Final Project Requirement Sheet 100 points

Project Goals

- Summarize all of your course knowledge in one significant coding project.
- A significant portion of your final course grade (15%) is the project component. For your project, you will work as a team or some cases (individually). A project team can have a maximum of 3 members.

Project Details

There are a total of six tracks for the final project, namely, Track 1, Track 2, Track 3, Track 4, Track 5, and Track 6. Of course, if a student would want to pursue a completely different track, this option is also available. Consult with and get the Professor's approval, if you like to develop an idea outside the first five tracks.

It is required for all students to follow the honor code. Some important points from the class honor code are outlined below for your reference:

1. Students are not allowed to share code files and/or other implementation details outside their team. It is acceptable to have a healthy discussion with your peers. However, this discussion should be limited to sharing ideas only.
2. Submitting a copy of the other team's program(s) is strictly not allowed. Please note that all work done during this project will be an opportunity for team members to learn, practice, and master the materials taught in this course. By doing the work individually, and within their team, students maximize the learning objectives.

At any duration during and/or after the project, students are recommended to team up with the Professor and/or the Technical Leader(s) to clarify if there is any confusion related to the items in the project sheet and/or class materials.

Project Track 1: New Language

- Design and implement your own programming language. It can be very small but it must contain and specify all of the components pertaining to a programming language, and have some purpose/application behind it.

Project Track 2: Combination

- Select a realistic problem and implement a solution to it using a combination (at least two) of programming languages. The languages must interact with each other in some way. You are allowed to make use of tools that help bridge the gap between languages. You need to be able to justify how mixing languages is beneficial for the problem you have chosen.

Project Track 3: Functional Programming

- Select a realistic problem where a functional language implementation is beneficial, and provide a solution to the problem in a functional language (e.g., Lisp, Haskell). You need to be able to justify why a particular functional language is useful for the problem you have chosen.

Project Track 4: Logical Programming

- Select a realistic problem where a logical language implementation is beneficial, and provide a solution to the problem in a logical language (e.g., Prolog). You need to be able to justify why a particular logical language is useful for the problem you have chosen.

Project Track 5: Translation

- Select two languages and write a translation program that takes a program written in one language and translates it into another language. It does not need to be very complicated or sophisticated, you may choose which aspects of programming languages you want to incorporate and which you want to avoid. Select existing programs or write your own as test cases; your translator should be able to translate at least 500 lines of code - combined from your test cases.

Project Track 6: 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 programming languages. After receiving the course instructor's approval for your idea, you will complete the project and report on your results.

If you're completely stumped in coming up with a project idea, you can certainly talk to me and we will set up a brainstorming session. Be creative and choose something interesting to you!

The course project must have a significant implementation part where you will develop and expand on the course topics covered this semester.

The course project must be extensive enough to qualify as a project (think of work for at least 3 to 4 one-week lab assignments), but not too extensive so that you cannot finish it in the remainder of the semester.

Timeline and Deliverables

The timeline and deliverable details are provided below for your reference:

1. Proposal – Start developing an idea for your final project. Write a 1-page technical report (single or double spaced) of what you propose to do in your final project and submit a PDF copy of your proposal through the GitHub link shared. I don't expect the proposal to be very detailed at this point. But, it should summarize what project you are going to pursue, what you want to do (the real low-end problem you will tackle, how you plan to solve this low-end computing problem, and at least a couple of references to indicate that you have done some research about the problem. **Deadline:** November 19th, 2021, 2:00 PM.

2. Progress Report – Start developing a 5-page technical report (single/double spaced) to document the progress done by the team. By this point, you should have made a good amount of progress towards implementing your project. Were there any unexpected challenges? Did you have to change your initial model/framework or the project skeleton code? Include everything you have done so far in your progress report, even if it is incomplete. No need to include the actual code (unless you want my help with it), just describe what progress you have made with it. Submit a PDF copy of your progress report using the GitHub link shared. **Deadline:** December 3rd, 2021, 5:00 PM.

3. Presentation – Teams can present their course project by recording a 10 minutes video to virtually present their course project. It is expected that students use Slides during their presentation. The deadline for this part is December 13th at 5:00 PM EST. The link below may be used to do the recordings. OBS is another option to record videos.

   https://www.apowersoft.com/free-online-screen-recorder

   By the presentation session, you should have finished implementation, run some basic testing, and done some code overview. In the presentation, you should describe the motivation, problem definition, challenges, approaches, and results and analysis. Use diagrams and a few bullet points rather than long sentences and equations. The goal of the presentation is to convey the important high-level ideas and give intuition rather than be a formal specification of everything you did. Design at least 6 to 10 slides, including a slide with the title of your project and your name. Also, it is required to show a short demo of your tool at the end of the presentation. Team members should contribute equally to the presentation. You may upload one video with an edited version of the different team members presentation or upload separate videos. The presentation slides and video should be uploaded to the following google driver folder.

   `https://drive.google.com/drive/folders/1u3PYFHg4WAQhVzcZZ4w9jUwbYqLK7w0C?usp=sharing`

4. Final Report – Start developing a 8-page technical report (single/double spaced) to summarize the technical details of the tool developed by the team. The final report should be clear and well written, which includes no typos or grammatical errors. The report should be written professionally and technically. The deadline for this part is December 13th at 5:00 PM EST. The report should include the following:

   - The motivation for your project. Why is the problem you decided to solve important or useful?
   - Background of the proposed problem. What have others done for it already? Include references.
   - Detailed project overview, a summary of the proposed implementation, and the methodology used. Include pseudocode, diagrams, and examples if appropriate. If you are extending existing work, briefly describe previous work and include references to it.
   - Conclusion. Give a short overview of your project and its results. Describe what you learned, what were the biggest challenges, and the biggest rewards.
   - For each deliverable, you need to submit a PDF with your report (or presentation slides). For your final report, you need to submit any supplementary material (code, data, a README file documenting what everything is, and how to run your program) to the git repository using the GitHub link.

Grading Rubric

1. Proposal – 10 points
2. Progress report – 20 points
3. Presentation – 35 points
4. Final report and implementation – 35 points
5. Please make sure to include the honor code statement in all submission files.

6. If a student needs any clarification on their project credits, it is strongly recommended to talk to the Professor. The project credits may be changed if deemed appropriate.