GENESIS - Learning Outcome & Mini-project Summary Report

LTTS
GLOBAL
ENGINEERING
ACADEMY

L&T Technology Services

Details

Ver. Rel. No.	Release Date	Prepared. By	Reviewed By	To be Approved	Remarks/Revision Details
1.	15/04/2021	Neha Tabassum			

Contents

List of Figures:

List of Tables:

Miniproject -1 [Team] : SDLC (Software Development Life Cycle)

Module/s

Module linked to the miniproject is SDLC.

Topic and Subtopics

Below is the list of core topics and subtopics being implemented:

SDLC (Software Development Life Cycle) :-
SDLC is a process followed by a software project, within a software organization. Contains a detailed description of how you can improve, maintain, modify and modify or improve a particular software. The life cycle describes how to improve software quality and the overall development process.

Phase 1: Requirement and Planning Analysis
Analysis is the most important and fundamental phase in the SDLC. It is done by senior team members with input from customers, industry experts, market surveys and sales department. This information is also used to plan the basic course of the project and to conduct feasibility studies in the technical, conservation and operational areas.

Phase 2: Requirement Definition
Once the requirements has been taken the next step is to clearly define and document the product requirements and is approved for customers or market analysts. This is done through an SRS (Software Requirement Specification) document that contains all the product requirements that will be created and developed during the life cycle of the project.

Phase 3: Product Design
SRS is the trust of product manufacturers to come up with the best technology for product development. In accordance with the requirements set out in SRS, usually more than one method of product design is proposed and documented in the DDS - Design Document Specification. An excellent design approach is chosen that clearly defines all product building modules and the representation of communication and data flow through external and third party modules.

Step 4: Building or developing a product
At this stage of the SDLC, real development begins and the product is built. Application code is generated per DDS. If construction is done in a detailed and systematic way, coding can be accomplished without much hassle. Various programming languages such as C, C ++, PHP, Java and Pascal are used for encoding the design.

Step 5: Product testing
This section is usually the basis of all categories as modern models of SDLC, testing activities are very involved in all sections of SDLC.

Section 6: Market Delivery and Care
Once the product has been tested and ready for shipment it is officially released from the relevant markets.

SWOT Analysis:
SWOT analysis is used to assess the organization’s current position before going to any new strategy.

SWOT stands for Strengths, Weaknesses, Opportunities, and Threats, so SWOT Analysis is a way to evaluate four business aspects. SWOT analysis is used to make the most of what a person has gained, benefiting their organization. It can also reduce the likelihood of failure, and eliminating the risks that would otherwise be unknown.

Strength
It is the thing that an organization does best, or in a way that separates the organization from its competitors.

Weaknesses
Now is the time to look at organization weaknesses. Be honest! SWOT analysis will be useful only if one collects all the necessary information. Therefore, it is better to be realistic, and to deal with unpleasant facts very quickly.

Opportunities
Opportunities to open up or opportunities for something good to happen, but one needs to seek them out through the organization.

Threats
Threats include anything that could adversely affect the business from the outside, such as procurement problems, changes in market demand, or a shortage of employees.

Unified Modelling Language (UML) Diagrams:
UML is a common language for interpreting, visualizing, constructing and transcribing software archeology. UML drawings are designed not only for engineers but also for business users, the general public, and anyone with an interest in understanding the system. There are two broad categories of it and they are further sub-divided as follows:

Structural Diagrams:

- Class Diagram
- Object Diagram
- Component Diagram
- Deployment Diagram
- Profile Diagram
- Package Diagram
- Composite Structure Diagram

Behavioral Diagrams:

- Use Case Diagram
- State Machine Diagram
- Activity Diagram

- Communication Diagram
- Sequence Diagram
- Interaction Overview Diagram
- Timing Diagram

Objectives & Requirements

The designed product has all the necessary features required by the undergraduates and postgraduates students including scholars. The High level requirements include arithmetic operations, decimals, trigonometric functions, nth root, power of n, fractions, percentage, logarithms, exponentials, number conversions etc. The low level requirements of the product includes addition, subtraction, multiplication, division under arithmetic operation, decimal upto 8 digits, trigonometric functions with square root and radians. Under number conversions there are the features like Binary to decimal conversion, Decimal to binary conversion, Decimal to octal conversion. Calculators also includes the functions of database management, higher accuracy, wider and has smart touch, solar cell operations, battery charging and is waterproof.

