Answer:

Final Exam S25 Final V4
CSCI 13500: Software Analysis and Design 1
Hunter College, City University of New York

June 23, 2025, 11:30 AM - 1:30 PM, N 1001 D

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. **If any electronic device is out of backpack, you will get zero for this exam.**
- 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 `string houses[] = {"ranch", "cape cod", "townhouse"}`, what is houses[1].substr(2).length()?

Answer: houses[1].substr(2).length() is 6. houses[1].substr(2) returns "pe cod". Explanation: houses[1] is the second element of array of strings, which is "cape cod". Expression houses[1].substr(2) is the substring from the third letter – index 2 – of this string to the end, which is substring "pe cod". The number of characters in this substring is 6.

(2) Given a declaration `std::vector<int> v(2, 1); v.push_back(-4);`, what is the value of v[1] + v[2]?

Answer: -3

explanation: `vector<int> v(2, 1);` creates a vector of integers with 2 elements, each element has value 1. After push element -4 to it, v has one more element. Hence, vector v has elements 1, 1, -4, so v[1] + v[2] is 1 - 4 = -3.

(3) What is the **maximum** integer that expression (rand() % 5 + 6) % 4 can generate?

Answer: answer: 3

`rand() % 5` generates a random int in [0, 4].

`rand() % 5 + 6` generates a random int in [6, 10].

The maximum integers of (rand() % 5 + 6) % 4 is 3. For example, 6 % 4 returns 2, and 7 % 4 returns 3, 8 % 4 returns 0, and so on.

(4) Given `int num = std::to_string(135).size() + 3;`, where `to_string` converts an integer to a string and `size` method returns the number of characters of a string. What is the value for num?

Answer: answer: 6.

(5) What is the value of `2 + (6 + 2) / (5 % 6)` in C++?

Answer: 3

Explanation: (1) expression in parentheses runs first. Run 8 / (5 % 6). It is like to divide 5 pens among 8 persons, each person get 0 pen, 5 pens left. (2) Operator / has higher precedence than +. So 8 / 5 runs first. Divide 8 pens among 5 persons, each person get 1 pen. (3) 2 + 1 returns 3.

(6) Write **header** of a function called percentOdd, given an array *arr* of string type with *size* many elements, return the percentage (may contain decimal parts) of elements in array with odd size.

Answer: `double percentOdd(string* arr, int size);` or `double percentOdd(string arr[], int s`

(7) Declare class Student as follows.

```
class Student {
public:
   string name;
```

```
    string major;
};
```

Declare a Student object `me` and initialize its name as *your name* and major as *your major*.

Answer:

```
Student me = {"Tong YI", "Computer Science"};
```

or

```
Student me{"Tong Yi", "Computer Science"};
```

or

```
Student me;
me.name = "Tong Yi";
me.major = "Computer Science";
```

(8) Given `string names[] = {"Ann", "Bob", "Charles"};` What is the value of `*(names + 1) + " Greg"`?

Answer: "Bob Greg"

(9) Given the following code segment.

```
int main() {
    int numRows = 10;
    double** arr;

    //TODO: write a statement to initialize arr to be
    //a dynamically allocated array with numRows elements of double* type.
    //WRITE YOUR ANSWER IN THE FOLLOWING BOX.


    return 0;
}
```

Answer: `arr = new double*[numRows];`

Here is a complete code of an example.

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

int main() {
    int numRows = 10;

    double** arr = new double*[numRows];
    for (int i = 0; i < numRows; i++)
        arr[i] = new double[i+1];

    double value = 1.1;
    for (int i = 0; i < numRows; i++)
        for (int j = 0; j < i+1; j++) {
            arr[i][j] = value;
            value++;
        }

    for (int i = 0; i < numRows; i++) {
        for (int j = 0; j < i+1; j++)
            cout << arr[i][j] << " ";

        cout << endl;
    }

    for (int i = 0; i < numRows; i++) {
        delete[] arr[i];
        arr[i] = nullptr;
    }

    delete[] arr;
    arr = nullptr;
    return 0;
}
```

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

