- **Everything written in black color font stays there. Do not delete them. In addition, write your report using black color font.**
- **Everything written in blue color font are instructions to help you understand what to write in those sections. Delete them and replace them with your text as needed with black fonts.**
- **Everything written in red color font are examples. These should be deleted as well and replaced by your text using black fonts.**

Department of Computer Science
CS 298/299 Proposal
Title: write a suitable title for your project. JestScript - A Whimsical Dive into Humorous Programming

Write your name John von Neumann

Advisor
Write your advisor's name and email address Dr. Donald Knuth, advisorEmail@sjsu.edu
Committee Members
Write your committee member's name and email addresses Dr. Edsger W. Dijkstra, member1Email@sjsu.edu, Dr. Claude Shannon, member2Email@sjsu.edu

Introduction

- This section includes two to three sentences that introduces the general background of your research project.
- You should write general introduction to a broader field and a specific field of research.

In the realm of code seriousness, JestScript emerges as a playful endeavor, infusing the traditionally stoic world of programming with humor. Motivated by the desire to bring joy to coding, JestScript presents a unique take on syntax, error messages, and debugging experiences.

Keywords:

- Write up to 6 keywords to help identify/classify your report.

- Bear in mind that keywords help others to access the report and improves its visibility. *programming language, enjoyable coding, programming language design, user experience, light-weight programming language*

Problem Definition and Motivation

- This section highlights the motivation of your project and clear and concise definition of your project.
- The motivation sentence should clarify why an audience should care about your project.
- The problem statement clearly sets a scope of the project.

The motivation behind JestScript lies in challenging the stereotypical perception of coding as a dry and serious task. With a plethora of programming languages prioritizing functionality over fun, JestScript aims to provide developers with a light-hearted alternative, fostering a more enjoyable coding atmosphere.

Results Achieved in CS 297

- List a few main results from CS 297.
- Write a very brief description of each item.

• JestScript Compiler: The JestScript compiler that translates JestScript code into executable instructions for the target platform

• JestScript Syntax Guide: A detailed guide documenting the JestScript syntax, language rules, and conventions for developers.

Expected Deliverables in CS 298/299

- Define a few main deliverables in your CS 298/299.
- Each item should be a tangible product not a vague action item.
  - [Bad example] Think about the characteristics of my new programming language
  - [Good example] Syntax Guide: A detailed guide documenting language rules and conventions for developers.

• Open-Source Repository: The public repository on a platform like GitHub containing the JestScript source code, documentation, and issue tracker.

• User Feedback Analysis Report: A report summarizing user feedback from testing phases, outlining areas of improvement and potential future enhancements.

Timeline and Milestones for CS 298/299

- Define several key milestones in your CS 298/299
- Make sure that each milestone is well-defined so you and your advisor can track the project progress (and possibly measure the delay of the project)
- Plan the duration and timing of each milestone and create a Gantt chart

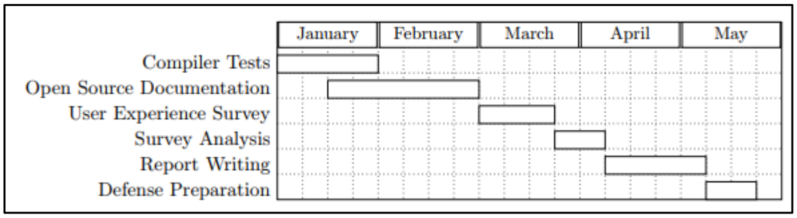

Table 1: Timeline of the project draw your timeline preferably in a Gantt chart and delete the above chart

References

- List the references here using a consistent format

[1] C. A. R. Hoare, “An axiomatic basis for computer programming," *Communications of the ACM*, vol. 12, no. 10, pp. 576{580, 1969.

[2] P. J. Landin, “The next 700 programming languages," *Communications of the ACM*, vol. 9, pp. 157{166, Mar. 1966.

[3] R. Milner, “A theory of type polymorphism in programming," *Journal of Computer and System Sciences*, vol. 17, pp. 348{375, Aug. 1978.

[4] G. Plotkin, “Call-by-name, call-by-value, and the λ-calculus," *Theoretical Computer Science*, vol. 1, pp. 125{159, 1975.

[5] J. C. Reynolds, “Towards a theory of type structure," in *Colloque sur la Programmation, Paris, France*, vol. 19 of *Lecture Notes in Computer Science*, pp. 408{425, Springer-Verlag, 1974.

[6] C. A. R. Hoare, “An axiomatic basis for computer programming," *Communications of the ACM*, vol. 12, pp. 576{580 and 583, October 1969.

[7] G. Morrisett, D. Walker, K. Crary, and N. Glew, “From System-F to typed assembly language," *ACM Transactions on Programming Languages and Systems*, vol. 21, pp. 527{568, May 1999.

[8] L. Cardelli, “A semantics of multiple inheritance," in *Semantics of Data Types* (G. Kahn, D. MacQueen, and G. Plotkin, eds.), vol. 173 of *Lecture Notes in Computer Science*, pp. 51{67, Springer-Verlag, 1984. Full version in *Information and Computation*, 76(2/3):138{164, 1988.

[9] L. Damas and R. Milner, “Principal type schemes for functional programs," in *ACM Symposium on Principles of Programming Languages (POPL), Albuquerque, New Mexico*, pp. 207{212, 1982.

[10] W. A. Howard, “The formulas-as-types notion of construction," in *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism* (J. P. Seldin and J. R. Hindley, eds.), pp. 479{490, Academic Press, 1980. Reprint of 1969 article.

[11] P. J. Landin, “The mechanical evaluation of expressions," *Computer Journal*, vol. 6, pp. 308{320, Jan. 1964.

[12] E. Moggi, “Computational lambda-calculus and monads," in *IEEE Symposium on Logic in Computer Science (LICS), Asilomar, California*, pp. 14{23, June 1989. Full version, titled *Notions of Computation and Monads*, in Information and Computation, 93(1), pp. 55{92, 1991.