

Assignment 2

For this assignment, you will be exploring a new programming language. The basic task is to evaluate various features of the language, using the concepts you have learned in the course.

Language Choice

You can choose one of these languages as the topic of your assignment:

- › Clojure (<https://en.wikipedia.org/wiki/Clojure>)
- › Erlang (https://en.wikipedia.org/wiki/Erlang_%28programming_language%29)
- › Julia ([https://en.wikipedia.org/wiki/Julia_\(programming_language\)](https://en.wikipedia.org/wiki/Julia_(programming_language)))
- › OCaml (<https://en.wikipedia.org/wiki/OCaml>)
- › Scala ([https://en.wikipedia.org/wiki/Scala_\(programming_language\)](https://en.wikipedia.org/wiki/Scala_(programming_language)))

You should learn the basics of your chosen programming language. I suggest you start with a decent tutorial for the language: most languages have one or two. Watch for tutorials for programmers: some will have tutorials for beginning programmers, which isn't you.

Report

Once you have learned the basic concepts of the language, write a 4–5 page report summarizing its features (page count excludes diagrams/code examples; standard formatting, single-spaced). This should be a generally well written essay-style report.

The report should summarize the features of the language using topics from the “Language Features” part of the class (as appropriate). You should also discuss how this language compares with other languages you have used.

You should think of the overall goal of your report as explaining the language to average member of this class, so that they can get a better idea of what this language is good (and bad) for.

To give you an idea of what kind of information should be included, I have provided some [sample outlines for the report](#).

Exercises

You should also create re-implementations of some of the exercises, to help you compare (at least) your chosen language and Haskell. You can ignore any implementation details required by the exercise (like “recursive” or “using `foldl`”): just create a function/method that calculates the same result. Write implementations of these functions:

- › **Exercise 2:** the `divisors`, `primes`, `join`, and `pythagorean` functions.
- › **Exercise 3:** `mergesort` (and thus probably also `merge`). For `mergesort`, be as general as you can with one implementation in your language: don't worry about being able to sort every type if that's difficult.
- › Also **Exercise 3:** `isPrimeDay` and `isFriday`, using whatever date types are available in your language

Of course, you should adapt the data types and argument/return values as appropriate to your language.

Provide a main function (or whatever the code that runs first is called in your language) to demonstrate calling these functions/methods.

Sample Programs

You should write some additional sample programs in the language. These programs should demonstrate the critical features of the language as well as its strengths.

These programs are up to you. They do not have to be large projects: you should have probably 3 or 4 examples that are a little more involved than a weekly exercise question. Try to provide at least one example of a problem that your language solves better (faster, more elegantly, etc.) than Haskell or C++. Look for another where it's worse.

Of course, these programs should be your work, not examples from tutorials or Wikipedia. Please comment your code well enough that we can figure out what it's supposed to be doing.

Submitting

As with any work submitted at University, the report and code should be your own work.

Submit your files through CourSys for [Assignment 2](#). For the report, you must submit a PDF file. For the code parts, a ZIP archive containing your work.

Updated Tue Oct. 27 2020, 22:04 by ggbaker.