

CSE 240 Spring 2021 Homework 8:

C++ Class and Input/Output (50 points)

Due Saturday, March 20, 2020 at 11:59PM, plus a 24-Hour grace period

Introduction

The aim of this assignment is to make sure that you understand and are familiar with the concepts covered in the lectures and input/output. By the end of the assignment, you should have understood

- Class, data members and function members
- Classes and header files
- Memory management and garbage collection
- C++ input and output

Reading: Textbook Chapter 3, Sections 3.1, 3.2, and 3.3 on C++ examples, class definition and memory management.

Preparation: Complete the multiple choice questions in the textbook exercise section. The answer keys can be found in the course Web site. These exercises can help you prepare for your weekly quiz and the exam. You are encouraged to read the other exercise questions and make sure you understand these questions in the textbook exercise section, which can help you better understand what materials are expected to understand after the lectures and homework on each chapter.

You are expected to do the majority of the assignment outside of class meetings. Should you need assistance, or have questions about the assignment, please contact the instructor or the TA during their office hours.

You are encouraged to ask and answer questions on the course discussion board. However, **do not share your answers and code** in the course discussion board.

Programming Assignment (50 points)

1. You are given a partially completed project containing:

1 header file:

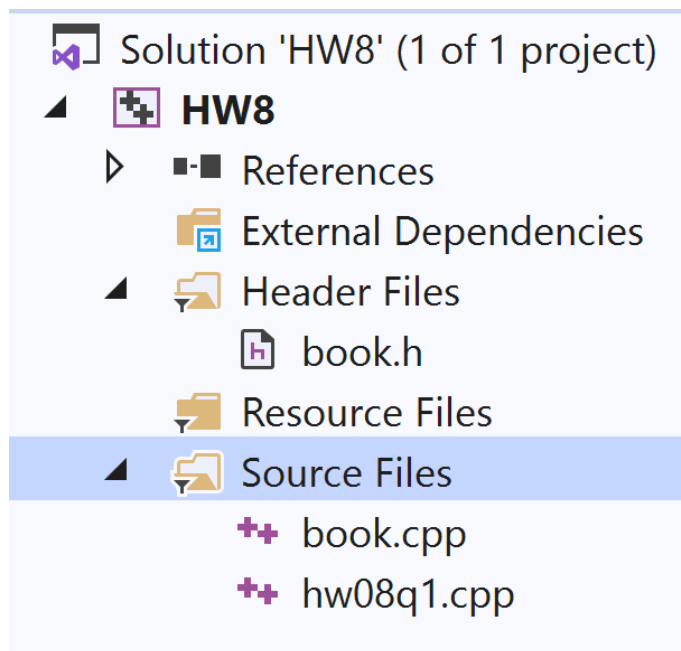
book.h (contains the class 'Book')

2 C++ files:

book.cpp (contains the class function definitions)

hw08q1.cpp (contains the program to work on array of 'Book' objects)

If you use VS, then create an empty C++ project. Then add the header file and CPP files, respectively, into your project as shown in the following screenshot:



If you use ASU General g++, then simply put the .h and .cpp files in one folder and compile with this command:

```
g++ book.cpp hw08q1.cpp -o out
```

Execute with:

```
./out
```

Follow the instructions given in the comments of the hw08q1.cpp and book.cpp files to complete the missing parts of the project so that the program executes properly. You should first complete book.cpp and then go on to hw08q1.cpp.

NOTE: In the homework, list means an array (not a linked list).

This is a menu-driven program that uses the following options:

- Add a new book. The book details (title, ID, aisle , bookType) are given as function arguments. You should add the book only if the book is not already present in the array and there is space in the array to add a new book.
- Display the all book's information. See expected output below.
- Display only the books whose ID falls within a range (in descending order)
- Display the book with the longest title among the books whose bookType contain a specific substring. Ask the user for a substring. This function is used to demonstrate garbage collection.

You should start completing the program beginning from Q1. Question numbers are given on line 29 in hw08q1.cpp. (Q1 - Q2 in book.cpp and Q3 - Q6 in hw08q1.cpp)

Expected outputs:

addBook():

- This function adds a new book with the details given in function arguments.
- Add the book in 'b' (array of objects) only if there is remaining capacity in the array and if the book does not already exist in the list (title or ID)
- This function returns 1 if the book is added successfully, else it returns 0 for the cases mentioned above.
- Assume user enters ID and aisle in 0 - any positive integer range.

```
CSE240 HW8
Please select an action:
    a: add a new book
    d: display book list
    s: sort the books in descending order based on ID (within a range)
    n: display the book with the longest title among the books whose bookType contain a specific substring
    q: quit
a
Please enter book title: The Alchemist
Please enter book ID: 1
Please enter aisle number: 101
Please enter bookType: Fiction

Book successfully added to the list!
```

displayBooks():

- This function displays the list of books.
- Parse the object array 'b' and display the details of all books in the array. See expected output given in question file.
- You can call the class function 'displayBook()' here. Note that these are two different functions.

```

CSE240 HW8
Please select an action:
    a: add a new book
    d: display book list
    s: sort the books in descending order based on ID (within a range)
    n: display the book with the longest title among the books whose bookType contain a specific substring
    q: quit

d
Title: The Alchemist
ID: 1
Aisle: 101
Book Type: Fiction

Title: A Promised Land
ID: 2
Aisle: 102
Book Type: Nonfiction

Title: Sapiens
ID: 3
Aisle: 103
Book Type: Nonfiction

