

Raster Graphic Project in C++

Due Time: 23.59, Sat 13 October 2018 **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 a continuation of assignment #0 but in C++. You are to write the code in Visual Studio 2015 for a C++ language console application that holds the data of a raster graphic application (there is no actual graphics in the assignment) using a forward list (aka singly-linked list) in dynamic memory for its data. Part of the code is shown on the next page; it is also on Brightspace in a text file that you can copy and paste. You **MUST** use this code **without modification (not a single character changed): no code added or removed, no new classes, no new global variables, no macros, no defines and no statics**. Your task is to write in C++ in the RasterGraphic.cpp and GraphicElement.cpp files the definitions of the member functions of the RasterGraphic and GraphicElement classes. You must not add any new functions or classes. The files you submit are RasterGraphic.cpp and GraphicElement.cpp.

The RasterGraphic is a series of GraphicElements held in a forward list. When the list runs it displays the details of each GraphicElement at intervals of 1 second using the system clock (you can adapt the code from assignment #0 for this).

You can:

- add a new GraphicElement to the RasterGraphic at a position selected by the user,
- delete the first GraphicElement in the list,
- report the RasterGraphic to show the list of GraphicElement details one after another at 1 second intervals (code supplied),
- quit leaving no allocated dynamic memory (i.e. no memory leak),

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 are only RasterGraphic.cpp and GraphicElement.cpp,
- dynamic memory management is done with new and delete and only the minimal memory is allocated at any given time,
- you can only use functions like strlen() and strcpy() or similar etc. from the standard C library to handle strings of null terminated char arrays,
- when the application terminates it releases all dynamically allocated memory so it does not have a 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 (not a single character) - no code added (except header includes) or removed, no macros, no defines, no statics and no additional global functions or variables,
2. it fails to build in Visual Studio 2015,
3. It crashes in normal operation (such as running an empty list etc.),
4. it doesn't produce the example output.

Part of the code is shown on the next page. You **MUST** use this code without modification. Your task is to write, in C++ in the RasterGraphic.cpp and GraphicElement.cpp files, the definitions of the member functions of the RasterGraphic and GraphicElement classes using the style of the posted Submission Standard.

CST 8219 – F18 - Assignment #1

What to Submit: Use Brightspace to submit this assignment as a zip file (**not** RAR, not 9zip, not 7 zip) containing only the source code files RasterGraphic.cpp and GraphicElement.cpp. The name of the zipped folder **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 file headers so they can be identified as yours.

Before you submit the code,

- check that it builds and executes in Visual Studio 2015 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.

There is a late penalty of 25% per day. Don't send me the files as email attachments – they will get 0.

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

<pre>// GraphicElement.h #pragma once class GraphicElement { char* fileName; GraphicElement* pNext; public: GraphicElement(); ~GraphicElement(); char*& GetfileName() { return fileName; } // inline GraphicElement*& GetpNext(){ return pNext; }; // inline };</pre>	<pre>//RasterGraphic.h #pragma once class RasterGraphic { GraphicElement* GraphicElements; public: RasterGraphic(); ~RasterGraphic() { delete GraphicElements; } // inline void InsertGraphicElement(); void DeleteGraphicElement(); void ReportRasterGraphic(); };</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> #include "GraphicElement.h" #include "RasterGraphic.h" using namespace std; bool running = true; int main(void) { char selection; bool running = true; RasterGraphic A; _CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF _CRTDBG_LEAK_CHECK_DF); while (running) { cout<<"MENU\n 1. Insert a GraphicElement\n 2. Delete a GraphicElement\n 3. Report the RasterGraphic\n 4. Quit\n"<<endl; cin>>selection; switch (selection) { case '1': A.InsertGraphicElement(); break; case '2': A.DeleteGraphicElement(); break; case '3': A.ReportRasterGraphic(); break; case '4': running = false; break; default: break; } } return 0; }</pre>	

Example Output:

```
MENU
1. Insert a GraphicElement
2. Delete a GraphicElement
3. Report the RasterGraphic
4. Quit

1
INSERT A GRAPHIC ELEMENT
Please enter the GraphicElement filename: GraphicElement_1
```

CST 8219 – F18 - Assignment #1

This is the first GraphicElement in the list

MENU

1. Insert a GraphicElement
2. Delete a GraphicElement
3. Report the RasterGraphic
4. Quit

1

INSERT A GRAPHIC ELEMENT

Please enter the GraphicElement filename: GraphicElement_2

There are 1 GraphicElement(s) in the list. Please specify the position (<= 1) to insert at : 1

MENU

1. Insert a GraphicElement
2. Delete a GraphicElement
3. Report the RasterGraphic
4. Quit

1

INSERT A GRAPHIC ELEMENT

Please enter the GraphicElement filename: GraphicElement_3

There are 2 GraphicElement(s) in the list. Please specify the position (<= 2) to insert at : 1

MENU

1. Insert a GraphicElement
2. Delete a GraphicElement
3. Report the RasterGraphic
4. Quit

3

REPORT THE RASTER GRAPHIC

GraphicElement #0, time = 1 sec

Image file name = GraphicElement_1

GraphicElement #1, time = 2 sec

Image file name = GraphicElement_3

GraphicElement #2, time = 3 sec

Image file name = GraphicElement_2

MENU

1. Insert a GraphicElement
2. Delete a GraphicElement
3. Report the RasterGraphic
4. Quit

2

MENU

1. Insert a GraphicElement
2. Delete a GraphicElement
3. Report the RasterGraphic
4. Quit

3

REPORT THE RASTER GRAPHIC

GraphicElement #0, time = 1 sec

Image file name = GraphicElement_3

GraphicElement #1, time = 2 sec

Image file name = GraphicElement_2

MENU

1. Insert a GraphicElement
2. Delete a GraphicElement
3. Report the RasterGraphic
4. Quit

2

MENU

1. Insert a GraphicElement
2. Delete a GraphicElement
3. Report the RasterGraphic
4. Quit

2

MENU

1. Insert a GraphicElement
2. Delete a GraphicElement
3. Report the RasterGraphic
4. Quit

3

No GraphicElements in the RasterGraphic

MENU

1. Insert a GraphicElement
2. Delete a GraphicElement
3. Report the RasterGraphic
4. Quit