**CSE 310 – Applied Programming**

# Module Submission

Name: Gabriel Wahlberg

Date: 5/18/2024

Module # (1-3): 2

1. Copy the link to your public GitHub repository here: https://github.com/gawahlb/rust\_factorial\_calculator
2. Copy the link to your video here:
3. Mark an “X” next to the module you completed:

Cloud Databases

Data Analysis

Game Framework

GIS Mapping

Mobile App

Networking

SQL Relational Databases

Web Apps

Language – C++

Language – Java

Language – Kotlin

Language – R

Language – Erlang

Language – JavaScript

Language – C#

Language - TypeScript

X Language – Rust

1. Complete the following checklist to make sure you completed all parts of the module. Mark your response with “Yes” or “No”. If the answer is “No” then additionally describe what was preventing you from completing this step.

|  |  |
| --- | --- |
| **Question** | **Your Response** |
| Did you implement the entire set of unique requirements as described in the Module Description document in I-Learn?  List each requirement from the requirements document and mark if you completed it with a yes or no. | |  |  | | --- | --- | | All Basic Requirements | Complete | | 1. Variables | Yes | | 2.Expressions | Yes | | 3.Conditionals | Yes | | 4.Loops | Yes | | 5.Functions | Yes | | 6. |  | | 7. |  | | 8. |  | | One Additional Requirement |  | | 1.Data Structure (Vec) | Yes | |
| Did you write at least 100 lines of code in your software and include function level comments on all the functions you wrote? | Yes |
| Did you use the correct README.md template from the Module Description document in I-Learn? | Yes |
| Did you completely populate the README.md template? | Yes |
| Did you create the video that includes you in a window, and reference it in the README.md file? | Yes |
| Did you post a link to your video in the proper MS Teams Channel? | Yes |
| Did you publish the code with the README.md (in the top-level folder) into a public GitHub repository? | Yes |

1. How many hours did you spend on this module this Sprint? Include all time including planning, researching, implementation, troubleshooting, documentation, video production, and publishing.   
   Record your total time here:

Paste your time log here including time spent each day on your project:

|  |  |  |
| --- | --- | --- |
|  | **First Week of Sprint** | **Second Week of Sprint** |
| **Monday** |  | Continue writing code (2 hours) |
| **Tuesday** |  |  |
| **Wednesday** | Do module plan (.5 hours) |  |
| **Thursday** |  | Continue writing code (3 hours) |
| **Friday** | Learn about Rust, setup, create repo (2 hours) | Continue writing code (3 hours) |
| **Saturday** | Begin writing code (1.5 hours) | Insert comments, make video and readme and submit (2 hour) |

Total time spent: 14 hours

1. What learning strategies worked well in this module and what strategies (or lack of strategy) did not work well? How can you improve in the next module?

The strategies that worked best for me this module was to prepare by thoroughly reading the basics of Rust before beginning. I also started much earlier in the first week of this sprint, so I had more time to complete this project.

The strategies that didn’t work so well for me was ignoring forums for the first while when I came across problems. I also could have split the main function of my program into a few more functions to keep that precise rather than getting lost trying to read through it.

For my next sprint, I will split my program into more functions to keep everything much simpler. I will also use all resources available to me rather than ignoring a large portion of them.