IMY 320 Written Research

Overview

Every year new programming standards, techniques and languages are introduced to the mix of existing computer languages. Many of these are temporary fads, and some are becoming long-lasting trends. In recent years, WebAssembly has picked up popularity and is now one of four languages to run natively on browsers along with HTML, CSS and JavaScript. According to WebAssembly's official site, https://webassembly.org, the language was designed to enable high-performance applications on the Web.

For this assignment, you will create a short "Introduction to WebAssembly" article. The article should not only show the basic aims, goals and general information about the language but also a step-by-step tutorial on creating your first WASM application. The tutorial should demonstrate how developers can create programs using languages like C, C# or Rust (pick one) and compile these programs into WebAssembly.

Writing – introduction

The article should be targeted at 2nd-year students of your respective degrees (include your degrees within the document). The purpose is to…

- Give readers an overview of what WASM is.
    - You should cover as many corners of WASM as you possibly can, but for the sake of this assignment, you can leave out any security implications.
- What role WASM would play in their future?
    - Based on your research and degrees.
- How does WASM fit into the degree as a whole?
    - Refer to other courses or projects done throughout the degree to provide context.

This overview paragraph should not exceed 500 words in length excluding citations. For starters, take a look at WebAssembly's official site mentioned above and the WebAssembly Specification.

Tutorial

Select a programming language you are proficient in and work through a few existing tutorials to get a general understanding of the structure of WASM. The list of programming languages that WASM supports currently can be found on the WASM website. Using this newly found knowledge, create a tutorial by mixing and matching the elements you have learnt.

The tutorial does not have to be highly difficult but aims to give the reader an understanding of the language. This should include everything from:

- The initial setup process
  - E.g. installing and using emscripten
- Conversion/compiling process
  - E.g. explaining the process of wasm conversion
- The final publication (usage) of the product.
  - E.g. how everything will be called from the web side

*Note: This is a general outline.

You are welcome to implement any kind of framework and libraries as long as you explain within the tutorial their uses.

The final products should not be as plain as a "Hello World" program; be creative in your implementation and allow the user to create something interactable. For starters, you can take a look at Tutorialspoint's wasm tutorial or the online book Rust and WebAssembly, where they teach users how to create Conway's Game of Life.

Note: The final tutorial reflects how well you understand the language; each point you cover should contribute to aiding a reader's understanding. It is not sufficient to provide steps; you will need to provide justification or explanation on why certain steps are taken.

Push everything (code) onto a GitHub repository and share that as part of your submission.

Evaluation

The final grading of this assignment is not about how highly technical your document is or how impressive the final product is. You will be mainly assessed on how well you can convey your understanding of the language in a tutorial form, the details of your tutorial, and how well you cater the tutorial to the user.

Writing	This includes assessing the clarity of expression, coherence, and adherence to proper formatting and citation styles.
Purpose	Writing depletes the purpose of the tutorial and gives needs/reasons for the uprising of WASM.
Accuracy	The accuracy and completeness of the information given on WASM.
Explanation	Detail explanation of the code used (why, what, how) Explanation of how the different systems interact with each other to create your final product.

Completeness	Completeness of your tutorial. The reader’s level of knowledge after working through your tutorial.
Structure	The overall logical progression of the tutorial
Other	Sources, citations and style of the document

Submission

The final document should be submitted using clickUP in PDF format in the provided upload slot. You should cover as many corners of WASM as you possibly can, but for the sake of this assignment, you can leave out any security implications.