

CSCE 240 – Exam Two

Due: 11:59am on Friday, October 28. Late exam submissions will not be accepted.

This is an exam. As you work on these problems, you may use your textbook, class notes, and the recorded lectures. You may ask your instructor clarifying questions. You are not to discuss the problems with other students or seek help from other individuals. All work submitted must be your own. All code submitted will be examined for plagiarism and violations will be reported to the office of Student Conduct and Academic Integrity.

Test all of your code on a Linux lab computer. All source files submitted must compile and run on a Linux lab computer of the instructor's choice. Submissions that do not compile on the Linux workstation will receive no compilation or execution/correctness points.

Problem 1

Deliverable: *problem1.cc*

Purpose: Write a *Remove* function that takes an array of integers, the size of the array, and an integer to remove as arguments. The function should remove all instances of the third argument from the array. The remaining elements should be shifted into the vacated positions, with a zero added to the end of the array for each the removed elements.

For example, if the original array is

```
int my_array[7] = {1, 4, 2, 1, 3, 6, 1};
```

and we make the function call

```
Remove(my_array, 7, 1);
```

my_array should hold {4, 2, 3, 6, 0, 0, 0} after the function call.

Specifications:

- The files for this problem are in the attached *problem1.zip*
- Implement your function in *problem1.cc* this is the only file to be submitted to the assignment for this problem.
- Do not change any code in the attached *problem1.h*
- Your function should compile and link with the command

```
g++ -std=c++17 -I . problem1.cc testproblem1.cc -o testproblem1
```

to run the initial tests provided in *testproblem1.cc*

Initial Testing:

Initial tests have been provided in *testproblem1.cc*. You are encouraged to implement more rigorous tests. Your program will be graded with a modified version of *testproblem1.cc*

Points:

style: 0.5 point

documentation: 0.5 point

clean compilation: 0.5 point

execution / correctness: 1.5 points

Problem 2

Deliverables: *volume.h* and *volume.cc*

Purpose: Create a *Volume* class that has a private data member for the volume quantity (a double) and a private data member for volume unit of measure (a string). You will include a constructor, accessor and mutator functions for the private data members, a public member function to convert between units of measure, and a public member function to add a *Volume* object to the current object. Read the comments in the attached *volume.h* and *volume.cc* files for more details.

Specifications:

- The files for this problem are in the attached *problem2.zip*
- Add all of your code to the attached *volume.h* and *volume.cc* files, and attach your updated *volume.h* and *volume.cc* files to the assignment. These are the only two files to be submitted for this problem.
- Your code should compile and link with the command

```
g++ -std=c++17 -I . volume.cc problem2.cc -o problem2
```

to run the initial tests provided in *problem2.cc*
- Ensure that your class will compile, link and run with the initial tests provided in order to earn compilation points and to be eligible for correctness points.

Initial Testing:

problem2.cc has been included with some minimal initial tests for the constructor, convert, and overloaded stream insertion operator. You are encouraged to include more rigorous tests before submitting your class. This problem will be graded with a modified version of *problem2.cc*

Points:

style: 0.5 point

documentation: 0.5 point

clean compilation: 1 point

constructor passes instructor tests: 1 point

accessor / mutator functions pass instructor tests: 1 point

convert passes instructor tests: 1 point

add function passes instructor tests: 1 point

Problem 3

Deliverable: *problem3.cc*

Purpose: Write functions that use the *Book* class defined and implemented in the attached *bookproblem3.h* and *bookproblem3.cc* files. This *Book* class is similar to the one created in class.

Functions

- Write a function named *WhichChapter* that takes a pointer to a *Book* object and a string as arguments. The function will return an integer corresponding to which chapter in the *Book* has the string argument as its name. If no chapter matches the string argument, the function should return -1. See *problem3.h* for additional details.

- Write a function named *RemoveChapter* that takes a pointer to a Book object and a string as arguments. If the string is the name of one of the chapters in the Book, that chapter should be removed from the object's chapter list. See *problem3.h* for additional details.

Specifications:

- The files for this problem are in the attached *problem3.zip*
- No changes are to be made in the attached *bookproblem3.h*, *bookproblem3.cc*, and *problem3.h* files
- Implement both functions in *problem3.cc* and attach your revised *problem3.cc* file to the assignment. This is the only file to be submitted for this problem.
- Your code should compile and link with the command
`g++ -std=c++17 -I . bookproblem3.cc testproblem3.cc problem3.cc -o problem3`
to run the initial tests provided in *testproblem3.cc*

Initial Testing:

Minimal initial tests have been included in *testproblem3.cc*. You are encouraged to create more rigorous tests. A revised version of *testproblem3.cc* will be used to test your functions for correctness.

Points:

style: 0.5 point

documentation: 0.5 point

clean compilation of *problem3.cc*: 1 point

execution / correctness of *WhichChapter* function: 2 points

execution / correctness of *RemoveChapter* function: 2 points