A Testing Tool for Introductory Programming Courses

Thesis B Seminar

Kyu-Sang Kim
z5208931

Supervised by Andrew Taylor (UNSW)
Assessed by John Shepherd (UNSW)

Term 2, 2022

Contents

Thesis Statement

A user-friendly and maintainable general code testing tool is important to streamline the administration of introductory programming courses

Thesis Statement

A user-friendly and maintainable general code testing tool is important to streamline the administration of introductory programming courses

We will:

Thesis Statement

A user-friendly and maintainable general code testing tool is important to streamline the administration of introductory programming courses

We will:

1. Develop an extensible and easy to use software package which parses and runs pre-written tests on submitted code

Thesis Statement

A user-friendly and maintainable general code testing tool is important to streamline the administration of introductory programming courses

We will:

1. Develop an extensible and easy to use software package which parses and runs pre-written tests on submitted code
2. Implement development procedures that minimise both current and future technical debt

Thesis Statement

A user-friendly and maintainable general code testing tool is important to streamline the administration of introductory programming courses

We will:

1. Develop an extensible and easy to use software package which parses and runs pre-written tests on submitted code
2. Implement development procedures that minimise both current and future technical debt
3. Remediate known flaws in the existing autotest package

Thesis Statement

A user-friendly and maintainable general code testing tool is important to streamline the administration of introductory programming courses

We will:

1. Develop an extensible and easy to use software package which parses and runs pre-written tests on submitted code
2. Implement development procedures that minimise both current and future technical debt
3. Remediate known flaws in the existing autotest package
4. Maintain backwards compatibility with legacy tests written for the existing autotest package

Thesis Statement

A user-friendly and maintainable general code testing tool is important to streamline the administration of introductory programming courses

We will:

1. Develop an extensible and easy to use software package which parses and runs pre-written tests on submitted code
2. Implement development procedures that minimise both current and future technical debt
3. Remediate known flaws in the existing autotest package
4. Maintain backwards compatibility with legacy tests written for the existing autotest package
5. Perform proving and performance tests on the new software package

Thesis Statement

A user-friendly and maintainable general code testing tool is important to streamline the administration of introductory programming courses

We will:

1. Develop an extensible and easy to use software package which parses and runs pre-written tests on submitted code
2. Implement development procedures that minimise both current and future technical debt
3. Remediate known flaws in the existing autotest package
4. Maintain backwards compatibility with legacy tests written for the existing autotest package
5. Perform proving and performance tests on the new software package
6. Deprecate and replace the existing autotest used for introductory programming courses at UNSW CSE

Schedule

The original plan was set to the following:

1. Thesis B:
   - Implement Core Main Module

Schedule

The original plan was set to the following:

1. Thesis B:
   - Implement Core Main Module
   - Implement Core Testcase Parser Module

Schedule

The original plan was set to the following:

1. Thesis B:
   - Implement Core Main Module
   - Implement Core Testcase Parser Module
   - Implement Core Testcase Runner Module

Schedule

The original plan was set to the following:

1. Thesis B:
   - Implement Core Main Module
   - Implement Core Testcase Parser Module
   - Implement Core Testcase Runner Module
   - Run correctness and performance testing on Parser and Runner

Schedule

The original plan was set to the following:

1. Thesis B:
   - Implement Core Main Module
   - Implement Core Testcase Parser Module
   - Implement Core Testcase Runner Module
   - Run correctness and performance testing on Parser and Runner
2. Thesis C:
   - Implement Core Testcase Program Correctness Module

Schedule

The original plan was set to the following:

1. Thesis B:
   - Implement Core Main Module
   - Implement Core Testcase Parser Module
   - Implement Core Testcase Runner Module
   - Run correctness and performance testing on Parser and Runner
2. Thesis C:
   - Implement Core Testcase Program Correctness Module
   - Implement any extensions that have been deemed necessary by the Design Document

Schedule

The original plan was set to the following:

1. Thesis B:
   - Implement Core Main Module
   - Implement Core Testcase Parser Module
   - Implement Core Testcase Runner Module
   - Run correctness and performance testing on Parser and Runner
2. Thesis C:
   - Implement Core Testcase Program Correctness Module
   - Implement any extensions that have been deemed necessary by the Design Document
   - Run correctness and performance testing on complete package and make final adjustments