CS 524 Graduate Project Outline

1. Title Page
2. Introduction to Elm
   1. Design and History
      1. Elm was created in 2012 by Evan Czaplicki. Why? This topic will look into Evan’s purpose for the language, as well as how he designed and implemented it.
   2. Basic Features
      1. Elm is a web development language; consequently, it shares some features. Elm has some unique features that offset it. This topic will describe the primary features that Elm has as well as what type of language it is.
   3. Domain of Use
      1. Elm is a web language, as said before. This topic, however, will discuss primarily where Elm is used. Where on the web is Elm especially useful?
3. Elm Characteristics
   1. What Need Does Elm Fulfill?
      1. For Elm to be created, Evan must have seen some need that other languages just couldn’t fulfill. Why does Elm exist? This topic is an elaboration of the general introduction of Elm’s purpose.
   2. Comparison to Similar Languages
      1. Since Elm compiles into Javascript and is used for web development, this section will primarily focus on comparing Elm to Javascript, HTML, and other web languages to demonstrate what distinguishes Elm.
   3. Analysis of Elm
      1. The analysis portion is simply looking at Elm as a language and breaking down each component: the language’s lexemes, syntax, grammar, semantics, etc. This section is going to go over some of these components to fully define Elm as a programming language.
   4. Developing with Elm
      1. How and where would one develop with Elm? In what method(s) is Elm implemented? The main point here is to give a general example of developing with Elm to not only demonstrate Elm’s usefulness, but to also give context and an understanding as to why Elm is used for web development.
      2. Snippets of code along with some explanations will be the primary medium for this topic.
   5. Quintessential Unique Feature
      1. What feature sets Elm apart? How useful is this feature, and is there anything similar in other languages? What does this feature do that requires it to be a separate entity? If this feature is so important, why don’t other languages have this feature?
4. Evaluation According to Cutsem’s Criteria
   1. Features that Contribute to Readability
      1. A list of Elm’s important features along with several sentences worth of analysis and commentary will be provided here. Each feature listed will be evaluated based on the standards of Readability provided in the textbook, and in class.
   2. Features that Contribute to Writability
      1. In the same vein as the previous major topic, this section will break down some features that greatly contribute to or greatly diminish the writability of Elm. Some of the features listed here may be the same as in the Readability section if the feature is important enough to impact both aspects.
   3. Abstraction
      1. In accordance with Cutsem’s paper, this section will analysis Elm’s ability for language abstraction; the extent of which abstraction played a role in Elm’s creation will also be explored.
   4. Paradigm
      1. How much of a paradigm shift, if any, did Elm induce upon its invention? Does Elm build upon that “path of least resistance” mentality? What features contribute to this, if any?
   5. Simplification
      1. Like some of the languages before it, does Elm reduce the complexity of some constructs? What features are generally considered to be simplifications, and if these features exist, were these features in mind when the language was being designed? Did they contribute to its purpose?
   6. Enforcement
      1. The last but not the least of Cutsem’s criteria for language creation. How well, if at all, does Elm *enforce* pre-existing ideas? Elm shares some common features with other similar languages; with this in mind, does Elm reinforce these features through its own interpretation, or does it have other ways of implementing the same idea?
   7. Judgment Consensus
      1. With all these things considered, in my own opinion, is Elm a *good* programming language? Does it satisfy enough of these criteria to be considered of great value? This analysis follows the guidelines in Cutsem’s paper for thoroughness.
5. Conclusion
   1. Closing Thoughts, Final Judgment
      1. As the title of this suggests, I will offer my final thoughts on the matter, as well as giving my two cents on whether or not I find Elm to be useful and simple to learn.
6. Bibliography
   1. All sources used, whether they be textbook, Internet, lecture slides, etc. will be listed here in alphabetical order. All in-text citations will be linked back to this bibliography.

With the conclusion of this outline, I would like to say that I greatly anticipate the completion of this project, as Elm has been fun to work with so far. Please mark this outline up as much as you would like and suggest improvements where necessary. I tried to maintain the correct balance of generality and specificity when providing details, so I hope this suffices.