Name: _____ Date:_____

Term Project Requirements

Important Dates

Date	Event
2/13	Project Proposal deadline
2/20	Midpoint Status Report Update
3/6	Optional – Test run and preload your software configuration on the instructor computer and test with the projector.
3/11 & 3/13	Class presentations
3/15	Last day to submit source code and documentation

Introduction

For your term project, you will pick a language that you **do not already know**, learn the language, and develop a project using that language. You can work individually or with a partner. I encourage you to work with a partner.

You can choose from the following languages. If you would like to use a language not on this list, check with me for approval as soon as possible.

- Ada
- ALGOL
- Clojure
- COBOL
- Dart
- Delphi
- Eiffel
- Elixir
- Erlang
- Forth
- Fortran
- F#
- Go
- Hadoop
- Haskell
- Lua
- Kotlin
- Oz
- Pascal
- Perl
- PHP
- R
- Ruby (With Rails)
- Rust
- Scala
- Scheme
- Simula
- Smalltalk
- Snowball/SNOWBOL
- Visual Basic

You might notice there are languages in the list that we have already covered in class. I have added them because I felt we could expand more on them. If you choose to pick those languages, then my expectations for a project will be slightly higher and a simple project won't cut it.

Reserving your Language

Language selection will be first come first served. To guarantee your language, reserve your spot early. You can declare your language and team on the Canvas pinned forum post. I will update the main topic official list so if you declared a language and don't see your name updated on the main post, let me know.

Proposal (25 Points)

Before you get started on coding your project, you will need to write up a proposal to discuss the project idea and define the scope of your project. The scope will consist of a list of features that your project will implement and identify the completion point of the project. This list will be used to identify how complete the project is at the point of submission and will be used for grading. If you feel the scope has changed during the implementation of the project, let the instructor know as soon as possible so the list of requirements can be updated to meet with the project's new specifications.

Write an "elevator pitch" for your project. The project proposal must give the following:

1) Chosen language
2) Brief statement on the strengths of the language and/or libraries and frameworks you wish to use:
    a. What application domain was it intended for?
    b. What programming paradigms does it primarily support?
3) Description of your project that covers the following:
    a. What program are you going to write?
    b. **Mandatory requirements:** What are its primary features? What do you want it to do? This should outline the minimum scope needed for the project to be successful. I will use these requirements as part of my grading criteria.
    c. **Optional requirements:** If you have extra time, what are some additional features you would implement?
4) Language fit: why is your language a good choice for this particular project?
5) Resources: what do you need for a development environment and how can you get that environment?

I expect you to use proper English in your proposal. The quality of your writing and formatting must be such that you would not be embarrassed to present this to your manager.

When I receive your proposal, I will meet with you or your team to discuss the scope of your project. If it seems too large, I will encourage you to scale back. If it is too small, I will encourage you to pick something bigger. Aim at something two to three times as big as a weekly lab.

The proposal is due by the end of the week 6, but the sooner that you submit it so I can review and provide feedback, the more time you will have to work on your project.

Project (100 Points)

Develop a project using your language that shows off some of its major features. The project should be appropriate for the language.

Points will be awarded based on the scope of the project (harder projects may receive extra credit), quality of the implementation, and whether the project actually does what it's supposed to.

Backup project

If you have looked around and still need ideas for a project to do in the lab, you may consider doing your project in the form of a language introduction lab assignment. Your lab manual must include the following:

- An introduction to the lab that discusses the scope and learning objectives.
- Instructions for setting up and configuring any runtimes, tools, IDEs, dependencies, etc.
- A detailed lab exercise that introduces the language and highlights some of its distinctive features:
    - This should show off the language's strengths or something that is drastically different about the language's programming paradigm or approach to solving a particular kind of problem than other languages you already know.
    - If your lab exercise revolves around a specific library or framework:
        - Discuss the problem domain that the library/framework is meant to help with.
        - Ensure that someone who goes through your lab exercise still has an opportunity to explore the target language a bit. Try to stretch yourself and challenge the person who is running through your lab.
    - If your lab has any starter code, you will submit that in addition to the completed lab code.
- Your work must be your own! Do not simply copy/paste or re-implement someone else's tutorial.

Write-up/Presentation (100 Points)

Write-up

You will need to do a write-up on your language. The write-up must include the following:

1. Language name, origin/history, and design goals; and development tool availability
2. Programming paradigm(s) you used in your project. Is the language statically typed, dynamically typed? What data representations are available? Is the language compiled or interpreted?
3. Discussion of language design criteria based on the material in Chapter 2 of the text. (Pearson)
4. Lessons learned while using the language on your project

I expect proper English and professional formatting for your report. The report is due by 11:59 p.m. on the Sunday of dead week.

Presentation

You will also need to give a presentation of your project. Your presentation should focus on the following:

1. Language features: what makes your language different from others
2. Project demonstration: show-and-tell
3. Lessons learned during development: How did your language make your job easier or harder vs. doing a similar project in another language (such as C++).

Your presentations should be 5-10 minutes long. If you take less than 5 or more than 15 minutes, I will take points off.

Points will be evenly divided between your write-up and presentation. Presentations will take place during dead week both during the labs and class time.

For the demonstration, you will have the choice to use the instructor computer to run your application to display on the projector or you can connect your personal computer to the projector.

Note: If using your personal computer, make sure it has the ability to connect to a 15 pin VGA connector.

Note: If using the instructor computer, please come to class early to provide enough time to go through a dry run to solve any unforeseen problems so as to not hold up the rest of the presentations.

A test run of your presentation using the projector or instructor computer would be ideal to do on the lab day prior to the presentation day.

Reserving your Presentation Day

Depending on the number of groups we have, we have a very limited amount of time for presentations. You will have a choice to present either on Lecture day or Lab day of dead week. You can select which day you want to preset on in the Canvas forum post.

What To Submit

Everything will be submitted via Git in Azure DevOps. You will create a new term project repository in your organization and commit all your material there. If you are working as a team, one of your teammates will host the repository in their organization and invite both your partner and myself to the team.

- Project
  - All source code.
  - Any specific build scripts or important files required for compiling
  - A README file that provides instructions on how to build.
  - Lab manuals or documentation if doing the backup project
- Write-up
  - Midpoint Status Update Word Document
  - Project Write Up Word Document (see previous section)
- Presentation
  - Any presentation materials
    - Notes
    - Slides
    - Demonstration code

Grading Rubric and Information

Proposal

Metric	Description
Proposal (25 points)	Implemented all the proposal requirements as specified in this document.
Check off (Mandatory)	My stamp of approval of scope.
Midpoint Status Update (25 points)	Status Update letting me know of your progress on the project.
Total: 50 points	

Project

Metric	Description
Requirements (40 points)	Implemented all the project requirements as specified in this document. Implemented all the functionality as described and agreed upon during the proposal.
Documentation (10 points)	Readme document containing all the project details to build and configure as described in this document.
Source Code (50 points)	Source code compiles with no errors and is sufficiently commented. Code implements all features defined in project scope. A minimum amount of error checking and exception handling preventing the code from crashing from erroneous user input.
Total: 100 points	

Presentation / Write-up

Metric	Description
Documentation (50 points)	Word document containing all the project details as described in this document.
Presentation (50 points)	The presentation introduced the project's purpose and described the functionality. The demonstration correctly displayed the intended output of the project.
Total: 100 points	

Grand Total: 250 Points