

CSE-165-Lab 8

100 Points

Write a separate .cpp file for each of the following tasks. For each question try to come up with various test cases to test your code. You may be evaluated based on test cases provided during demo.

1. Abstract Datatypes (5 Points)

Download the file ADT.h. It contains an abstract data type called ADT. Create a type named Derived which inherits from ADT. Save your Derived class in a file named Derived.h. To get an idea about the implementation of the methods in Derived, study the file ADTs.cpp, which has an ADT pointer pointing to an instance of Derived.

Sample output from ADTs.cpp

I did something

I did something else

2. Integer Sorting (10 points)

Your task is to write a class called Data, stored in a file named Data.h. Your class should be able to store a collection (vector) of integers. In addition to the appropriate constructors, your class should also have the following methods.

`void add(int number);` // Adds a number to the data set.

`void print();` // Prints out the entire data set on a single line, separated by space.

`void sort();` // Sorts the data set in ascending order. You may implement any sorting algorithm here, for example, max sort, bubble sort, insertion sort, merge sort, quick sort, etc... There is no need to try to implement the most efficient one, any one will do.

Make sure your class works as expected with the file intSort.cpp.

Sample output from intSort.cpp

4 5 2 3 1

1 2 3 4 5

CSE-165-Lab 8

100 Points

3.Sortable Objects (10 Points)

Study the file Sortable.h. It contains an abstract data type called Sortable, which will act as a base class for any object a collection of which can be sorted. It has two virtual methods, compare and print. The compare method tells the object how to compare itself with another object. It returns true if the object it is being called from is smaller than the other object. The print method allows the object to display itself.

Modify your Data class so that it operates on Sortable objects instead of int objects. Recall from the textbook that we cannot allocate abstract data types. Therefore, in order to accomplish the task in this exercise, you will need to use pointers.

Also, you are not provided with any .cpp files in order to test your code. Testing your code will involve creating a class that extends Sortable and pushing some objects of that class into your Data object and trying to sort and print them. This is the goal of the next exercise.

4.Sortable Objects II (10 Points)

Reuse your files Data.h and Sortable.h from the previous exercise (Problem 3 above).

Now create a class called Circle which inherits from Sortable. A Circle object will only have a radius. Circle objects are compared according to radius. Circle objects should be printed one per line as follows, Circle with radius: <x>, where <x> is the radius of the circle.

You can test your code with the file sortingCircles.cpp as well as the Sortable.h file and the Data.h file that you submitted for the last question.

5. Sortable Objects III(10 Points)

This question is almost the same as the last one. You can reuse the files Data.h and Sortable.h. This time we will be storing objects of the Participant class. Each Participant has a name, an age and a score.

CSE-165-Lab 8

100 Points

These are string, int, and double, respectively. To print a Participant simply print the name, age, and score separated by a tab character.

Participant objects should be ordered by score from highest to lowest. If two Participants have the same score then the younger of them takes priority and if they have the same score and the same age, simply order them alphabetically. You can test your code with the file `sortingParticipants.cpp`.

6. (55 Points) In this lab, we will learn how to create OpenGL window using Qt framework.

Task 1: 2D Graphics:

Download the zip folder "QT_OPENGL_Example1.zip" and extract the content. Next open Qt Creator and load QT_OPENGL_Example1.pro. Study `mainwindow.h`, `mainwindow.cpp` and `main.cpp`. Try to change code and see the changes by running the project.

Hint: To understand code, watch the tutorial below

- a. <https://youtu.be/a2CpOQsgB84>
- b. <https://youtu.be/6eu8oMULqaU>
- c. <https://youtu.be/ITaICqf4Ifc>

Task 2: 3D Graphics

Download the zip folder "QT_OPENGL_Example2.zip" and Extract the content. Next open Qt Creator and load the QT_OPENGL_Example2.pro. Study `mainwindow.h`, `mainwindow.cpp` and `main.cpp`. Try to change code and see the changes by running the project.

Hint: To understand code, watch the tutorial below:

<https://youtu.be/mENP56CmeVw>

Task 3: 3D Graphics

Your task is to use "QT_OPENGL_Example2.zip" and draw a Tetrahedron. You need to create an animation such that the tetrahedron rotate around z axis and along x-y plane. Your project should be named as `Tetrahedron_Animation`. You need to submit the entire project folder as zip file.

CSE-165-Lab 8

100 Points

As part of your demo, you will be asked questions about Task1, Task2 and Task3 code. So, prepare yourself when you demo.

Instruction for submission:

1. Create folder for each solution and Zip all your folder together
2. Submit the zip folder to catcourse