./

GENESIS - Learning Outcome & Mini-project Summary Report



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **To be Approved** | **Remarks/Revision Details** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Details**

Contents

[Contents 3](#_Toc69378245)

[Mini project -1: Applied SDLC [Team:99003748, 99003745,99003744,99003747] 4](#_Toc69378246)

[**Module topic:** 4](#_Toc69378247)

[**SWOT Analysis:** 4](#_Toc69378248)

[**REQUIREMENTS:** 5](#_Toc69378249)

[**DESIGN: Roots and Powers** 5](#_Toc69378250)

[HLR DIAGRAM: 5](#_Toc69378251)

[LLR DIAGRAM: 6](#_Toc69378252)

[**TEST PLAN:** 7](#_Toc69378253)

[**Implementation Summary** 8](#_Toc69378254)

[Git Link 8](#_Toc69378255)

[Git Dashboard 8](#_Toc69378256)

[Summary 9](#_Toc69378257)

[Challenges faced and how were they overcome. 9](#_Toc69378258)

[Objectives & Requirements 9](#_Toc69378259)

[Design 9](#_Toc69378260)

[Test Plan 9](#_Toc69378261)

[Implementation Summary 9](#_Toc69378262)

[Video Summary 10](#_Toc69378263)

[Git Link 10](#_Toc69378264)

[Git Dashboard 10](#_Toc69378265)

[Summary 10](#_Toc69378266)

[Individual Contribution & Highlights 10](#_Toc69378267)

[Summary 10](#_Toc69378268)

[Challenges faced and how were they overcome 10](#_Toc69378269)

[Future Scope (If applicable) 10](#_Toc69378270)

[Miniproject -2 [Team/Individual] 12](#_Toc69378271)

[Module/s 12](#_Toc69378272)

[Topic and Subtopics 12](#_Toc69378273)

[Objectives & Requirements 12](#_Toc69378274)

[Design 12](#_Toc69378275)

[Test Plan 12](#_Toc69378276)

[Implementation Summary 12](#_Toc69378277)

[Git Link 12](#_Toc69378278)

[Git Dashboard 12](#_Toc69378279)

[Summary 12](#_Toc69378280)

[Individual Contribution & Highlights 12](#_Toc69378281)

[Summary 12](#_Toc69378282)

[Challenges faced and how were they overcome 12](#_Toc69378283)

# Mini project -1: Applied SDLC [Team:99003748, 99003745,99003744,99003747]

## **Module topic:**

An **electronic calculator** is typically a portable electronic device used to perform calculations, ranging from basic arithmetic to complex mathematics.

The first solid-state electronic calculator was created in the early 1960s. Pocket-sized devices became available in the 1970s, especially after the Intel 4004, the first microprocessor, was developed by Intel for the Japanese calculator company Busicom. They later became used commonly within the petroleum industry (oil and gas).

Modern electronic calculators vary from cheap, give-away, credit-card-sized models to sturdy desktop models with built-in printers. They became popular in the mid-1970s as the incorporation of integrated circuits reduced their size and cost. By the end of that decade, prices had dropped to the point where a basic calculator was affordable to most and they became common in schools.

Computer operating systems as far back as early Unix have included interactive calculator programs such as dc and hoc, and calculator functions are included in almost all personal digital assistant (PDA) type devices, the exceptions being a few dedicated address book and dictionary devices.

**My product name:** Math operation calculatorSub top

**Subtopic:** roots and cubes program

## **SWOT Analysis:**

|  |  |
| --- | --- |
| STRENGTHS  multi utility  user friendly  compact design  sturdy and  2-line display | WEAKNESS  Does not support very complex calculations |
| OPPORTUNITIES  Useful for all age groups  Viable in all educational institutions | **THREATS**  **New added features from other brands** |

## **REQUIREMENTS:**

Fraction calculations

Comes with slide-on hard case.

Equation calculations

Integration/differential calculations

Roots and powers

Sin, tan, cos, floor, ceil functions.

Matrix calculations

User step by step guide

40 scientific constants

## **DESIGN: Roots and Powers**

### HLR DIAGRAM:

#### CLASS DIAGRAM:

Graphical user interface, text, website

Description automatically generated

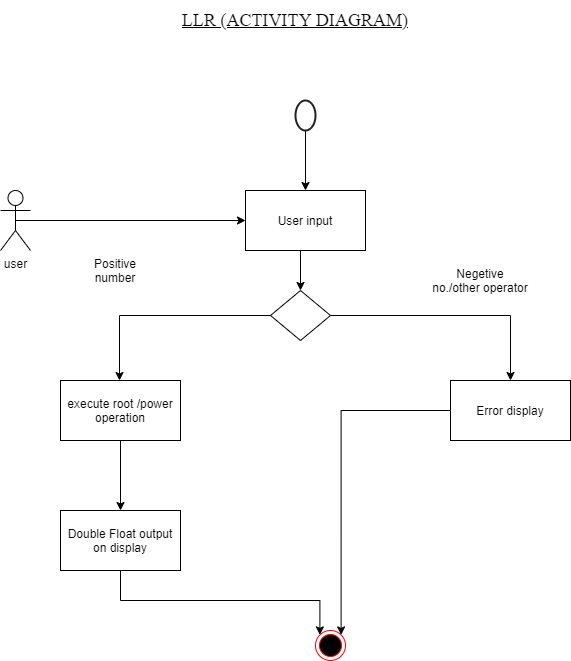
### LLR DIAGRAM:

