**University of Michigan – Dearborn**

**CIS 200 – Computer Science II**

**Lab# 1**

**Professor Robert Mann**

Demetrius Johnson

[meech@umich.edu](mailto:meech@umich.edu)

September 15, 2020

**Table Content**

Contents

[Question 1 3](#_Toc51069484)

[Source Code –see JohnsonDemetriusLab01-MAIN.cpp, student.cpp, and student.h 3](#_Toc51069485)

[Screenshots 3](#_Toc51069486)

# Question 1

// LAB 1 - CIS 200

Question 1

-------------------------

Program:

Use separate source files and header files to create a class to represent students.

a. A student object has attributes of major (string) and credit hours taken (integer).

b. Member functions (methods) are as follows (no additional ones may be created):

i. Using a constructor initializer list, create a constructor with default parameter values of “General Studies” and -1, and that displays “\n\tConstructor called for major <major> and hours <hours>\n”, replacing <major> and <hours> with attribute values.

ii. individual const get methods (in-line definition) for each attribute

iii. a const get method (NOT in-line) that uses pass-by-reference for both parameters and has return-type void

iv. one set method (NOT in-line) that sets both attributes

v. PrintMe method (NOT in-line) that returns string “I’m a <major> major and have completed <x> credit hours.”, replacing <major> and <x> with the appropriate attribute values

vi. Destructor (in-line) that displays “\n\tDestructor called for <major> major.\n”, replacing <major> with the appropriate attribute value

Write a complete program that uses the student class you created and tests its functionality

b. Declare a single student object and during declaration initialize it to have major “Astrophysics”; do not provide a value for number of credit hours

c. Call the PrintMe method of the single student object and display to the screen/console

d. Ask the user for the name of a file to which output should be written

e. Declare a vector of 5 objects of student class

i. Ask the user for 5 majors and 5 credit hours and assign them to each student in the vector. The credit hours must be verified as non-negative or else get another number. This must be tested in your output.

ii. Open the file and write to it using the value returned by PrintMe for every student in your vector

f. Explicitly close the output file

g. Any other tests you deem appropriate to prove your program works perfectly

In order to properly capture the destructor messages, put a breakpoint on the return statement in main. When the program pauses at that line, press F10 once, capture the output screen as a snippet, then press F5 to finish execution.

