The goal of this assignment is to develop, within a team of students, an application related to programming language design. The deliverables and due dates for the project are as follows.

- **Feb 28**  Project proposal due
- **Mar 21**  Revised proposal due
- **Apr 9**  Status update due
- **Apr 23**  Status update due
- **May 2**  Presentation, source code, peer evals due

Each of the deliverables is further described below.

**Project Proposal (Initial draft due Feb 28, Revised version due Mar 21).** To complete your initial proposal, you must do the following steps.

1. Assemble a team of three students. Note that Section 1 will require two teams of two students and seven teams of three students. Section 2 will have three teams of three students.

2. Select a topic. A list of potential topics are given below.

3. Define the scope of the project (what will be accomplished).

4. Write up a proposal describing each of the above items.

The following is a list of possible topics for your project. For your initial proposal, your team must pick a topic and define the specifics of what you plan to accomplish by the end of the semester. You will have a chance to revise your initial proposal in your final (revised) proposal.

- **Extend MyPL to include a new language feature.** Examples of possible extensions might include: adding a “module” system and a standard library; adding classes (with member variables and methods); adding garbage collection (for structs); adding higher-order functions (i.e., being able to pass function names as arguments); or improving the type inference capabilities for functions (e.g., so that function return types can be fully inferred). Note these are just examples, and there are many other possibilities for extensions.

- **Create a back-end compiler for MyPL.** For this project, you should consider using the LLVM framework or compiling directly to bytecode (e.g., for the JVM). Note that getting a small subset working for MyPL would likely be sufficient for this project.

- **Benchmark and profile your MyPL implementation, comparing it to different languages.** Benchmarking would require creating a set of programs to use that progressively stresses the language implementation and then recording execution results. Profiling would involve analyzing the current implementation and determining where performance issues exist. A number of profiling tools exist for Python, including built-in libraries (e.g., see [docs.python.org/3/library/profile.html](http://docs.python.org/3/library/profile.html)).

- **Define and implement a domain-specific language.** A domain-specific language (DSL) is a small or simplified programming syntax designed for a specialized purpose. Examples
of DSLs include sed and awk, bc and calc (calculator programs in linux), make (and other build systems), CSS, XSLT, and so on. If you choose this option, you must implement an interpreter for your DSL using steps similar to those for MyPL (including lexical, syntax, and semantic analysis). Your DSL must also involve some calculation/computation component.

- **Learn a new language.** This would involve exploring and learning a programming language that is new to you, creating an application using the language, and developing a “mini” tutorial. The language you select should be considerably different than those discussed in class or that you already know. Examples could include Rust, Scala, Erlang, Smalltalk, Javascript, F#, OCaml, R, or Ruby. Your application must demonstrate the main and/or innovative constructs in the language you select.

**Status Updates (April 9 and April 23).** Once your proposal is approved, you should begin working on your project (spreading the work out over the rest of the semester). You must provide a one-page status update twice, one on April 9 and one on April 23. Each status update should describe the progress you have made as a team, the contributions made by each team member, and the work remaining.

**Presentation and Final Submission (May 2).** Each team will present their project results on May 2. Your team will have at most **6 minutes** for their presentation. Your presentation must include:

1. The goals of your project (what you were trying to accomplish)
2. What you accomplished
3. A demo of your work
4. What you would do given more time
5. What you learned by doing the project

Each team member should speak for approximately the same amount of time (i.e., for about 2 minutes). Your presentation must be well-rehearsed and high quality. In addition to the presentation, you will need to turn in a hard-copy of your source code, print outs showing your program works correctly, the test cases you used, your presentation slides, and a completed peer evaluation form for each team member (which I will provide later in the semester).

**How you will be graded.** Your grade will be determined based on the following criteria.

1. Quality of your implementation/project work
2. Completeness of your implementation/project work
3. Difficulty level of your project
4. Quality of your presentation
5. Your contributions to your team

Please be sure to ask me about project requirements and expectations if you have questions.