# New Leaf Term Project Report

## Section 2: Test Plan

Luke McGuire, Chris Taylor, Kasper Dugaw

College of Charleston

Fall 2020

### Introduction

In this section of our term project we will discuss our test plan and describe our methodology for determining and implementing test cases and developing the testing framework that will automate these tests.

### The Testing Process

- Determine what methods / functions have the most urgent need for testing for the project's purposes & isolate them
- For each method, determine a set of suitable test cases
- Implement test cases utilizing Oracle & ensure they are able to run via the terminal; confirm Oracle output vs expected output

### What We Are Testing

Moodle includes a central math library called EvalMath which evaluates all of their string mathematical expressions for math questions within the LMS. We will be testing the evaluate method (shortened as "e"; these methods are identical) from this library for both edge cases and normal use cases, all of which are detailed in this document.

Additionally, for future test cases we will evaluate other core components of the moodle project and identify other testable methods and functionality, for example methods for retrieving user data and manipulating users.

### Requirements traceability

Requirements for EvalMath are basic mathematical literacy; evaluate takes an expression in as a string and returns the result. If the string is not parseable as a mathematical expression, it should throw an error. If it is, the result is obviously expected to be correct.

**Testing schedule**

The plan for the test cases on e will be complete by October 6, 2020. A full framework with functioning test cases will be complete by November 5, 2020; more deadlines can be found on our wiki.

**Test recording procedures**

Tests will be run automatically in a batch program, and output an HTML report of its results.

**Hardware and software requirements**

Hardware required: Ubuntu virtual machine with enough memory and space to run Moodle
Software required:

**Constraints**

Both time and available manpower are potential constraints to this project.

**System tests**

Check that the tests run properly with the expected results in a clean system, implying no reliance on particular file paths or components from individual machines.

**Test case specifications template:**

Fields:

1. test number or ID
2. requirement being tested
3. component being tested
4. method being tested
5. test input(s) including command-line argument(s)
6. expected outcome(s)

Json File:

```
{
        “id”: 1,
        “requirement”: “Requirement”,
        “component”: “Component Name”,
        “method”: “Method Name”,
        “input”: “input”,
        “output”: “output”
}
```

**Defining Test Cases 1-5**

Test cases 1 through 5 have been defined and can be found in the TestCases directory of our project. As discussed, these first 5 test cases focus upon testing the functionality of the evalmath function in moodle's math library. For example, in one of the test cases we check the functionality of the addition subroutine by defining " 4 + 2 " as input, during the test this input will be passed to the method and an output of 6 is expected.