**Calculator App Release 3**

Craig McGuinness

Grand Canyon University

ITT 310

Dr. Bob

November 21st, 2021.

**A calculator with a cartoon face on it

Description automatically generated with medium confidence**

|  |  |
| --- | --- |
| **Topic:** | Milestone 4: Application Release 3 |
| **Date:** | November 28, 2021 |
| **Revision:** | Revision #3 |
| **Milestone Summary:** The goal for this week is to add union and structures to the code and prepare to clean up to release the full application. | |  |  |  | | --- | --- | --- | | **User Story / Task** | **Hours Worked** | **Hours Remaining** | | Rerun the first release and see where it went wrong | 1hr | Deadline Dec 12th | | Look over the requirements for this app release | 30 min | Deadline Dec 12th | | Research and produce a game plan for this week’s app release | 2hr | Deadline Dec 12th | | Start rewriting the code and make you have improved from last week | 2hr | Deadline Dec 12th | | Break for the Evening | 0 | 0 | | Review and make sure to add anything new | 30m | Deadline Dec 12th | | Run the code and troubleshoot | 1hr | Deadline Dec 12th | | Submit to Git | 1hr | Deadline Dec 12th | | Record the video via look | 10m | Deadline Dec 12th | | Review the assignment | 20m | Deadline Dec 12th | | Submit the Work | 5m | Deadline Dec 12th | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |
| **GIT URL:** | *https://github.com/pekimcg/CalculatorProjetct#calculatorprojetct* |

**Design Report Template**

**Overview**

In this milestone, we must change the code to follow the use structures and unions. This will highlight our skills in writing in the C language and the ability to use structures and unions along with commented codes.

**Initial Logic and Technical approach**

The main goal of this project is to create a simple text-based calculator to perform simple calculations. However, this release has been most difficult because of the overview and asking myself if this code is completely different or do I need to incorporate it with my actual calculation? So, this release will show how the structure and unions were set up differently and show how the size of the union will be at 4 bytes and the size of the data structure integer will be at 12 bytes.

**Flow Chart**

Diagram

Description automatically generated

**Pseudocode**

Start

type structure and union code

Show that union will have 4 bytes

Show struct will have 12 bytes

END.

**Risks**

The most common mistakes made by new coders creating applications are human errors. This is the main topic split into many different sections. To start, new coders tend to make a lot of errors, and this could be through spelling mistakes or invalid code formats. Next, coders can become lazy and impact their productivity. This could be a noticeably big distraction and new coders may hate their team and members or be given a long deadline for a project and cram in all their work weeks before because of poor time management. Finally using online resources. Some of the online resources are not verified and if coders use codes online, they could be causing serious harm to their project and potentially their company security. Now as the projects get more complicated so do the risks, but the black box and white box testing methods would help with testing the software and application before release to help mitigate any doubt for risks. A black box test just means it is used to test software without any knowledge of the structure of the thing being tested. A white box is an approach the opposite, where the tester knows the internal structure.

**Test Cases**

**<ReadMe.Txt>** The test case will be in the project folder.