

**FINAL EXAM F25 FINAL V1**  
CSCI 13500: Software Analysis and Design 1  
Hunter College, City University of New York

December 18, 2025, 9:00 AM - 11:00 AM, N 118

## 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, i-watch, or any other electronic device. **Any device found outside a bag will be considered a violation of academic integrity and will result in a grade of zero.**
- Unless the problem explicitly requests, no need to include libraries and using namespace std.
- **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 will not touch any electronic device, including but not limit to cell phone, airpod, or electronic watch during the whole exam, except when showing a virtual ID.

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 `std::string colors[] = {"green", "light blue", "cyan"}`, what is `colors[1].substr(3)`?

(2) Given a declaration `std::vector<int> v(10, 3); v.pop_back();`, what is the value of `v.size()`?

(3) How to generate a random integer in [100, 160]?

(4) Given `std::string str = std::to_string(135) + "-01";`, where `to_string` converts an integer to a string. What is the value for `str`?

(5) What is the value of `2 * (1 + 3) % 5` in C++?

(6) Write **header** of a function called `remove`, given an array `arr` of **string** type with `size` many elements, remove the last element from the array. Return type of the function is **void**.

(7) Declare class `Coord3D` as follows.

```
1 class Coord3D {  
2     public:  
3         double x;  
4         double y;  
5         double z;  
6     };
```

Declare a `Coord3D` object `point` and initialize its `x, y, z` to be `1, 2, 3`, respectively.

- (8) Given int grades[] = {76, 100, 72, 96}; What is the value of \*(grades+1) - 1?

- (9) Given the following code segment, finish the TODO part.

```
1 int main() {
2     std::cout << "Enter size of an array: ";
3     int size;
4     std::cin >> size;
5     while (size <= 0) {
6         std::cout << "Re-enter, the size is not positive: ";
7         std::cin >> size;
8     }
9
10 //TODO: write a statement to declare and initialize a pointer arr to
11 //a dynamically allocated array with size elements of double type.
12 //WRITE YOUR ANSWER IN THE FOLLOWING BOX.
```

13  
14  
15  
16  
17

```
//omit other code ...
return 0;
}
```

- (10) Suppose we have main function defined as follows.

```
1 int main() {
2     double m = 1.1;
3     double n = 2.2;
4     //In foo, if m < n, exchange the values of m and n.
5     //The return type is void.
6     foo(&m, &n);
7
8     std::cout << m << " " << n << " " << std::endl;
9     //expected output: 2.2 1.1
10    return 0;
11 }
```

What is the **header** of function foo?

(11) What is output calling `foo` with an array with elements -5, -6, 0, 7, 2 and its corresponding size?

```
1 int foo(int* arr, int size) {
2     int i = 0;
3     while (i < size && arr[i] <= 0) {
4         i++;
5     }
6
7     return i;
8 }
```

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

```
1 std::vector<int> nums = {1, -2, 2, -1};
2
3 int result = 0;
4 for (int i = 0; i < nums.size(); i++) {
5     if (nums[i] > 0) {
6         result += nums[i];
7     }
8 }
9
10 std::cout << result << std::endl;
```

(13) Given `arr` with values 1, -2, 2, 6, 0 with size 5, what will be the value of `arr` after calling `foo` on `arr` and `size`?

```
1 void foo(int arr[], int size) {
2     int i = 0;
3     int j = size-1;
4     while (i < j) {
5         std::swap(arr[i], arr[j]);
6         i++;
7         j--;
8     }
9 }
```

- (14) What is the output of the following code? For simplicity, we omit libraries.

```
1 int main() {
2     int size = 3;
3     for (int row = 0; row < size; row++) {
4         for (int col = 0; col < size; col++) {
5             if (row + col < size) {
6                 std::cout << "*";
7             }
8             else {
9                 std::cout << "-";
10            }
11        }
12        std::cout << std::endl;
13    }
14
15    return 0;
16 }
```

- (15) What is the output of the following code? Assume that libraries are set up.

```
1 std::string foo(std::vector<std::string> v, char ch);
2
3 int main() {
4     std::vector<std::string> v = {"apple", "", "blue", "corn", "berry", "ant"};
5     std::cout << foo(v, 'a') << std::endl;
6     return 0;
7 }
8
9 std::string foo(std::vector<std::string> v, char ch) {
10     std::string result;
11     for (int i = 0; i < v.size(); i++) {
12         if (v[i] != "" && v[i][0] == ch) {
13             result += v[i] + ",";
14         }
15     }
16
17     return result;
18 }
```

## 2 (15 points) Answer the following questions.

- (1) Define a function, `getInitial`, for a string in the format of first name first, followed by a space symbol (' '), followed by last name, return a string representing the initial.