```
int main() {
   int m = 2;
   int n = 1;
   //In foo, if m > n, exchange the values of m and n, then return true,
   //otherwise, return false.
   bool b = foo(m, n);

   cout << m << " " << n << " " << boolalpha << b << endl;
   //expected output: 1 2 true
   return 0;
}
```

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

Answer: `bool foo(int& a, int &b);`

A complete code is shown as follows.

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

bool foo(int& m, int& n);

int main() {
   int m = 2;
   int n = 1;
   //In foo, if m > n, exchange the values of m and n, then return true,
   //otherwise, return false.
   bool b = foo(m, n);

   cout << m << " " << n << " " << boolalpha << b << endl;
   //expected output: 1 2 true
   return 0;
}

bool foo(int& m, int& n) {
    if (m > n) {
       swap(m, n);
       return true;
    }

    return false;
}
```

(11) What is output calling `foo` with an array with elements -5, -1, -5, 5 and its corresponding size?

```cpp
int foo(int* arr, int size) {
    int i = 0;
    while (i < size && arr[i] < 0)
        i++;

    return i;
}
```

Answer: 3

Explanation: the above loop is to return the index of the first non-negative element in the array; if every element of the array is negative, then return the number of elements of the array.

In this example, the first three elements are negative but not the fourth element, which is indexed at 3. Note that the index of the first element is 0.

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

```cpp
vector<int> nums = {2, 3, -1, 5};
```

```

int result = 1;
for (int i = 0; i < nums.size(); i++)
    if (nums[i] % 2 != 0)
       result *= nums[i];

cout << result << endl;
```

Answer: -15

Print out the product of all odd integers in the vector.

(13) What the output of the following code? For simplicity, we omit library and using namespace statement.

```
int main() {
    int numRows = 5;
    int numCols = 4;
    for (int row = 0; row < numRows; row++) {
        for (int col = 0; col < numCols; col++) {
            if (row >= numRows / 2 && col >= numCols / 2)
               cout << "b";
            else cout << "-";
        }
        cout << endl;
    }
    return 0;
}
```

Answer:

```
----
----
--bb
--bb
--bb
```

See a complete code.

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

int main() {
    int numRows = 5;
    int numCols = 4;
    for (int row = 0; row < numRows; row++) {
        for (int col = 0; col < numCols; col++) {
            if (row >= numRows / 2 && col >= numCols / 2)
               cout << "b";
            else cout << "-";
        }
        cout << endl;
    }
    return 0;
}
```

(14) What is the output of the following code? Assume that libraries and standard namespace are set up.

```
int foo(vector<string> v, char ch);

int main() {
    vector<string> v = {"", "jello", "bcc", "hello", "uc"};
    cout << foo(v, 'o') << endl;
    return 0;
}

int foo(vector<string> v, char ch) {
    int result = 0;
    for (int i = 0; i < v.size(); i++)
        if (v[i] != "" && v[i][v[i].length() - 1] == ch)
           result++;

    return result;
}
```

Answer: 2

Explanation: the number of strings in vector v that are ended with char ch.

(15) Given arr with values 1, -2, 7, 0, 8, -1 with size 6, what will be the value of arr after calling foo on arr and size?

```
void foo(int arr[], int size) {
    int value = arr[0];
    int i = 1;
    int j = size -1;
    while (i < j) {
        while (i < j && arr[i] <= value)
            i++;

        while (j > i && arr[j] > value)
            j--;

        if (i < j)
```

```
            swap(arr[i], arr[j]);
    }
    swap(arr[0], arr[i-1]);
}
```

Answer: 0 -2 -1 1 8 7

A complete code is as follows.

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

void foo(int arr[], int size);

int main() {
    //int arr[] = {3, -1, 2, 7, 6, -2}; //-1 -2 1 3 6 7
    int arr[] = {1, -2, 7, 0, 8, -1}; //0 -2 -1 1 8 7
    int size = sizeof(arr) / sizeof(arr[0]);

    foo(arr, size);

    for (int i = 0; i < size; i++)
        cout << arr[i] << " ";
    cout << endl;

    return 0;
}

void foo(int arr[], int size) {
    int value = arr[0];
    int i = 1;
    int j = size -1;
    while (i < j) {
        while (i < j && arr[i] <= value)
            i++;

        while (j > i && arr[j] > value)
            j--;

        if (i < j)
           swap(arr[i], arr[j]);
    }
    swap(arr[0], arr[i-1]);
}
```

2 (15 points) Answer the following questions.

