

CS 215 Program Design, Abstraction, and Problem Solving

Programming Project #3

INTRODUCTION

The goal of this programming project is to enable the student to practice designing a program that solves a problem using a class and a linked-list and developing a C++ program to implement the solution.

PROJECT TASKS

1. Read the problem definition below and then analyze it before designing a solution. You will produce a document (external documentation) of this definition, analysis, and design.
1. Write a C++ program that implements your solution. The program must match your design. Readability of the program will be graded based on;
 - a variable naming,
 - b indentation,
 - c commenting (not too little and not too much),
 - d spacing,
 - e consistency
 - f styling in general including choice of functions.
 - g **Ternary** – Ternary operation are not permitted. Ternary syntax: (expression 1) ? expression 2 : expression 3;
2. After you finish writing and Testing your program, you will need submit the program for grading.

GOALS (or why are we doing this project)

The following are the things to be learned or show you have learned by this project.

- pointers
- How Linked list work.
- Working with a class containing only data and may not need a *.cpp file.

PROBLEM

Introduction

Write a program that simulates a customer service line at an Apple store. The line is a linked-list of pointers. Customers arrive at time specified by month/day/year plus hour/minute. The customers are served on a first come first serve basis. Here is the rest of the specification.

Your solution

Write a C++ program that has the following and does the following:

- Write a **class** that will represent the template of a node in the linked list. This class must be in its own file(s). The members of the class should be:

- a) sequenceNumber: type unsigned int
 - b) name: type string
 - c) month: integer
 - d) day: integer
 - e) year: integer
 - f) hour: integer
 - g) minute: integer
 - h) serviceRequired: type string
 - i) next: pointer to a node
- Write a **class** whose data members are:
 - i a head pointer to a node (The head will represent the start of a linked list).
 - ii a pointer to the last node.
 - iii **DO NOT USE C++11 STL containers (list, queue, stack, ...)**

The class's definition (in a h or hpp file) and implementation (in a cpp file) must be in separate files.

The member functions are specified below (define functions outside class).
- (b) **Constructor:** initialize all class instance variables.
 - (c) **InsertNode:** When a customer has just arrived, allocate memory for a new node (using the struct) and assign details received in parameters to the new node. The first node will be given sequence number 1, and subsequent nodes will have sequence numbers in increments of 1. Make the new node the new head (a FILO queue). The month, day, year, hour, and minute should be set to the current time and sequenceNumber is assigned. The parameters should be serviceRequired and name.
 - (d) **DeleteNode:** A customer may leave before being served. This method should receive the name of the customer name as a parameter Delete the node by putting its next pointer into the previous node next pointer. It should return true if the customer was found and deleted else false if the customer was not found.
 - (e) **SearchNode:** This function receives the customer name as a parameter. Recursively process the list to find the customer. Return a pointer to a copy of the current node when found or null when end of list is reached.
 - (f) **ServeCustomer:** Delete the customer at the tail-end of the list. Display a message saying the customer has been served.
 - (g) **ListAll:** recursively go through the linked list from beginning to end displaying line:

N Customer-name MM/DD/YYYY HH:MM XXXXXXXX

where N is the sequence number, Customer-name is the name of the customer,

MM/DD/YYYY is the date from the date members of the class, HH:MM is the time from the time data members, and XXXXXX is the purpose of the visit from the service_required data member.

Return the number of nodes in the linked list.

(h) **Destructor:** delete all the nodes.

- **The main() function:** Write a main function that instantiates an object using the linked list class. The main function must be in a separate file from the classes above. The main function should have a *do ... while ...* loop that runs until quit is selected and offers the user the following menu:
 - 1) New customer arrival
 - 2) Serve customer
 - 3) Customer left – remove from waiting list
 - 4) List all customers waiting for service
 - 5) Quit

Next in the loop, an option made by the user should be input. Depending on the option selected by the user, one of the operations listed in the bulleted list below should be performed. You may use a *switch* statement to handle this selection.

- **New customer arrival:** The user inputs the name, service request. Then insertNode is called with name and service request as parameters. The member function of the class creates and inserts a node while setting the date time fields to the current date and time and the sequence number to its correct value.
- **Serve customer:** when this option is selected, call the ServeCustomer member function of the class.
- **Customer left:** when this option is selected, input the name of the customer who left. This should be passed to the DeleteNode function of the class and the node should be deleted when found. The function should return the status of the delete. If the customer was found, display a message stating that the customer was not served and is removed otherwise state that the customer was probably already served anyway.
- **List all:** First display the following table headings:

APPLE CUSTOMER SERVICE WAITING LIST

SEQUENCE	NAME	ARRIVAL DATE/TIME	PURPOSE OF VISIT
----------	------	-------------------	------------------

Next, call the ListAll member function of the object putting it on the right of an assignment to an integer to receive the returned number of nodes in the list.

Display a message:

There are NN customers waiting to be served.

Where NN is an integer.

EXTERNAL DOCUMENTATION

Submit the external documentation in Word or PDF. The structure of your document is four clear sections each with a subheading as follows:

PROBLEM DEFINITION

This is a summary of the PROBLEM section above. Do not copy the text that I have written. Rather, phrase the problem in a few English sentences.

DESIGN

Give the algorithm in pseudocode form of the step by step statements to carry out the required programming task in structured English phrases **for your main() function only**. Remember that pseudocode should not be in C++ or any programming language. Pseudocode should not be in essay form. Rather, it should be short phrases with indentation with a bias towards what the final program will look like.

IMPLEMENTATION

Write in essay form, a description of your implementation. Implementation details include the following:

- Platform and programming environment used. (Note that platform means CPU/Operating System combo. Example of a platform is Dell Computer with Intel 8-Core 2.8GHz processor running Windows 10.)
- Give the name of the development environment that you used such as Visual Studio or GNU pico/g++ on grace.bluegrass.kctcs.edu.
- How did you test your program?
- Did you encounter any unusual situations when running the program? Example of unusual encounters include the program crashing when a letter is entered where a number is required.

SUBMISSION

- a) Upload to Blackboard a copy of your external documentation in Word or PDF.
- b) Upload to Blackboard a copy of your source program (the file(s) with a .cpp and *.h/*.hpp extension).

MAXIMUM POSSIBLE SCORE

This program will be graded out of 75 points distributed as follows:

<u>ITEM</u>	<u>MAX. POINTS</u>
External documentation	15
Style: comments, meaningful names, indentation, and readability in general	10
Program written to specification	25
Program works correctly	25

Note: Programs that do not implement the Linked list as a class will receive a zero.

Note: Class must be in their own file(s), the main function must be in a file separate from any class's files otherwise the program will receive a score of 0. See the videos from Project #2 for more help.