

**Spike: 3****Title:** Debugger Use**Author:** Krishna Adhikari, 4953193**Goals / deliverables:**

The goal of this spike was to use the debugger tool provided in Visual Studio to locate the bugs and fix them. The given project was downloaded and debugging process was implemented.

Besides this report, the visual studio solution was modified to obtain the desired outcome.

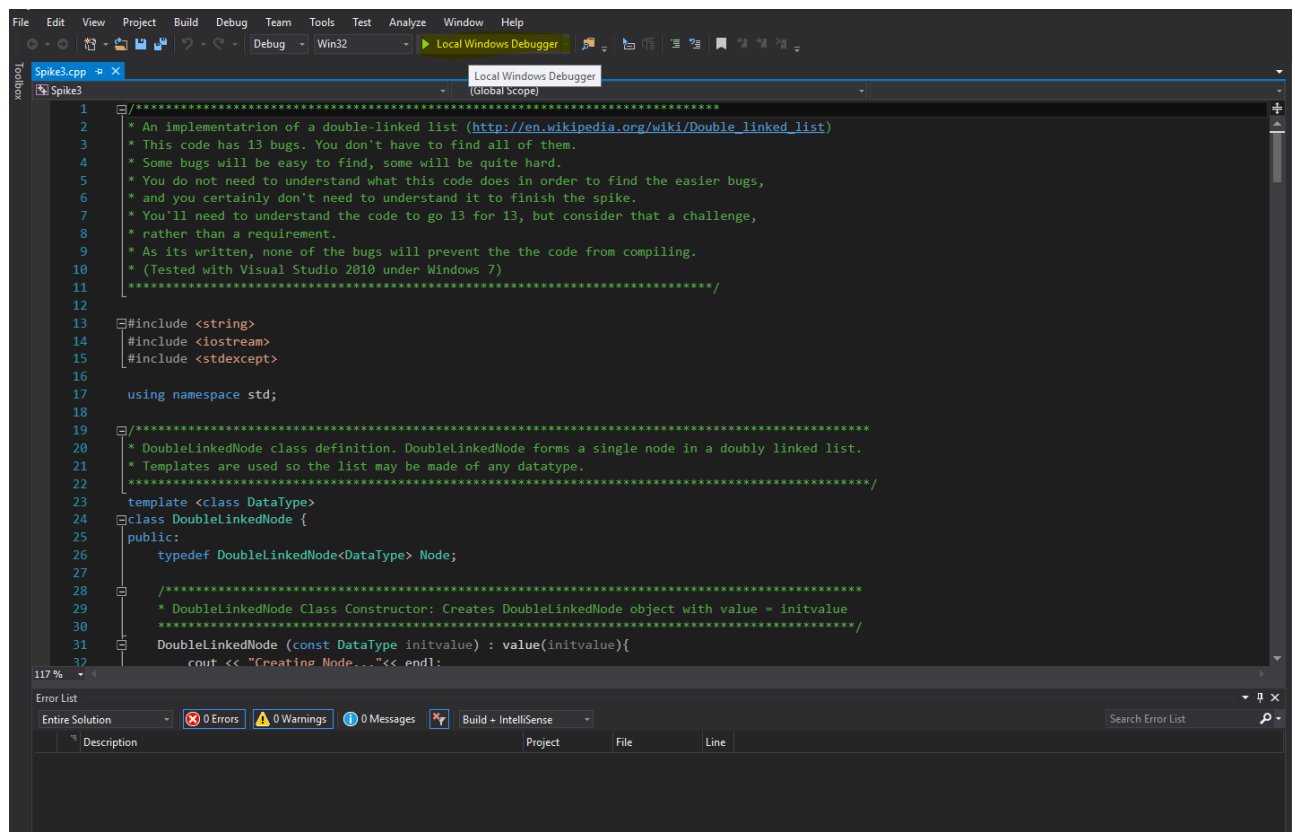
**Technologies, Tools, and Resources used:**