(1) Define a function, `rem_succ_spaces`, for a string, return a string with all spaces after the last non-space character from the original string rmoved.

For example, given a string with value `"  hello, how are you    "`,

the returned string is `"  hello, how are you"`.

Hint: you might use the following functions from cctype library.

int isspace (int c); Check if character is space or not

Answer:

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

string rem_succ_spaces(string str);

int main() {
    cout << '"' << rem_succ_spaces(" hello, world ") << '"' << endl;
    //print out " hello, world"
    return 0;
}

string rem_succ_spaces(string str) {
    int i = str.length() - 1;
    while (i >= 0 && isspace(str[i]))
        i--;

    return str.substr(0, i+1);
}
```

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

If size is 0, return nullptr.

For example, suppose an array has elements `"or"`, `"hey"`, `"us"`, `"wow"`, then the return of the function is a pointer to string "us".

Answer:

```cpp
string* pointerToLastShortest(string arr[], int size) {
    if (size == 0)
       return nullptr;

    int idx = size-1;
    string shortest = arr[idx];
    for (int i = idx - 1; i >= 0; i--)
        if (arr[i].length() < shortest.length()) {
           idx = i;
           shortest = arr[i];
        }

    return arr + idx;
}
```

A complete code is as follows.

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

string* pointerToLastShortest(string arr[], int size);

int main() {
    string arr[] = {"or", "hey", "us", "wow"};
    int size = sizeof(arr) / sizeof(arr[0]);

    string *result = pointerToLastShortest(arr, size);
    //0x16cf9af88, actual print depends on systems.
    cout << result << endl;
    cout << result - arr << endl; //2, the third item in the array
    return 0;
}

string* pointerToLastShortest(string arr[], int size) {
    if (size == 0)
       return nullptr;

    int idx = size-1;
    string shortest = arr[idx];
    for (int i = idx - 1; i >= 0; i--)
        if (arr[i].length() < shortest.length()) {
           idx = i;
           shortest = arr[i];
        }

    return arr + idx;
}
```

3 (10 points) Programming exercise on class

1. Define class for representing 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 `multiply`, given Length object len and integer parameter times, the function should create and return a length object that is the product of `len` and `times`. Example:

   Suppose `len` is {2, 7} and `times` is 3. Then the return of multiply function on the above parameters is {7, 9}.

   Reason: 2 feet 7 inches is 2 * 12 + 7 = 31 inches. Multiply 31 by 3 is 93 inches, which equals 7 feet and 9 inches.

   Answer:

```cpp
class Length {
public:
    int foot;
    int inch; //value in [0, 11]
};
```

   Answer:

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

class Length {
public:
    int foot;
    int inch; //value in [0, 11]
};

Length multiply(Length len, int times);

int main() {
    Length len = {2, 7};

    Length total = multiply(len, 3);
    cout << total.foot << " " << total.inch << endl; //7 9

    return 0;
}

```

```
Length multiply(Length len, int times) {
    int total_inches = len.foot * 12 + len.inch;

    int product = total_inches * times;
    return {product / 12, product \% 12};
}
```

4 (10 points) Write codes of vector

Define a function called `all_identical`, for a vector `v` of ints, if `v` is empty, return false, otherwise, return true if and only if all elements are identical.

For example, given an empty vector, the return is false.

Given a vector of ints with elements `1, 2, 1`, the return is false.

Given a vector of ints with elements `1, 1`, the return is true.

Answer:

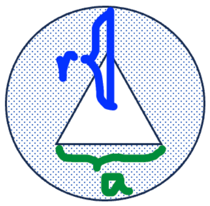
```
bool all_identical(vector<int> v) {
    if (v.size() == 0)
       return false;

    int i = 1;
    while (i < v.size() && v[i] == v[0])
        i++;

    return (i == v.size());
}
```

A complete code is shown as follows.

