

Animation Project in C++

Due Time: 23.59, Sat 12 Oct. 2019

Earnings: 8% of your final grade

NOTE: Plan to finish a few days early to avoid last minute hardware/software holdups for which no allowance is given.

NOTE: The code in this assignment must be your own work. It must not be code taken from another student or written for you by someone else, even if you give a reference to the person you got it from (attribution); if it is not entirely your own work it will be treated as plagiarism and given a fail mark, or less.

Purpose: This is assignment #0 converted to C++, but now using classes. You are to write the code in Visual Studio 2019 for a simple C++ language console application that holds the data of an animation application (there is no actual animation graphics in the assignment) using a linked list in dynamic memory.

The start of the code is shown on the next page; it is also on Brightspace in a text file that you can copy and paste. Because I will use this code to mark the assignment you must not modify it (not a single character changed): no code added or removed, no new global variables, no new functions, no macros, no defines and no statics; leave the file contents exactly as you are given them. Your task is to implement, in Frame.cpp and Animation.cpp using C++, only the member functions of the Frame and Animation classes (as defined in the header files Frame.h and Animation.h. You submit only Frame.cpp and Animation.cpp.

The Animation is a series of Frame objects held in a single linked list (forward list) in dynamic (heap) memory. This memory model will be adapted to a C++ template in a later assignment.

When the application is running you can:

- add a new Frame to start of the Animation Frame list
- delete the last Frame in the list
- edit a selected Frame to change the frame name
- report the Animation to show the list of Frame details
- quit.

An example of the output of the running application is given at the end. Yours must work identically and produce identical output.

Note the following:

- the files you submit must be named Frame.cpp and Animation.cpp,
- dynamic memory management is done with new and delete,
- input and output is done with cin and cout
- there is never any excess dynamic memory allocated - only allocate exactly what is needed for each string entered by the user
- you can only use functions like strlen() and strcpy() etc. from the standard C library to handle strings of null terminated char; do not use the string container class,
- when the application terminates it releases all dynamically allocated memory so it does not have a resource (memory) leak (or you lose 30%).

See the Marking Sheet for how you can lose marks, but you will lose at least 60% if:

1. you change the supplied code in any way at all - no code added or removed, no macros, no defines, no statics and no additional global functions or variables; leave the file contents exactly as you are given them,
2. it fails to build in Visual Studio 2019
3. It crashes in normal operation (such as reporting an empty list etc.),
4. it doesn't produce the example output.

CST 8219 – F19 - Assignment #1

What to Do: Make an empty project in Visual Studio 2019 and add the files Frame.h, Animation.h and ass1.cpp that are supplied (they are in a text file on Brightspace that you can copy and paste). Add your new source code files Frame.cpp and Animation.cpp to the project and write your code in them. Then on Brightspace in the Assignment Submission folder submit a zip file (**not** RAR, not 9zip, not 7 zip) containing your Frame.cpp and Animation.cpp. The name of the zip file **must** contain your name as a prefix so that I can identify it, for example using my name the file would be tyleraAss1CST8219.zip. It is also vital that you include the Information (as specified in the Submission Standard) as a file headers in your source files so they can be identified as yours. Use comment lines in the file to include the header.

Before you submit the code,

- check that it builds and executes in Visual Studio 2019 as you expect - if it doesn't build for me, for whatever reason, you get a deduction of at least 60%.
- **make sure you have submitted the correct file – if I cannot build it because the file is wrong or missing from the zip, even if it's an honest mistake, you get 0 – no compromises.**

There is a late penalty of 25% per day. Don't send me the file as an email attachment – it will get 0.

Code you must use (also in a text file on Brightspace you can copy and paste). Don't change it.

<pre>//Frame.h #pragma once class Frame { char* frameName; Frame* pNext; public: Frame(); ~Frame(); char*& GetFrameName() { return frameName; } Frame*& GetpNext() { return pNext; } };</pre>	<pre>//Animation.h #pragma once class Animation { char* animationName; Frame* frames; public: Animation(); ~Animation(); void InsertFrame(); void EditFrame(); void DeleteFrame(); void ReportAnimation(); };</pre>
<pre>// ass1.cpp #define _CRT_SECURE_NO_WARNINGS #define _CRTDBG_MAP_ALLOC // need this to get the line identification //_CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF _CRTDBG_LEAK_CHECK_DF); // in main, after local declarations //NB must be in debug build #include <crtdbg.h> #include <iostream> using namespace std; #include "Frame.h" #include "Animation.h" int main(void) { char response; bool RUNNING = true; Animation A; _CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF _CRTDBG_LEAK_CHECK_DF); while (RUNNING) { cout<<"MENU\n 1. Insert a Frame at the front\n 2. Delete last Frame\n 3. Edit a Frame\n 4. Report the Animation\n 5. Quit"<<endl; cin>>response; switch (response) { case '1':A.InsertFrame(); break; case '2':A.DeleteFrame(); break; case '3':A.EditFrame(); break; case '4':A.ReportAnimation(); break; case '5':RUNNING = false; break; default:cout<<"Please enter a valid option"<<endl; } } return 0; }</pre>	

Example Output:

```
Please enter the Animation name: Animation1
MENU
1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit
1
Insert a Frame in the Animation
Please enter the Frame frameName: Frame1
```

CST 8219 – F19 - Assignment #1

```
MENU
1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit
1
Insert a Frame in the Animation
Please enter the Frame frameName: Frame2
MENU
1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit
1
Insert a Frame in the Animation
Please enter the Frame frameName: Frame3
MENU
1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit
4
Animation name is Animation1
Report the Animation
Frame #0 file name = Frame3
Frame #1 file name = Frame2
Frame #2 file name = Frame1

MENU
1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit
2
MENU
1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit
4
Animation name is Animation1
Report the Animation
Frame #0 file name = Frame3
Frame #1 file name = Frame2

MENU
1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit
3
Edit a Frame in the Animation
There are 2 Frame(s) in the list. Please specify the index (<=1) to edit at : 0
The name of this Frame is Frame3. What do you wish to replace it with?
Frame1
MENU
1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit
4
Animation name is Animation1
Report the Animation
Frame #0 file name = Frame1
Frame #1 file name = Frame2

MENU
1. Insert a Frame at the front
2. Delete last Frame
3. Edit a Frame
4. Report the Animation
5. Quit
```