

## Vector Graphic with Classes

**Due Time:** 23.59, 18 Feb 2017 **Earnings:** 8% of your final grade

**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 assignment is a direct continuation of assignment 0 that uses C++ syntax. It uses the Stash2.h and Stash2.cpp code from the Stash2 example from "Thinking in C++" V1 (the C++ course book), so that must be added to your Visual Studio project.

Part of the code is shown on the next page; it is also on the Web Site in text files that you can copy and paste from so you don't make any mistakes. You must use this code without modification because I will use it to mark your assignment. Your task is to implement the member functions that are declared in the GraphicElement.h and VectorGraphic.h header files and not add any new ones - your code is in the files GraphicElement.cpp and VectorGraphic.cpp.

In this assignment, when the application is running the user can

- Add a new Graphic Element (together with its lines) to the Vector Graphic
- Print all the details of the Graphic Elements in the Vector Graphic

**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 GraphicElement.cpp and VectorGraphic.cpp.
- you must use C++ syntax including `new` and `delete` and `cin` and `cout` etc.
- You can only use functions like `strlen()` and `strcpy()` or similar etc. from the standard C library to handle strings (you cannot use the C++ string class).
- When the application terminates it releases all dynamically allocated memory so it does not have a resource leak (or you lose 30%).

See the Marking Sheet for how you can lose marks, but you will lose at least 60% if: you change the supplied code, it fails to build in Visual Studio 2013, it crashes in normal operation (such as printing from an empty list etc.), 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 add the implementation of the functions that are declared using the style of the posted Submission Standard. All the code is in the files GraphicElement.cpp and VectorGraphic.cpp.

**What to Submit :** Use Blackboard to submit this assignment as a zip file (**not** RAR) containing only the source code files (GraphicElement.cpp and VectorGraphic.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 the Cover Information (as specified in the Submission Standard) as a file header in your source file so the file 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 2013 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's a late penalty of 25% per day. Don't send me file(s) as an email attachment – it will get 0.

## CST 8219 – W17 - Assignment #1

**Example code – I will use it to mark your assignment. Don't change or add to it.**

<pre>#ifndef GRAPHICELEMENT #define GRAPHICELEMENT  class GraphicElement {     char* name;     Stash Lines;          // a Stash of Line objects public:     GraphicElement():name(nullptr),Lines(sizeof(Line)){}     ~GraphicElement()     {         if (name)             delete []name;     }     char*&amp; GetName();    // Accessor     Stash&amp; GetLines();   // Accessor };  #endif</pre>	<pre>// Line.h  #ifndef LINE #define LINE  struct Point {     int x, y; };  struct Line {     Point start;     Point end; };  #endif</pre>
<pre>// ass1 W17  #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 &lt;crtdbg.h&gt; #include "Stash2.h" #include "Line.h" #include "GraphicElement.h" #include "VectorGraphic.h" #include &lt;iostream&gt; using namespace std;  enum{ RUNNING = 1 };  VectorGraphic Image;  int main() {     char response;     _CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF   _CRTDBG_LEAK_CHECK_DF);      while (RUNNING)     {         cout&lt;&lt;endl&lt;&lt;"Please select an option:\n"&lt;&lt;endl;         cout&lt;&lt;"1. Add a Graphic Element\n";         cout&lt;&lt;"2. List the Graphic Elements\n";         cout&lt;&lt;"q. Quit\n";         cout&lt;&lt;"CHOICE: ";         cin&gt;&gt;response;          switch (response)         {             case '1':Image.AddGraphicElement(); break;             case '2':Image.ReportVectorGraphic(); break;             case 'q':return 0;             default:cout&lt;&lt;"Please enter a valid option\n";         }         cout&lt;&lt;endl;     }     return 0; }</pre>	<pre>// VectorGraphic.h  #ifndef VECTORGRAPHIC #define VECTORGRAPHIC  class VectorGraphic {     static const int MAX_ELEMENTS = 1024;     GraphicElement Elements[MAX_ELEMENTS]; // pool of elements     unsigned int numElements;// number filled public:     VectorGraphic():VectorGraphic():numElements(0){}     void AddGraphicElement();     void ReportVectorGraphic(); };  #endif</pre>

## CST 8219 – W17 - Assignment #1

### Example Output:

```
Please select an option:

1. Add a Graphic Element
2. List the Graphic Elements
q. Quit
CHOICE: 1
Adding A Graphic Element
Please enter the name of the new GraphicElement(<256 characters): small mouth
How many lines are there in the new GraphicElement? 1
Please enter the x coord of the start point of line index 0 -2
Please enter the y coord of the start point of line index 0 -2
Please enter the x coord of the end point of line index 0 2
Please enter the y coord of the end point of line index 0 -2
```

```
Please select an option:

1. Add a Graphic Element
2. List the Graphic Elements
q. Quit
CHOICE: 1
Adding A Graphic Element
Please enter the name of the new GraphicElement(<256 characters): big nose
How many lines are there in the new GraphicElement? 4
Please enter the x coord of the start point of line index 0 -1
Please enter the y coord of the start point of line index 0 1
Please enter the x coord of the end point of line index 0 -1
Please enter the y coord of the end point of line index 0 2
Please enter the x coord of the start point of line index 1 -1
Please enter the y coord of the start point of line index 1 2
Please enter the x coord of the end point of line index 1 1
Please enter the y coord of the end point of line index 1 2
Please enter the x coord of the start point of line index 2 1
Please enter the y coord of the start point of line index 2 2
Please enter the x coord of the end point of line index 2 1
Please enter the y coord of the end point of line index 2 1
Please enter the x coord of the start point of line index 3 1
Please enter the y coord of the start point of line index 3 1
Please enter the x coord of the end point of line index 3 -1
Please enter the y coord of the end point of line index 3 1
```

```
Please select an option:

1. Add a Graphic Element
2. List the Graphic Elements
q. Quit
CHOICE: 2
VectorGraphic Report

Reporting Graphic Element #0
Graphic Element name: small mouth
Line #0 start x: -2
Line #0 start y: -2
Line #0 end x: 2
Line #0 end y: -2
```

```
Reporting Graphic Element #1
Graphic Element name: big nose
Line #0 start x: -1
Line #0 start y: 1
Line #0 end x: -1
Line #0 end y: 2
Line #1 start x: -1
Line #1 start y: 2
Line #1 end x: 1
Line #1 end y: 2
Line #2 start x: 1
Line #2 start y: 2
Line #2 end x: 1
Line #2 end y: 1
Line #3 start x: 1
Line #3 start y: 1
Line #3 end x: -1
Line #3 end y: 1
```

```
Please select an option:

1. Add a Graphic Element
2. List the Graphic Elements
q. Quit
CHOICE:
```