**Programming Assignment 1 Sprint Report**

Name: Matt Fletcher

**Sprint 1**

Epic: As a student of CS 221 I need to create the basic program architecture.

|  |  |
| --- | --- |
| **Backlog of User Stories** | **Done Date\*** |
| Create the project in Visual Studio, add a source file and add a main function to the source file. | 29 Aug |
| Add .h and .cpp files for the Student class. | 29 Aug |
| Write the class definition in Student.h. | 30 Aug |
| Write stub function definitions in Student.cpp for each function in the class. (Functions returning a value should return zero, NULL, etc. as appropriate.) | 31 Aug |
| Add a cout line to each function just to report that the function was reached. Do this for the constructors and destructor also. | 1 Sept |
| Add code to main to create an instance of Student using the default constructor. Call all functions and verify that all were reached. | 2 Sept |
| Add code to main to create an instance of Student using the parameterized constructor. Call all functions and verify that all were reached. | 2 Sept |

**Sprint 2**

Epic: As a student of CS 221 I need to plan how to test each function.

Epic: As a student of CS 221 I need to implement and verify each function.

|  |  |
| --- | --- |
| **Backlog of User Stories** | **Done Date\*** |
| For each function in Student determine exactly how you will test the function automatically from main and add code to perform each of the automatic tests. | 3 Sept |
| **Add code, test, and verify functions on which other functions will depend.** |  |
| Student () - default constructor done | 5 Sept |
| printStudentInfo() done | 12 Sept |
| **Add code, test and verify all other functions.** |  |
| Student (…) - parameterized constructor | 12 Sept |
| getStudentID(), setStudentID () | 12 Sept |
| getName(), setName() | 13 Sept |
| getHouse(), setHouse() | 12 Sept |
| getClass(), setClass() | 12 Sept |
| setGrade() | 13 Sept |
| getGrade() – using references | NF\* |
| getGrade() – using pointers | NF\* |
| Verify against the SOW that all functionality of the project has been fully implemented. | \*\* |

**\*Done means you have implemented the code AND fully tested it.**

**Double off if you say you have tested it and it fails when tested by the instructor.**

**\*Not functional**

**\*\* All parts of SOW have either been implemented or attempted. getGrade() has been implemented but is not fully working.**