

# Semester Project

## CMPT 440

Pablo Rivas

Assigned: Jan/23/17; Points: 100

## 1 Overview and Timeline

The goal of the semester project is to help you work on a larger project related to your major, computer science, information systems, information technology, mathematics, etc. You will gain experience in the development, implementation, analysis, and/or application of the things we discuss in class. It will help prepare you for further work in software development, or help you use Java for another project, e.g., your senior project, thesis, or even an independent study.

The timeline for this project is the following:

- **Proposals:** due via GitHub by 6:30 PM on Monday, February 6.
- **Milestone:** due via GitHub by 6:30 PM on Monday, April 3.
- **10-minute demo:** by appointment in the professor's office starting on Monday, April 24, and ending on Monday, May 1.
- **Final writeup:** due via GitHub by 11:59 PM on Tuesday, May 9.

## 2 Project Topics

You first need to pick a project topic. You can talk to professor Rivas during his office hours about choosing a topic, and brainstorm with each other. There are typically two kinds of projects:

1. **DFA project.** This is the most common type of project: pick an application that interests you, and explore the best way to apply your DFA and Java skills to solve the problem. For examples and inspiration, you should look at your textbook chapters on DFA applications.
2. **NFA project.** This project is a little harder (unless you can convert it to a DFA). Pick a problem and develop a solution based on an NFA and Java. For examples and inspiration, you should look at your textbook chapters on NFA applications.

Great projects can come from students combining their interest in an application with things they're learning from this class. So it's good to choose something you're excited about. This is a good chance to start work on a research project.

Getting started and choosing a topic can be a bit difficult, so it's good to look around for other ideas as inspiration. Good places are talking with your classmates or your professor, and looking at published work in your area of interest. You could use Google Scholar for this particular task.<sup>1</sup> A good project writeup will have a great reading flow and will be coherent and will be of publishable quality.

**Please note the following:** projects will be evaluated based on:

1. The technical quality of the work. (That is, do the technical choices make sense? Is the approach reasonable? Are the proposed algorithms or applications clever and interesting? Do the authors provide a correct DFA or NFA diagrams? Does the code follow the style guidelines?)
2. Significance. (Did the authors choose an interesting or a “real” problem to work on, or only a small “toy” problem? Is this work likely to be useful and/or have impact?)
3. The documentation of the work: the clarity of the writeup, DFA or NFA diagrams, good analysis, good design, etc.

Note that the amount of code that you write and the time that you spend on the project are as important as your ability to do interesting or significant work and communicate it clearly. So pick an interesting project where you can actually make some progress.

## 3 Project Submission Details

Here are more details on submitting the different parts of the project. Please see the important dates at the beginning of this document for when each part is due. For all project submissions you will use the **same** repository you have been using and submit your work in a folder named “prj/”.

### 3.1 Project Proposal

#### 10 points

Your project proposal should be a simple document (.pdf) giving the title of the project, and a 300-500 word description of what you plan to do. Please submit your proposal through GitHub, NOT as a normal email nor as an attachment to one. Submit your work in a folder named “proposal” inside the folder “prj/”.

### 3.2 Milestone

#### 20 points

This report should describe what you've accomplished so far, it must have the DFA or NFA diagram of your project, and very briefly state what else you plan to do. The milestone will help you keep on track. You should view it as an early draft of the writeup you will turn in at the end of the semester. Specifically, you can write it as if you're writing the first few pages of your final report, so you can use most of the milestone text in your final report. Please write the milestone (and final report) keeping in mind that the intended audience is Dr. Rivas or others familiar with formal languages and computability and Java. Thus, for example, you should not spend two pages explaining what a DFA is.

Your milestone report should be at most 3 pages long. Please submit the milestone through GitHub, NOT as an email attachment. Please submit your milestone in .pdf format. Submit your work in a folder named “milestone” inside the folder “prj/”. The name of the .pdf should be formatted as “lastname-ProjectTitle.pdf”, where ProjectTitle is the abbreviated title of your project and lastname is your lastname.

---

<sup>1</sup><https://scholar.google.com/>

As you write the milestone and final report, please pay attention to follow a similar format of the DFA and NFA diagrams seen in class and in your textbook. In particular, many experimental papers have the following structure:

- **Abstract.** A brief overview of the paper.
- **Introduction.** Describes the motivation of this work and outlines the rest of the paper.
- **Detailed System Description.** Describes what the system does and how specific users interact with it. It also describes in detail what the DFA does.
- **Requirements.** Describes what is the physical requirement of the system resources.
- **Literature survey.** Describes what else is “out there”, what other similar systems exist and how yours is different.
- **User Manual.** Briefly explains how the system is used and draws some guidelines for proper use and error prevention.
- **Conclusion.** Summarizes the paper and the project’s significance.
- **References / Bibliography.** Gives properly formatted references to other scholarly work that this work is built on. Note that the references should be scholarly, which means things like refereed conference and journal articles. Importantly, that rules out things like most websites, basic textbooks, and press articles. So, unless your project was research-oriented, I anticipate this section being very small.

### 3.3 10-Minute Demo

#### 30 points

You will give a 10-minute demo of your project in your professor’s office sometime during the last three classes of the semester. Each person is expected to give a 10-minute demo that shows-off their work. Ten minutes is not much time, so keep it brief. We will adhere to the 10 minute schedule so that everyone has a chance to present, and so that there is time for feedback after each demo. Your project must be fully functional by the time of the demo.

### 3.4 Final Writeup

#### 40 points

Final project writeups can be at most 8 pages long. Please submit your final writeup in a folder named “doc” inside a folder named “prj/writeup”. The name of the .pdf should be formatted as “lastname-ProjectTitle.pdf”, where lastname is your last name and ProjectTitle is the abbreviated title of your project. You will also submit any data you used/produced in a folder named “data” and all Java code produced inside a folder named code, especially code that will reproduce the exact results you claim to have achieved in your writeup.

The following is a tree-like directory structure of the desired organization for the semester project, where the root folder is your GitHub repository root folder:

```
cmpt440lastname/
├── prj/
│   ├── proposal/ ... This directory will have the proposal
│   │               .pdf files.
│   ├── milestone/ ... This directory will have milestone
│   │               .pdf file.
│   └── writeup/ ... This directory will have final
│                   writeup .pdf file (in doc/) and
│                   two folders where you will have any
│                   databases (in data/) and Java code
│                   produced (in code/).
│       ├── doc/
│       ├── data/
│       └── code/
└── hw/
    ├── 0/
    ├── 1/
    ├── 2/
    └── ...
```

## 4 Expectations

The semester project is a very challenging project that can be done individually if time is managed correctly. My recommendation is that you work responsibly. I will not allow team projects.

Undergraduate students are expected to produce high quality work, worthy of a Marist student. After all, this is an elective course and you are in it voluntarily. Do nothing less than your best and you will do just fine.