```

sort():

- This function sorts the books in descending order of ID, and then display the books within a given range.
- You need to get lower bound and higher bound from user after printing a prompt.

```

CSE240 HW8
Please select an action:
    a: add a new book
    d: display book list
    s: sort the books in descending order based on ID (within a range)
    n: display the book with the longest title among the books whose bookType contain a specific substring
    q: quit

s
Please enter the lower bound of ID: 0
Please enter the higher bound of ID: 2
Title: A Promised Land
ID: 2
Aisle: 102
Book Type: Nonfiction

Title: The Alchemist
ID: 1
Aisle: 101
Book Type: Fiction

```

bookTypeWithSpecificString():

- This function displays a book with the longest title among the books whose bookType contain a specific substring.
- You should find the book as follows:
 1. By traversing all books, you should find the books whose bookType include a specific substring.
 2. After step 1, you should find the book whose title is the longest. You may use 'titleLength' and 'index' variable.
 3. After step 2, copy the details of the book to 'bookWithLengthyTitle' object created using 'new' and display the book's details using 'bookWithLengthyTitle' object.

4. Finally delete the ' bookWithLengthyTitle ' object.

```
CSE240 HW8
Please select an action:
  a: add a new book
  d: display book list
  s: sort the books in descending order based on ID (within a range)
  n: display the book with the longest title among the books whose bookType contain a specific substring
  q: quit
n
Enter the substring of book's bookType: Non
Title: A Promised Land
ID: 2
Aisle: 102
Book Type: Nonfiction
```

What to Submit?

You are required to submit your solution in a compressed format (.zip). Make sure your compressed file is label correctly - lastname_firstname8.zip. (All lowercase, do not put anything else in the name like "hw8".)

The compressed file MUST contain the following:

hw08q1.cpp	(completed code)
book.cpp	(completed code)
book.h	(unedited, no change needed)

No other files should be in the compressed folder.

If multiple submissions are made, the most recent submission will be graded. even if the assignment is submitted late.

Submission preparation notice: The assignment may consist of multiple files. You must copy these files into a single folder for canvas submission. To make sure that you have all the files included in the zip file and they work after unzip operation, you must test them before submission. You must also download your own submission from the canvas. Unzip the file on a different machine and test your assignment and see if you can open and test the files in a different location, because the TA will test your application on a different machine. If you submitted an empty project folder, an incomplete project folder, or a wrong folder, you cannot resubmit after the submission linked is closed! We grade only what you submitted in the canvas. We cannot grade the assignment on your computer or any other storage, even if the modification date indicated that the files were created before the submission due dates. The canvas submission may take a few minutes. Be patient and wait for it to complete.

Where to Submit?

All submissions must be electronically submitted to the respected homework link in the course web page where you downloaded the assignment.

Late submission deduction policy

- No penalty for late submissions that are received within 24 hours after the deadline;
- 10% grade deduction for every day it is late after the grace period;
- No late submission after Tuesday at 11:59PM.

Grading Rubrics

Each sub-question (programming tasks) has been assigned certain points. We will grade your programs following these steps:

- (1) Compile the code. If it does not compile, 50% of the points given for the code under compilation will be deducted. Then, we will read the code and give points between 50% and 0, as shown in right part of the rubric table.
- (2) If the code passes the compilation, we will execute and test the code using test cases. We will assign points based on the left part of the rubric table.
- (3) In both cases (passing compilation and failed compilation), we will read your program and give points based on the points allocated to each sub-question, the readability of your code (organization of the code and comments), logic, inclusion of the required functions, and correctness of the implementations of each function.
- (4) Please notice that we will not debug your program to figure out how big or how small the error is. You may lose 50% of your points for a small error such missing a comma or a space!
- (5) We will apply the following rubrics to **each sub-question** listed in the assignment. Assume that points assigned to a sub-question is pts:

Major	Code passed compilation				Code failed compilation		
Points	Pts * 100%	Pts * 90%	Pts * 80%	Pts * 60%-70%	Pts * 40%-50%	Pts * 10%-30%	0
Each Sub-question	Meeting all requirements, well commented, and working correctly in all test cases	Working correctly in all test cases. Comments not provided to explain what each part of code does.	Working with minor problem, such as not writing comments, code not working in certain uncommon boundary conditions.	Working in most test cases, but with major problem, such as the code fail a common test case	Failed compilation or not working correctly but showing serious effort in addressing the problem.	Failed compilation, showing some effort, but the code does not implement the required work.	No attempt

Please read the FAQ file in the Course Information folder:

Q: For some reason, my assignment submission did not go through, but I thought it went through. I can show you on my local disk or in my Dropbox that I completed the assignment before the due date. Can my assignment be graded?

A: You should always download your own submission from the blackboard after submission and test if the submission contains all the required files. We will grade the assignment submitted to Canvas **only**. We cannot grade the assignment sent from email or stored in any other places, regardless its last-modified-time. If you submitted your assignment into the blackboard, it cannot be downloaded from the instructor side, but it can download from your side, we can download from your blackboard and grade the assignment. Please meet the instructor or TA in this case.