You will complete one large project that asks you to design, implement, and evaluate your own domain-specific language. Though the project will be due at the end of the course, I will provide milestones and guidelines throughout the course, to help you stay on track. Each milestone is graded separately. More details about each milestone and their due dates will become available when we start working on the projects. For now, here is a brief description of each milestone, roughly when it will be due, and how much each milestone contributes to your project grade:

- ***Design notebook entries (10%):*** **You'll keep a notebook where you write and reflect on how your project is going. At least one entry will be required before each individual meeting.**
- Critiques (10%): You will write critiques of your classmate's work, throughout the project. At least one critique will be required before each individual meeting.
- **Motivation for DSLs (10%): A relatively brief paper about DSLs and why they are useful. This paper will be original, written by you, cite existing sources, and be based on the material we learn during the first part of the course. This paper will be due before our first individual meeting.**
- Project description and plan (5%) + video-conference peer review (2.5%): Your description of the project and your schedule for completing it. This milestone will be due before our second individual meeting. You will also meet with all the students in the class to discuss your projects and plans.
- Language design and implementation overview (10%): A snapshot of your evolving design for your DSL and your strategy for implementing it. This milestone will be due before our third individual meeting.
- Prototype (10%) + video-conference peer review (2.5%): A version of your DSL that others can use. This milestone will be due before our fourth individual meeting. You will also meet with all the students in the class to discuss your prototypes.
- Preliminary evaluation (10%): Your assessment of how well your DSL matches its design goals. This milestone will be due before our fifth individual meeting.
- Final product (15%): The last version of your DSL, for this course. This milestone will be due by the end of the course.
- Final paper (15%): A document that describes and reflects on your project. It will incorporate, with modifications, many of your previous milestones. You can expect it to be 8–10 pages long. This milestone will be due by the end of the course.

Motivation for DSLs (10%): A relatively brief paper about DSLs and why they are useful. This paper will be original, written by you, cite existing sources, and be based on the material we learn during the first part of the course. This paper will be due before our first individual meeting.

Some users may have some specific special needs regarding to the field they work in such as the investment system. However, there were many concepts and formulas that the program developer does not understand. The barrier created between the user and developer leads to the creation of the DSL. DSL, Domain Specific Language, is a language designed for specific purpose within certain field, or domain. Compared to the most common languages like JAVA and C++, DSL only supports limited features but with comparatively simple expressions, in which provides more friendly user experience. The implementation of the DSL will improve the efficiency for the people working in the field with limited programming knowledge. Because of the DSL, those domain experts can now benefit from the progress of programing and therefore, catalyzes the progress of scientific discoveries. In general, DSL has three main categories: internal DSLs, external DSLs and language workbenches.[1] The internal DSLs are made from some major languages such as JAVA while the external DSLs are made from zero. DSLs have many advantages compared to major languages in the way it presents: the code is clear and understandable; it is easy to use and amend. In the meantime, DSL.have some disadvantages which mainly the result of its domain: the expression, as well as the function, is limited.

[1] Fowler, Martin, and Rebecca Parsons. *Domain-Specific Languages*. Addison-Wesley, 2011.