```
//File name: /Users/laptopuser/Documents/courses_macbook_pro/cs135/s25/final/v4/code/
    all_identical.cpp
#include <iostream>
#include <string>
#include <vector>
using namespace std;

bool all_identical(vector<int> v);

int main() {
    vector<int> v;
    cout << boolalpha << all_identical(v) << endl; //false

    vector<int> v2 = {1, 2, 1};
    cout << boolalpha << all_identical(v2) << endl; //false

    vector<int> v3 = {1, 1};
    cout << boolalpha << all_identical(v3) << endl; //true
    return 0;
}

bool all_identical(vector<int> v) {
    if (v.size() == 0)
       return false;

    int i = 1;
    while (i < v.size() && v[i] == v[0])
        i++;

    return (i == v.size());
}
```

5 (15 points) Define class.

1. Define a TriCir as the region between an equilateral triangle nested into a circle. The shapes are concentric (share the same center). It has two parameters:



   (a) edge of the eqilateral triangle **a**

   (b) radius of the circle **r**

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

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

   Answer:

```
TriCir::TriCir() {
    a = 1;
    r = 2;
}
```

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

   (a) If a is positive and r is at least $\frac{a}{\sqrt{3}}$, set data member **a** by given parameter a and set data member **r** by given parameter r.

(b) otherwise, set data members **a** to be 1 and **r** to be 2.

Answer:

```
TriCir::TriCir(double a, double r) {
    if (a > 0 && && r >= a / sqrt(3)) {
       this->a = a;
       this->r = r;
    }
    else {
        this->a = 1;
        this->r = 2;
    }
}
```

5. Define method **getArea**, return the value of $\pi r^2 - \frac{\sqrt{3}}{4}a^2$, where π is defined as `M_PI` in `cmath` library. Note that a and r are data members.

Answer:

```
double TriCir::getArea() const {
    return M_PI * r * r - sqrt(3) / 4 * a * a;
}
```

6. Define method **setEdge**, if given parameter a is positive and no larger than $\sqrt{3}\mathbf{r}$, where **r** is a data member, reset data member **a** by given parameter a.

Answer:

```
void TriCir::setEdge(double a) {
    if (a > 0 && a <= r * sqrt(3))
       this->a = a;
}
```

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

7. Create a TriCir object named **shape** from its non-default constructor with the edge of the triangle as 2 and the radius of the circle as 3.6.

Answer:

```
    TriCir shape(2, 3.6);
```

8. Reset the edge of shape to be 1.35.

Answer:

```
    shape.setEdge(1.35);
```

9. Find out and print the area of shape.

   Answer:

```
    cout << "area: " << shape.getArea() << endl;
```

Answer: A complete code is as follows.
code of TriCir.hpp

```
#ifndef TRI_CIR_H
#define TRI_CIR_H
class TriCir {
public:
    TriCir();
    TriCir(double a, double r);
    double getArea() const;
    double getPerimeter() const;
    void setRadius(double r);
    void setEdge(double a);

private:
    double r; //radius of the circle
    double a; //edge of the square
};
#endif
```

Code of TriCir.cpp

```
#include "TriCir.hpp"
#include <cmath>

TriCir::TriCir() {
    a = 1;
    r = 2;
}

TriCir::TriCir(double a, double r) {
    if (a > 0 && r >= a / sqrt(3)) {
       this->a = a;
       this->r = r;
    }
    else {
        this->a = 1;
        this->r = 2;
```

```cpp
    }
}

void TriCir::setRadius(double r) {
    if (r >= a / sqrt(3))
       this->r = r;
}

void TriCir::setEdge(double a) {
    if (a > 0 && a <= r * sqrt(3))
       this->a = a;
}

double TriCir::getArea() const {
    return M_PI * r * r - sqrt(3) / 4 * a * a;
}

double TriCir::getPerimeter() const {
    return 3 * a + 2 * M_PI * r;
}
```

code of TriCirTest.cpp

