

COP3331 Lab 5

Submission Instructions:

1. Create a folder named Lab5_LastName_FirstInitial (e.g. Lab5_Neal_T).
2. In your folder, place a **PDF** file containing your answers to questions with a \diamond .
3. Copy your directories containing your programs for questions with a \spadesuit into the folder; **these directories should only contain files needed to run your program, which may include one or more of the following file types: .cpp, .h., and .txt.** Do NOT include the full project (e.g., solution file). Test your program on CIRCE before submitting by compiling and running with g++. Your file containing main() should **always** be named main.cpp.
4. Ensure that all programs have block comments at the very beginning (starting at the first line) in the file containing main() with your name and the program's description. **The block comment's format should be identical to what's provided in Figure 2-1.**
5. Use single-line comments to describe your code's functionality as needed.
6. Do not submit anything for questions with a \clubsuit .
7. Zip the folder and submit it via Canvas.

\diamond = 5 points each, \spadesuit = 15 points each

1. \clubsuit Read *Chapter 11: How to work with STL algorithms.*
2. \clubsuit Read *Chapter 12: How to work with built-in arrays and C strings.*
3. \clubsuit Read *Chapter 13: How to work with exceptions.*
4. \diamond A function template
 - a. allows the function to accept different data types
 - b. allows the function to work with a varying number of arguments
 - c. allows the function to return different data types
 - d. all of the above
5. \diamond Given this vector:

```
vector<int> scores { 92, 87, 98, 76, 85 };
```

what is the value of the variable named i after the following code is executed?

```
auto max = max_element(scores.begin(), scores.end());  
auto i = max - scores.begin();
```

- a. 98
 - b. 2
 - c. 6
 - d. An iterator that points to the element with the largest value
6. \diamond Given this function:

```
int sum_nums(int total, pair<int, int> p) {
    return total + (p.first * p.second);
}
```

what is the value of the variable named sum after the following code is executed?

```
multimap<int, int> nums{ {2, 10}, {3, 20}, {3, 30}, {4, 40} };
int sum = accumulate(nums.begin(), nums.end(), 0, sum_nums);
```

- a. 100
- b. 200
- c. 330
- d. 1200

7. ◇ If you omit the size declarator when you define an array, you must
 - a. set the size before you use the array
 - b. use an initialization list so the compiler can infer the size
 - c. assign a value to each element of the array that you want to create
 - d. copy elements from another array
8. ◇ To copy an array, you should
 - a. use the assignment operator
 - b. store the second array in a reference variable and then use the assignment operator to copy to first array to the second array
 - c. use a loop to copy the elements from one array to the other
 - d. get a pointer to each element in the array and then dereference the pointer and assign the value to an element in the other array
9. ◇ Which of the following statements defines a C string?
 - a. `char message[] = {'W', 'e', 'l', 'c', 'o', 'm', 'e', '!'};`
 - b. `char message[] = {'W', 'e', 'l', 'c', 'o', 'm', 'e', '!', '\0'};`
 - c. `char message[] = 'Welcome!';`
 - d. `char message[] = 'Welcome!\0';`

10. ◇ What happens when the following code is executed?

```
void function2() {
    throw runtime_error("An error occurred in function2!");
}

void function1() {
    try {
        function2();
        throw runtime_error("An error occurred in function1!");
    }
    catch(const exception& e) {
        cout << e.what() << "\n\n";
    }
}

int main() {
    try {
        function1();
    }
    catch(const exception& e) {
        cout << e.what() << "\n\n";
    }
}
```

- a. An exception is thrown by function1() and caught by the main() function.
- b. An exception is thrown by function1() but isn't caught so the program crashes.
- c. An exception is thrown by function2() and caught by function1().
- d. An exception is thrown by function2() and caught by the main() function.

11. ♠ Binary Search (STL version)

Create a program that uses an STL algorithm to perform a binary search that checks whether a number is in a sequence of sorted numbers. Place your files in folder lab5-q11.

Console

```
Binary Search program

Enter 0 to exit

Random numbers:  [13, 16, 18, 29, 32, 71, 71, 77, 78, 90]

Enter a number from 1 to 100: 1
1 is NOT in random numbers.

Enter a number from 1 to 100: 32
32 is in random numbers.

Enter a number from 1 to 100: 100
100 is NOT in random numbers.

Enter a number from 1 to 100: 0
Bye!
```

Specifications

- The program should begin by generating a list of 10 random integers from 1 to 100 (both inclusive).
- The program should allow the user to enter a number from 1 to 100. Then, it should display whether that number is or isn't in the list of random numbers. Use data validation to ensure the number is within the right range.
- The program should use an STL algorithm to sort the numbers.
- The program should use an STL binary search algorithm to search the list of random numbers.

12. ♠ Project: Prime Number Checker

Create a program that checks whether a number is a prime number and displays its factors if it is not a prime number. Place your files in folder lab5-q12.

Console

```
Prime Number Checker

Please enter an integer between 1 and 5000: 5
5 is a prime number.

Try again? (y/n): y

Please enter an integer between 1 and 5000: 6
```

```
6 is NOT a prime number.  
It has 4 factors: 1 2 3 6  
  
Try again? (y/n): y  
  
Please enter an integer between 1 and 5000: 200  
200 is NOT a prime number.  
It has 12 factors: 1 2 4 5 8 10 20 25 40 50 100 200  
  
Try again? (y/n): n  
  
Bye!
```

Specifications

- A prime number is divisible by two factors (1 and itself). For example, 7 is a prime number because it is only divisible by 1 and 7.
- Assume that the user will enter a valid integer.
- If the user enters an integer that's not between 1 and 5000, the program should display an error message.
- If the number is a prime number, the program should display a message.
- If the number is not a prime number, the program should display a message. Then, it should display the number of factors for the number and a list of those factors.
- Store the factors for each number in a vector.
- Use functions to organize the code for this program.