For example, given a string with value "George Washington", the returned string is "GW".

Hint: extract the first name and the last name. Instantiate an empty string. Concatenate the empty string with the first letter of the first name, then the first letter of the last name. You might use the following methods from string class.

```
size_t find (char c, size_t pos = 0) const;
```

Searches the string for the first occurrence of the sequence specified by its arguments.

```
string substr (size_t pos = 0, size_t len = npos) const;
```

Returns a newly constructed string object with its value initialized to a copy of a substring of this object.

(2) Write a function `pointerToLastEven` that returns a **pointer** to the **last** occurrence (if there are more than one occurrence) of the even integer in an array of **int** type with *size* many elements.

If size is 0 or there is no even integer, return `nullptr`.

For example, given an array with elements 5, 4, 3, 2, 1, the return of the function is a pointer to element 2. Given an array with elements 1, 3, 5, the return is `nullptr`.

### 3 (10 points) Programming exercise on class

Define class `Length` to represent length in feet and inches. It is reasonable to define it to have two integer fields:

`foot` for the number of feet, and

`inch` for the number of inches. Note that a foot has 12 inches, so we need to make sure that `inch` is in  $[0, 11]$ .

**Define** non-member function `addInches`, given `Length` object `len` and integer parameter `toAdd`, the function should create and return a `Length` object that is the sum of `len` and `toAdd`. Example:

Suppose `len` is  $\{2, 7\}$  and `toAdd` is 30. Then the return of `addInches` function on the above parameters is  $\{5, 1\}$ .

Reason: 2 feet 7 inches is  $2 * 12 + 7 = 31$  inches. Add 31 by 30 is 61 inches, which equals 5 feet and 1 inch, where 5 is 61 divided by 12 and 1 is the remainder of 61 divided by 12.

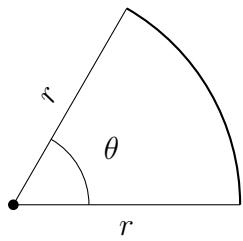
#### 4 (10 points) Write codes of vector

Define a function called `shortStrings`, for a vector `v` of strings and an int `n`, search the `v`, find out the strings whose lengths are smaller than `n`, put them to a vector of strings, in the same order. Return that vector.

Given a vector of strings with elements "hello", "", "hey", "hi" and integer 3, the return is vector with elements "", "hi".

## 5 (15 points) Define class.

1. Define an Arc as a portion of a circle.



- (a) radius of the circle  $r$   
(b) the **angle** between the two edges, represented as  $\theta$  in the above figure, in **radians**
2. Assume that **Arc.hpp** is provided where data members **r** and **angle** are declared as **double** types. Your job is to define **Arc.cpp** with the following requirement.
3. Define the default constructor, set data members **r** to be 1 and **angle** to be  $\pi/4$ , where  $\pi$  is represented by **M\_PI** in C++.

4. Define a non-default constructor, which takes formal parameters **r** and **angle**, both are double types.
  - (a) If **r** is positive and **angle** is strictly larger than 0 and **angle** is strictly smaller than  $2\pi$ , where  $\pi$  is represented by **M\_PI** in C++, set data member **r** by given parameter **r** and set data member **angle** by given parameter **angle**.
  - (b) otherwise, set data members **r** to be 1 and **angle** to be  $\pi/4$ , where  $\pi$  is represented by **M\_PI**.

5. Define method **getArea**, return the value of  $\frac{1}{2}r^2\text{angle}$ . Note that **r** and **angle** are data members.

6. Define method **setRadius**, if given parameter **r** is positive, reset data member **r** by given parameter **r**.

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

7. Create an Arc object named **shape** from its non-default constructor with the radius of the arc as 2 and the angle as  $\frac{\pi}{2}$ , where  $\pi$  is defined as **M\_PI** in C++.

8. Reset the radius to be 3.

9. Find out and print the area of shape.

## 6 (10 points) function on vectors

Define a function called `mixNums`, given a vector of `int` type `vec`, do the following:

Check whether all the elements in `vec` has at least a positive number **and** at least a negative number.

For example, if `vec` has values 1, -2, 0, the returned is true since there is one positive number 1 and one negative number -2.

If `vec` has values 1, 2, 3, the returned is false since there is no negative numbers.

## 7 (10 points) Recursive Function

Define a recursive function `maxLen`, given an array of `string` with size many elements, return the maximum length of all strings.

For example, for array with elements "hello", "hi", "how", the return is 5, since the longest string "hello" has 5 characters.

Hint: you should set the return type of `maxLen` as `size_t`, which is similar to `int` but represents only non-negative integers.

You may use `std::max` function from `algorithm` library, which takes two parameters of the same types and return the maximum. Or, you may use if-else statement to replace `std::max`.

**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.**