List of information needed by someone trying to reproduce this work

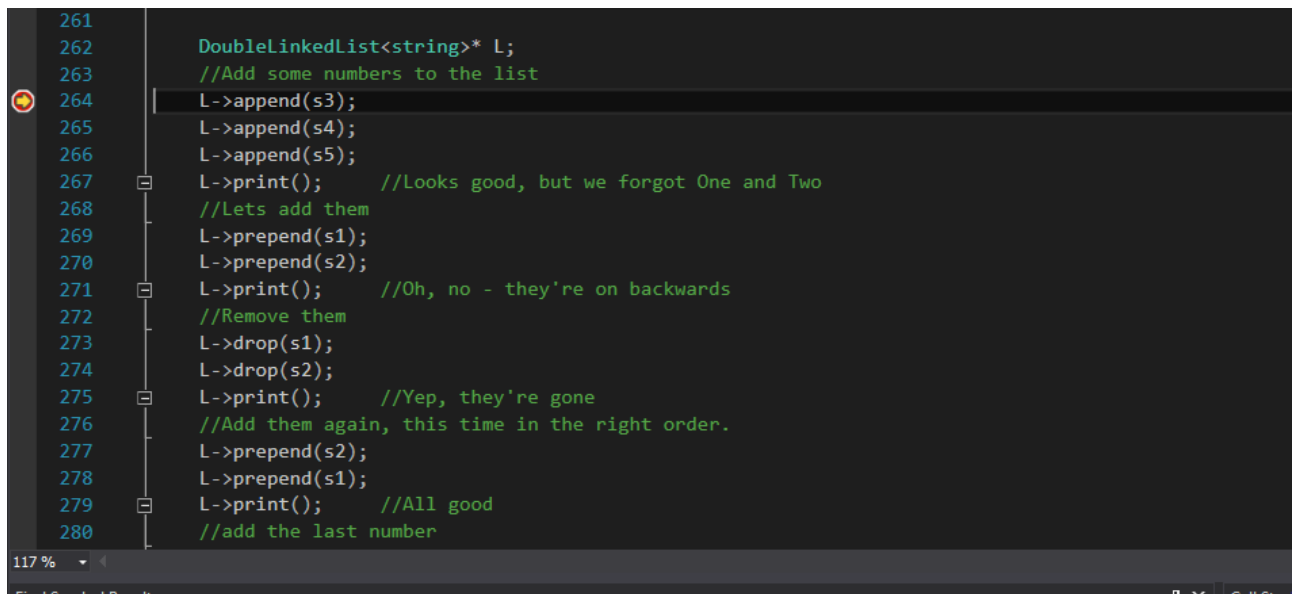
- Visual Studio 2010
- [http://www.tutorialspoint.com/data\\_structures\\_algorithms/doubly\\_linked\\_list\\_algorithm.htm](http://www.tutorialspoint.com/data_structures_algorithms/doubly_linked_list_algorithm.htm)
- <https://www.geekboots.com/cpp/doubly-linkedlist>

**Tasks undertaken:**

- Open the solution in the Visual Studio 2015
- Run the solution in the Local Window Debugger. The option has been highlighted in the screenshot below.

