CMPSC 112
Introduction to Computer Science I
Fall 2016

Final Project: Real-World Applications of Computer Science

Introduction

So far you have learned more about the fundamentals of computer science and Java programming by studying, in a hands-on fashion, topics such as the use and creation of object-oriented Java programs, recursion, fundamental data structures, and algorithm analysis. This final project invites you to explore, in greater detail, a real-world application of computer science. You will learn more about how to use, implement, test, and evaluate different types of real-world computer software. Since you will complete the final project with a partner, you will also learn more about how the Git version control system can effectively support collaborative software development.

Your project should result in a detailed report that includes all of your source code, in addition to written materials and technical diagrams that highlight the key contributions of your work. The report should include a description of why the chosen topic is important and discuss the implementation and/or experimentation that you undertook. The written material should be precise, formal, appropriately formatted, grammatically correct, informative, and interesting. The source code that you write must be carefully documented and tested. If you install and use existing computer software (e.g., a Java library for performance evaluation), the steps for installation and use should be clearly documented in your report. Also, the report must explain the steps to run your own Java program. Finally, your paper must detail the work completed by each member of your partnership; individual contributions should also be reflected in the Git repository's log.

Description of the Topics

Each partnership is invited to pick one of the following projects. Please note that a partnership selecting the student-designed project must first discuss the idea with the course instructor, during today's laboratory session, and receive feedback and then final approval. Please note that you and your partner are fully responsible for ensuring the feasibility of the project that you propose.

1. **Cryptography and Cryptanalysis**: Explore a topic in the fields that make up the "art and science of sending and decoding secret messages". This project invites you and your partner to implement and test several cryptography and/or cryptanalysis systems. To start, you should investigate, implement, and test ciphers such as the Caesar and Vigenere ciphers. Then, you should use your ciphers to demonstrate that you can successfully send secret messages through, for instance, an email server. In addition to creating and testing these Java programs, your report should include a detailed explanation of how your chosen algorithms work. To learn more about this topic, please review Section 3.1.4 of the textbook.

2. **Data Structures**: This topic invites you to either investigate a data structure that we did not already study this semester or, alternatively, further look into a data structure that we previously discussed. A team that selects this project might review the textbook's content about a specific data structure (e.g., Chapter 6's discussion of stacks, queues, and double-ended queues) and then identify one area for further study. For instance, you could decide

to implement and test your own version of these data types that is “backed” by a dynamic array. Alternatively, you could use as an inspiration some of the guidelines in Chapter 7 and implement your own dynamic array that can both grow and shrink in size.

3. **Performance Evaluation**: Since it is often important to implement computer software that exhibits acceptable time and space overheads, this project invites students to use and/or extend performance evaluation software like Expose. After finding, reading, and understanding textbooks and research papers on this topic, students who pick this project should identify a focus area and create/extend the benchmarking framework. The final version of the framework should allow students to measure the performance of computations in Java, report those measurements to the user of the benchmarks, and support informed decision making about design and implementation trade-offs. Along with including the source code of the benchmarking framework, this project invites students to write a performance evaluation report. To learn more about this topic, a team should carefully review Chapter 4 of the textbook.

4. **Software Testing**: In response to the fact that real-world software often contains serious defects, this project encourages students to learn more about techniques that can find program errors before they are delivered to end-users. Students who choose this topic will investigate software testing tools, such as JUnit and EvoSuite, and then create a software system that includes a test suite. In addition to implementing Java classes and methods that provide the main functionality, you will, as part of this project, also write (or, generate) tests that assess the correctness of the aforementioned methods. If your team picks this project, you should submit programs and test suites in addition to a report that explains your approach to software testing. To learn more about his topic, please review Section 1.9 of the textbook.

5. **Searching and Sorting**: In a previous laboratory assignment you experimentally studied the performance of sorting algorithms. A team that selects this project could implement and then analytically (i.e., with a proof) and empirically study the performance of a sorting algorithm. Or, you could investigate an algorithm that searches through a data structure (e.g., a tree or a hashtable) and looks for data that matches a specific pattern. Students who pick this project should review Section 3.1.2 and further study Chapter 12 of the textbook.

6. **System Implementation**: Now that you understand the key topics of this course, you are ready to design, implement, and test your own program. A team that picks this project should first identify a problem domain in which they will work. For instance, you might decide that you want to implement your own command-line client for the Twitter social network by investigating the Twitter4J library. Then, you will need to decide on the data structures and algorithms that you will need to store and manipulate your data. If you pick this project, then you should review Chapter 2 and the chapter(s) for your chosen data structures.

7. **Student-Designed Project**: Students will develop an idea for their own project that focuses on one or more real-world topics in the field of computer science. After receiving the course instructor’s approval for your idea, you will complete the project and report on your results.

Project Requirements

Whenever it is feasible and appropriate to do so, your final project submission should include well-documented source code, a test suite to demonstrate the correctness of your program, and a written report to explain your design and/or experimental results. If your final project includes informal proofs of an algorithm's worst-case time complexity, then that should also be submitted. The teams, consisting of two students, must create a new version control repository for the final project. The name of the repository must adhere to the following naming convention: "`cs112F2016<first-user-name>-<second-user-name>`" where both "`<first-user-name>`" and "`<seconduser-name>`" correspond to the user name of the Allegheny-approved email address for a student in this course. Please make sure that the course repository, which was initially created by one member of the team, is shared with both the other team member and the course instructor.

Final Project Deadlines

This assignment invites you to submit printed and signed versions of the following deliverables:

1. **Project Assigned and Project Proposal:** Monday, December 5, 2016

   After meeting with the course instructor and your partner, pick a topic for your final project. Remember, if your team selects the student-designed project, you must first have your project approved by the course instructor. Next, make sure that you create a Git repository that can be accessed by the instructor. Finally, write and submit a one-page proposal for your project. While you can use the project descriptions on the previous pages as a starting point, your proposal should have an informative title, an abstract, a description of the main idea, an initial listing of the tasks that you must complete, and a plan for completing the work.

2. **Status Update and Project Demonstration**: Monday, December 12, 2016

   As you continue working on your project, please submit a one paragraph status update in printed form and through your Git repository. In addition, you should give a demonstration, during the laboratory session, highlighting the most important code that you have finished. Your demonstration should clearly illustrate how much of the project you have finished and why it is feasible for you and your partner to finish the final project by the stated deadline.

3. **Final Project Due Date**: Thursday, December 15, 2016 by 5 pm

   You should submit the final version of your project, in printed form and through the Git repository. This submission should include all of the relevant source code and output, the written report, and any additional materials that will demonstrate the success of your project. While you are encouraged to turn in the final project before the examination session ends on the due date, students must submit the assignment no later than 5 pm on the due date.

In adherence to the Honor Code, students should complete this assignment while exclusively collaborating with the other member of their team. While it is appropriate for students in this class—who are not in the same team—to have high-level conversations about the assignment, it is necessary to distinguish carefully between the team that discusses the principles underlying a problem with another team and the team that produces an assignment that is identical to, or merely a variation on, the work of another team. Deliverables from one team that are nearly identical to the work of another team will be taken as evidence of violating Allegheny College's Honor Code. Please see the instructor if you have questions about the Honor Code. Finally, unless you ask the instructor not to do so, photographs of you completing the final project with your partner will be taken during laboratory sessions and shared on social networks like Twitter and Instagram.