#### CLASS DIAGRAM

A picture containing logo

Description automatically generated

#### ACTIVITY DIAGRAM



## **TEST PLAN:**

Test plan for square root and power:

* square root of negative no. is imaginary ---> error.
* Square root of zero ----> invalid
* Not whole no -----> invalid

Graphical user interface, application

Description automatically generated

## **Implementation Summary**

This mini project gave us an idea on how a software is developed in the industry. It follows two types of methodology Agile or V model. Different parts of the software were implemented like the header file, cpp file, test cases file and unit testing file. All badges including valgrind, cpp check, unity tests etc. were obtained.

### Git Link

https://github.com/99003745/T4\_SDLC\_PRODUCT\_CALC

#### Git inspector summary

Text

Description automatically generated

#### Build

#### 

#### Code quality and Issues or Bug Tracking

Text

Description automatically generated

#### Unit Testing

“Unit Testing setup alignment with test plans and summary of outcome”

## Individual Contribution & Highlights

“Brief on Contributions by you for Team”

### Git Dashboard

Graphical user interface, text, application

Description automatically generated

### Summary

Finally, the project is implemented, and test cases were passed.

### Challenges faced and how were they overcome.

* There was a problem during implementation of workflows.
* There was an error in Make file.
* Finally, we overcome by changing the code.

# Mini project -2: Advanced Python Mini Project

## Module/s

“Modules linked to the miniproject is Advanced python miniproject.”

## Topic and Subtopics

The topic is about the implementation of word search from a .txt file and writing a word prior and after the desired word.

## Objectives & Requirements

## **High Level requirement analysis**

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Requirements | Description | Status |
| **HL1** | Searching Word | Search the word from the text file given by the user. | Implemented |
| **HL2** | Writing | Write the words after and before of the searched word in new text file. | Implemented |
| **HL3** | Extracting user defined data | Write required data in the text file. | Implemented |

## **Low Level requirement analysis**

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Requirements** | **Description** | **Status** |
| **LL1** | Search Word | The user inputs the word to be searched through the text file. | Implemented |
| **LL2** | Searching word in every line of the text file. | The data to be searched is defined by the user and then searched throughout the entire text file. | Implemented |
| **LL3** | Writing the data into new text file. | Data given as an input by the user is searches in every line and the word after and before is extracted too and then added in a new text file along the searched word. | Implemented |

Python code for word search:

Text

Description automatically generated

### Text Description automatically generated

### Git Link

“Link to the repo: <https://github.com/99003748/Miniproject_python_99003748>”

### Git Dashboard

Graphical user interface, text, application, email

Description automatically generated

### Summary

“Word search has been implemented using python and all the test cases has been passed”

|  |  |
| --- | --- |
|  |  |

# Mini project -3: Embedded-C Mini Project

Design of Exterior lighting of car Exterior automotive lighting includes: • Head Lights Requirements • STM32cubeIDE 1.4.0• Arduino board Implementation of Exterior lighting Using HAL (hardware abstraction layer) we implemented our project.

## Objectives & Requirements

“High level and low level in the template ”

## Design

“System Level and subsystem level UMLs – Structural and Behavioral”

## Test Plan

“Integration level and unit level in the template”

## Implementation Summary

“Section focused toward’ s implementation aspects. Here it is only core summary while all the details are in the Git Repo

Note: The GitHub private repo should be documented (Readme.md files at each folder level)

Ensure code quality and clean code and description practices

Mandatory: To add the GitHub user - **stepin654321** as a contributor to the repo”

### Video Summary

“Please upload a short video on the repo for the walkthrough of the project (Team/Individual) less than 7min and less than 30MB File Size. Start is the Standard opening slide with title of miniproject + Team members followed by the walkthrough ”

### Git Link

“Link to the repo”

### Git Dashboard

“Screenshot of the GitHub Repo page with all the badges and summary”

### Summary

“Brief summary on the overall implementation”

#### Git inspector summary

“In linux install gitinspector and Run the command –

gitinspector -H -l -m -T -w -r --grading --format=html > gitinsp.html

and upload the same to your repo and paste the snapshot in the report”

#### Build

“Brief on outcome of the build and setup done”

#### Code quality and Issues or Bug Tracking

“Brief on code quality, errors and warnings flagged (issues created) and fixed ”

#### Unit Testing

“Unit Testing setup alignment with test plans and summary of outcome”

## Individual Contribution & Highlights

“Brief on Contributions by you for Team”

### Summary

“Key Highlights not covered till now, Softskills and technical side”

### Challenges faced and how were they overcome

“Brief and crisp”

### Future Scope (If applicable)

### 

# Miniproject -2 [Team/Individual]

## Module/s

### Topic and Subtopics

## Objectives & Requirements

## Design

## Test Plan

## Implementation Summary

### Git Link

### Git Dashboard

### Summary

#### Git inspector summary

#### Build

#### Code quality

#### Unit Testing

#### Issues

## Individual Contribution & Highlights

### Summary

### Challenges faced and how were they overcome