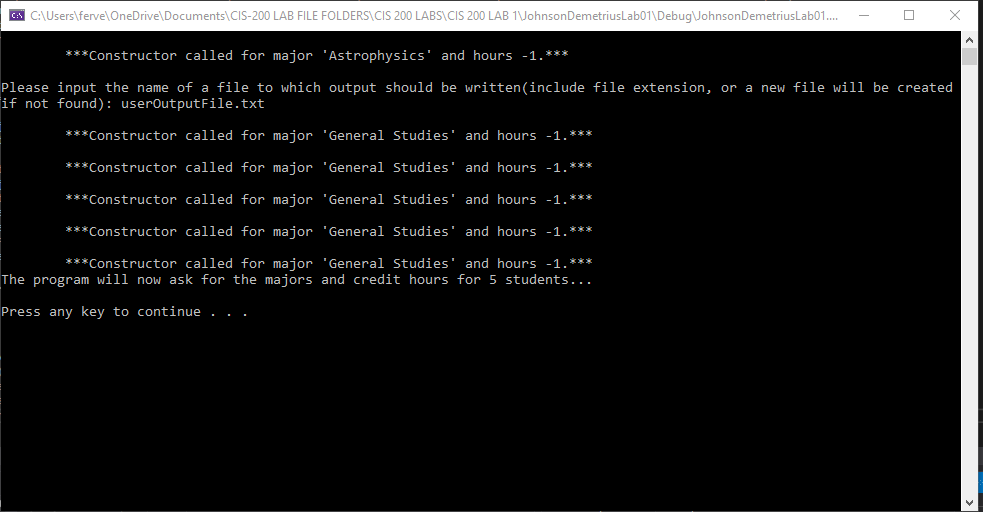
\*/

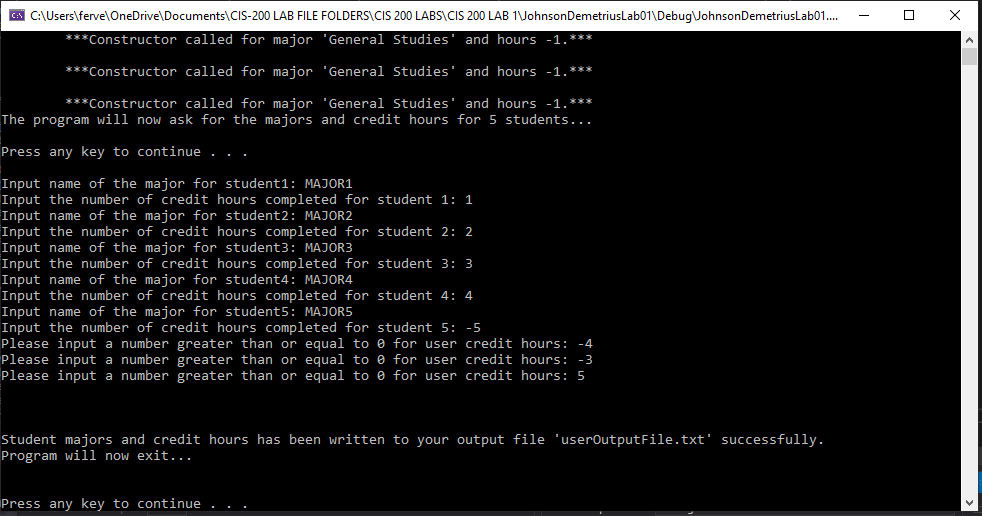
## Source Code –see JohnsonDemetriusLab01-MAIN.cpp, student.cpp, and student.h

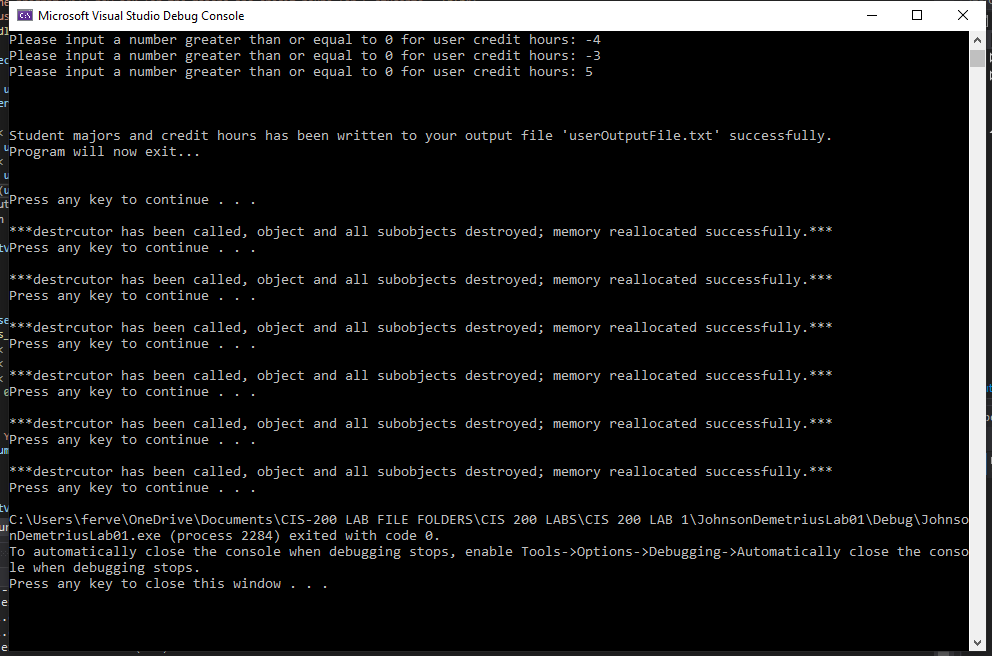
## Screenshots

Test Table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Expected Output | Actual Output | Test Pass / Fail |
| 1 | valid | Test to see if constructors are called | Created objects | \*\*constructor called\*\*\* | See screenshot | pass |
| 2 | valid | Test to see if input works and if -1 input value for credits results and a loop until it is corrected | Input MAJOR# and # to test major and credit hrs input; inserted negative numbers as well | Move to next student input unless negative number input for credits | See screenshot | pass |
| 3 | valid | Test to see if destructors are called |  | \*\*destroying objects\*\*\* | See screenshot | pass |
| 4 | valid | Test to see if program successfully output data into selected user output file |  | User vector input for type student | See screenshot | pass |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

TEST 1

TEST 2

TEST 3

TEST 4

