# Title

Lab 5: Test Driven Development

# Team

Aaron Decker

*Brendan Compton*

*Dillon Hiatt*

*Alex Richter*

*Dylan Jones*

# Team Summary Time Log

|  |  |  |
| --- | --- | --- |
| **Team Member** | **Time**  **( in minutes )** | **Activities (description)** |
| Aaron Decker |  |  |
| Brendan Compton |  |  |
| Dillon Hiatt |  |  |
| Alex Richter |  |  |
| Dylan Jones |  |  |
| **TEAM TOTAL** |  | Total time needed to produce this deliverable. |

## Time Reporting Problems *(all manual entry of time Shall be noted here)*

|  |  |  |
| --- | --- | --- |
| **Who** | **Date** | **Problem Documentation** |
|  |  |  |
|  |  |  |

# Why are we doing this Deliverable

*Dylan Jones:*

We are doing lab 5 in order to understand extreme programming and test driven development in a more realistic environment. By keeping track of our progress using subversion and a spreadsheet, we are able to see each step of the process as it works together. In the spreadsheet, we are able to see how Wake's Traffic Light metaphor works with test driven development. Each iteration includes writing the test case so the code doesn't compile, writing stubs so the code compiles but the test case fails, and finally writing the functionality to pass the test case. Lab 5 demonstrates how this process works well in software engineering.

Alex Richter:  
We are doing lab5 to get practice with Test Driven Development, and extreme programming. TDD also provides us with practice for agile development. It also shows us the process of documenting changes into a test document, for this example it is by using the Wake’s traffic light metaphor. Since we are only doing portions of code for this lab, we are also learning how to use test stubs so that we can make our test functions work. Finally by using SVN we can commit at each stage of a traffic light, so that we have records of changes and completed testing.

Dillon Hiatt

We are doing lab5 in order to understand Test Driven Development. We are using the Wake’s Traffic light metaphor to understand when the test is in the following stage: Yellow, Red, and Green. We are keeping track of our progress with the subversion tracking log. Each iteration the test case consist of code that doesn’t compile, writing a stub that the code compiles but the test case fails and the last test case the functionality passes the test case.

Brendan Compton

We are doing lab 5 to gain a better understanding of the TDD cycle. With TDD and the Traffic light metaphor it is possible to understand whether or not code is working immediately after writing it, which is extremely useful when creating quality software with an iterative model. By keeping track of the development and testing process using the spreadsheet and SVN logs, we can measure the process, so that it can eventually be improved.

# Project files for this Deliverable

|  |  |  |  |
| --- | --- | --- | --- |
| **File Name** | **Path** | **Purpose** | **Final Version**  **& Date** |
| Group1\_Lab5\_Code | S:\Courses\CSSE\shiy\3730\Group1\ Group1\_Lab5\_Code | The code for lab 5 |  |
| Group1\_Lab5\_Log | S:\Courses\CSSE\shiy\3730\Group1\ Group1\_Lab5\_Log | The log for lab 5 |  |
| Group1\_Lab5\_TrackSheet | S:\Courses\CSSE\shiy\3730\Group1\ Group1\_Lab5\_TrackSheet | The track sheet for lab 5 |  |
| Group1\_Lab5\_Report.docx | S:\Courses\CSSE\shiy\3730\Group1\ Group1\_Lab5\_Report.docx | The report for lab 5 |  |
|  |  |  |  |

# Method / Process

*The process for this lab is to write a linked list class and implement each function using Extreme Programming and Test Driven Development. Using a spreadsheet and subversion to track our progress, we implement a function as a single iteration. Each iteration includes writing the test case, writing the stub function so the program will compile but the test will fail, and writing the actual code to implement the functionality. Each time the code is compiled, it is also checked into subversion in order to keep track of the process.*

# Results

*At the end of the lab, we produced a linked list class with unit tests to prove its functionality. The logs and tracksheet demonstrate show how we used extreme programming and test driven development to develop the class and its unit tests.*