printArray`, given an array of double type numbers (number with decimals) with size, print all numbers from the first one to the last one, separated by a space, in the same line.

For example, if an array with elements 1.1, 2.2, and 3.3, the print is

1.1 2.2 3.3

Warning: If you do not use recursion, you will not get any point.

No repetition statement, global or static variables are allowed in this function.

Use array, not vector.