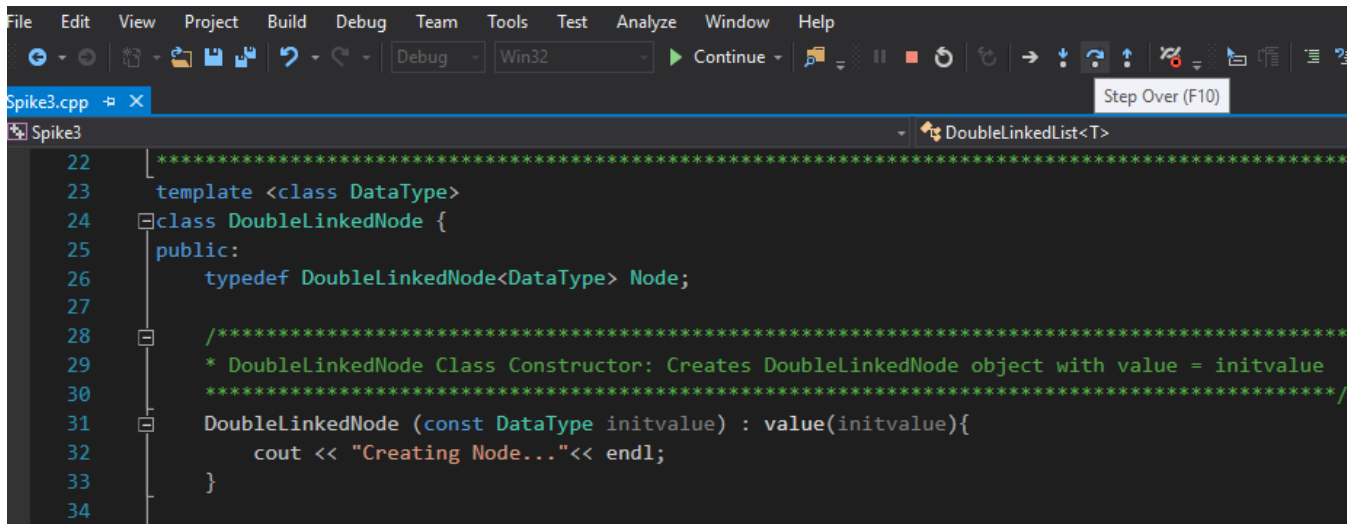


- The program breaks when a bug is discovered. The line will be highlighted. Try to fix the bug. In the below case, L has not been initialized.



```
261
262     DoubleLinkedList<string>* L;
263     //Add some numbers to the list
264     L->append(s3);
265     L->append(s4);
266     L->append(s5);
267     L->print();    //Looks good, but we forgot One and Two
268     //Lets add them
269     L->prepend(s1);
270     L->prepend(s2);
271     L->print();    //Oh, no - they're on backwards
272     //Remove them
273     L->drop(s1);
274     L->drop(s2);
275     L->print();    //Yep, they're gone
276     //Add them again, this time in the right order.
277     L->prepend(s2);
278     L->prepend(s1);
279     L->print();    //All good
280     //add the last number
```

- After fixing the bug, repeat the same process to discover the next bug.
- Repeat the process, until the program is running.
- You might find Step Over functionality really helpful as it shows every line of code that is being executed. You can find out where exactly, the code is breaking. Below is the screenshot.



```
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug Win32 Continue
Spike3.cpp
Spike3 DoubleLinkedList<T>
22
23     template <class DataType>
24     class DoubleLinkedListNode {
25     public:
26         typedef DoubleLinkedListNode<DataType> Node;
27
28         /*
29         * DoubleLinkedListNode Class Constructor: Creates DoubleLinkedListNode object with value = initvalue
30         */
31         DoubleLinkedListNode (const DataType initvalue) : value(initvalue){
32             cout << "Creating Node..."<< endl;
33         }
34
```

## What we found out:

After this exercise, I learnt about how double linked lists work. I got the understanding of working with the pointers, lists and nodes. Apart from them, I also enhanced my skills about the debugging tools provided by Visual Studio 2015. I used breakpoints and also used the Step Over functionality a lot to debug. Below is a screenshot of the program running.

```
C:\WINDOWS\system32\cmd.exe
Creating Node...
Creating Node...
Creating Node...
Three
Four
Five
-----
Creating Node...
Creating Node...
Two
One
Three
Four
Five
-----
Destroying Node...
Destroying Node...
Three
Four
Five
-----
Creating Node...
Creating Node...
One
Two
Three
Four
Five
-----
Creating Node...
One
Two
Three
Four
Five
Six
-----
Destroying Node...
Destroying Node...
Destroying Node...
Destroying Node...
Destroying Node...
Press any key to continue . . .
```