CompSci 390 Final Project

In the final project your **3-person project group** will:

1. Pick an “interesting” language, not taught in the class
2. Do a 30-50 minute video on one feature of that language
3. Build a project in that language that highlights the strengths/unusual features of the language you’ve chosen

## Making Your Group

This class should divide evenly into 5 groups of 3 – so that’s what we’ll do. To see who’s available take look at the team assignment spreadsheet here:

<https://docs.google.com/spreadsheets/d/1HI8ScoDg1i1mFtUPYxkCn1YM7gCKZhVgnHK7RnBzXBc/edit?usp=sharing>

Advertise your interest in particular languages on Piazza.

If you have a tentative team and language picked, feel free to more your team members into one of the groups on the Spreadsheet. Choices are not final however, until I approve your team. I won’t start the approval process until Tues 3/16 – starting on that Tuesday you can get me to approve your team by having a team member ask me in my office hours. Broadly speaking, generally will approve your team if:

1. Your team is the right size and you’ve gotten the agreement of every member
2. The language you picked is appropriate (see below for details on that)
3. There’s no other tentative teams with your same language on the spreadsheet

If there are multiple teams that want the same language we might have to work out a conference call to resolve the dispute. Generally you can save yourself a lot of annoyance if you just find a new language if you see that someone is trying to claim the one you want on the spreadsheet.

The end of team/language formation is 3/19. After that I start manually assigning people to teams and picking languages that I think are cool.

## Choosing Your Language

You can choose and programming language you want, with these restrictions:

1. It should be unusual in some way. Ruby, Java, C# - these are all very mainstream languages with features that (for the most part) are not very different from each other and would not make a good choice. One place to start looking is the other languages in your textbook (or maybe its sequel, *Seven More Programming Languages in 7 Weeks*) – but there are many interesting languages out there.
2. It should not be a joke language (e.g. [Whitespace](http://en.wikipedia.org/wiki/Whitespace_(programming_language))) – there should be a community of people who genuinely think this language is a good idea at least for some purposes.
3. It should be unique in the class & I must approve it (see above)

When you have picked a language you should get it informally approved by me. Then go on to do the official proposal (described at the bottom of the document).

# The Video

You will do a video on a feature of your language that will be shared with the class. This should about one normal lecture in size (30-50 minutes – but it should be a very solid 30 minutes), and every team member should in some way participate. The presentation should be well-rehearsed (with notes, slides, demo programs), clear, and interesting.

Because the class will only get a 1-day introduction you should try your best not to just give them a “hello world” talk but introduce them to some specific and interesting feature of the language. Seriously, jump right in here and just say “hey we’re talking about feature X in language Y.”

It is understandable that they will not get a full flavor of the language in 50 minutes but try to do something memorable that will give a real feel for the language’s strengths/uniqueness.

You do not have to demo your project as part of the class presentation – do it only if it is educationally useful.

Please host your video on duke’s warpwire - <https://launch.warpwire.duke.edu/> (basically youtube but Duke specific). The due date is on the course schedule.

## Grading the Class Presentation (35 points)

|  |  |
| --- | --- |
| Polished and appropriate materials – demos/slides/notes slides clearly illustrate the points made and are well designed for their purpose | 10 points |
| Presentation is clear and interesting with one well described educational purpose | 10 points |
| Topic is well chosen to be achievable in one lecture, interesting, and achieves its educational goals | 10 points |
| Good division of labor between presenters with everyone contributing approximately equally | 5 points |

# The Programming Project (65 points)

You also will do a medium/large scale project if your choice in your language of choice. The project should be a task for which your chosen language is well suited. This project should be at minimum as large as the “medium projects” you have done in class – it could be a little larger. Every member must participate – so you should give some thought to how to divide things up so that everyone is doing something interesting.

## The Programming Project Proposal

You will formally propose a programming project in writing to me. You should:

1. Briefly describe the project in general terms and why your language is well suited for it
2. Give a breakdown of the 65 points of the project into functionality areas like I do on my assignments. Make sure it is clear what you require for at least a majority of the sections – this is going to be my approximate “rubric”. You can have one “random cool ideas as we decide later” section (not required).
3. Note who is going to work on what sections of the project – pairs of people are fine if you are actually pair programming, not if you’re just too lazy to decide. This is not set in stone once you start, but I want to see you’ve at least got a plausible division of labor that ensures everybody does something.
4. The proposal due date is on the schedule.

## Grading the Project

I will probably stick approximately to the rubric you have given me. The whole team will get the same grade unless there’s some inter-team conflict or someone does not do their work. If there is a conflict:

1. Address it with me before it delays the overall success of the project
2. Address it in a professional manner that does not involve ignoring team meetings/emails/phone calls or going off in a huff and implementing the whole project yourself.

## Submitting the Project

You will use source control for this project. You can use duke’s gitlab (gitlab.oit.duke.edu) or github or some other public hosting. Do not use something without change history (e.g. a shared dropbox). You are required to commit code under your own name, using your own account. In case of a conflict, I will look to source control to determine who wrote what code when. Your project’s final code due date is your demo date (see below).

You will demo the project for me live via Zoom. **Check the schedule for when demos will start, though we will work out an official demo time for each time closer to the event.**