```cpp
#include <iostream>
#include <string>
#include "TriCir.hpp"
using namespace std;

//sample output:
//area: 39.9259
//perimeter: 26.6695
int main() {
    TriCir shape(2, 3.6);
    shape.setEdge(1.35);
    cout << "area: " << shape.getArea() << endl;
    cout << "perimeter: " << shape.getPerimeter() << endl;

    return 0;
}
```

6 (10 points) function on vectors

Define a function called `pickEqualElms`, given two **sorted** vectors of chars `v1` and `v2`, do the following:

Pick up the elements that exists in both `v1` and `v2`, also in **sorted** order. **Warning**: cannot use sort or find method from algorithm library. For simplicity, we assume that there is no redundant elements in each vector.

For example, if v1 is {'a', 'b', 'c'} and v2 is {'b', 'c'}, the returned vector has elements {'b', 'c'}. If v1 is {'b', 'c', 'e'} and v2 is {'a', 'f'}, the returned vector is empty.
Hint: how do we merge two sorted vectors to get a merged sorted vector?

Answer: function `pickEqualElms` is defined as follows.

```
vector<char> pickEqualElms(vector<char> v1, vector<char> v2) {
    int i = 0;
    int j = 0;
    vector<char> v3;
    while (i < v1.size() && j < v2.size()) {
        if (v1[i] == v2[j]) {
           v3.push_back(v1[i]);
           i++;
           j++;
        }
        else if (v1[i] < v2[j]) {
                i++;
        }
        else j++;
    }

    return v3;
}
```

A complete code is as follows.

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

vector<char> pickEqualElms(vector<char> v1, vector<char> v2);

int main() {
    vector<char> v1{'a', 'b', 'c'};
    vector<char> v2{'b', 'c'};

    vector<char> v3 = pickEqualElms(v1, v2);
    for (char elm: v3)
        cout << elm << " ";
    cout << endl;

    return 0;
}

vector<char> pickEqualElms(vector<char> v1, vector<char> v2) {
    int i = 0;
    int j = 0;
    vector<char> v3;
    while (i < v1.size() && j < v2.size()) {
        if (v1[i] == v2[j]) {
           v3.push_back(v1[i]);
           i++;
           j++;
        }
        else if (v1[i] < v2[j]) {
                i++;
        }
        else j++;
    }

    return v3;
}
```

7 (10 points) Define recursive function

Define a recursive function `isSumOdd`, given an array of int with size many elements, test whether the sum of the array is odd or not.

Hint: the sum of two odd integers or two even integers are even. The sum of an odd and an even integer is odd.

For example, for array with elements 1, 2, the return is true. For array with elements 1, 2, 3, the return is false.

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.

Answer: Code of function is as follows.

```
bool isSumOdd(int arr[], int size) {
    if (size == 1) {
       if (arr[0] % 2 != 0)
          return true;
       else return false;
    }

    //size > 1
    if ( isSumOdd(arr+1, size-1) ) { //arr[1..size) is odd
       if (arr[0] % 2 != 0)
          return false;
       else return true;
    }
    else {
          if (arr[0] % 2 != 0)
```

```
            return true;
         else return false;
    }
}
```

A complete code is as follows.

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

bool isSumOdd(int arr[], int size);

int main() {
    int arr[] = {2, 1, 3, 1};
    int size = sizeof(arr) / sizeof(arr[0]);

    cout << boolalpha << isSumOdd(arr, size) << endl;
    return 0;
}

bool isSumOdd(int arr[], int size) {
    if (size == 1) {
       if (arr[0] % 2 != 0)
          return true;
       else return false;
    }

    //size > 1
    if ( isSumOdd(arr+1, size-1) ) { //arr[1..size) is odd
       if (arr[0] % 2 != 0)
          return false;
       else return true;
    }
    else {
          if (arr[0] % 2 != 0)
             return true;
          else return false;
    }
}
```