Requirement:-

High Level Requirement (HLR):

Table 1: HLR of Calculator

ID	Requirements	Description	Status
HLR_01	Arithmetic operations	Addition, Subtraction, Multiplication, Division, Remainder, GCD	Implemented
HLR_02	Percentage and power	Calculates percentage of one number wrt to another and raised to power of a number to the number provided.	Implemented
HLR_03	Trigonometric functions	Sine, cosine, tangent functions	Implemented
HLR_04	Number Conversions	Converts Binary to Decimal, Decimal to Binary and Decimal to Octal.	Implemented
HLR_05	Area Calculation	Calculates area of Square, Rectangle, Triangle and circle.	Implemented

Low Level Requirement (LLR):-

1. Arithmetic Operations:

Table 2: LLR of Arithmetic Operation

ID	Requirements	Description	Status
LLR_01	Addition	Takes two numbers as an input and adds them	Implemented
LLR_02	Subtraction	Takes two numbers as an input and subtracts them	Implemented
LLR_03	Multiply	Takes two numbers as an input and multiplies them	Implemented
LLR_04	Divide	Taking two numbers as an input and divides them	Implemented
LLR_05	Percentage	Takes two numbers as an input and calculate percentage of one number with the other	Implemented
LLR_06	Greatest Common Divisor(GCD)	Takes two numbers and calculate their GCD	Implemented

2. Percentage and Power:

Table 3: LLR of Percentage and Power

ID	Requirements	Description	Status
LLR_01	Percentage	Calculates percentage of two numbers	Implemented
LLR_02	Power	Calculates power of one number raised to the other	Implemented

3. Trigonometric Functions:

Table 4: LLR of Trigonometric Function

ID	Requirements	Description	Status
LLR_01	Sine Function	Calculate sine function of a number in both degree and radian	Implemented
LLR_02	Cosine Function	Calculate cosine function of a number in both degree and radian	Implemented
LLR_03	Tangent Function	Calculate tan function of a number in both degree and radian	Implemented

4. Number conversions:

Table 5: LLR of Number Conversion

ID	Requirements	Description	Status
LLR_01	Binary Function	Convert binary number to decimal number	Implemented
LLR_02	Decimal Function	Convert Decimal number to binary number	Implemented
LLR_03	Octal Function	Convert Decimal to octal number	Implemented

4. Area Calculator:

Table 6: LLR of Area Calculator

ID	Requirements	Description	Status
LLR_01	Square Function	Calculate area of a square	Implemented
LLR_02	Rectangle Function	Calculate area of a rectangle	Implemented
LLR_03	Circle Function	Calculate area of a Circle	Implemented
LLR_04	Triangle Function	Calculate area of a Triangle	Implemented

Design

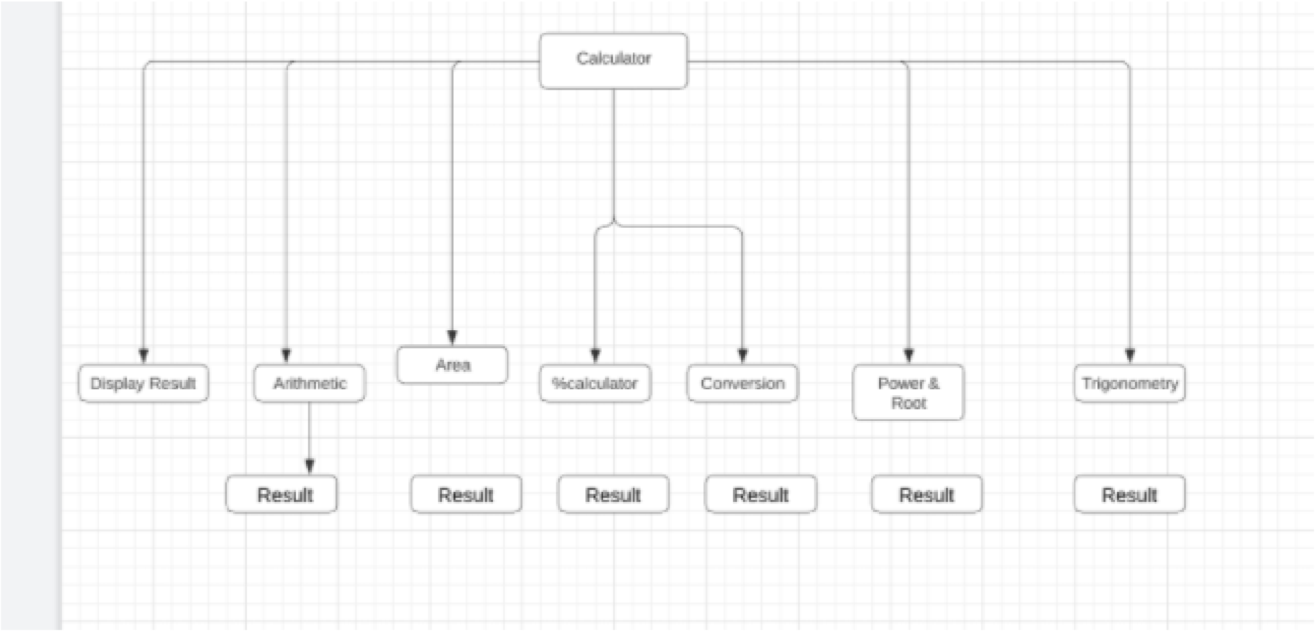


Figure 1: High Level Diagram of Calculator

Structural Diagram:-

Arithmetic Operation:

System Level:

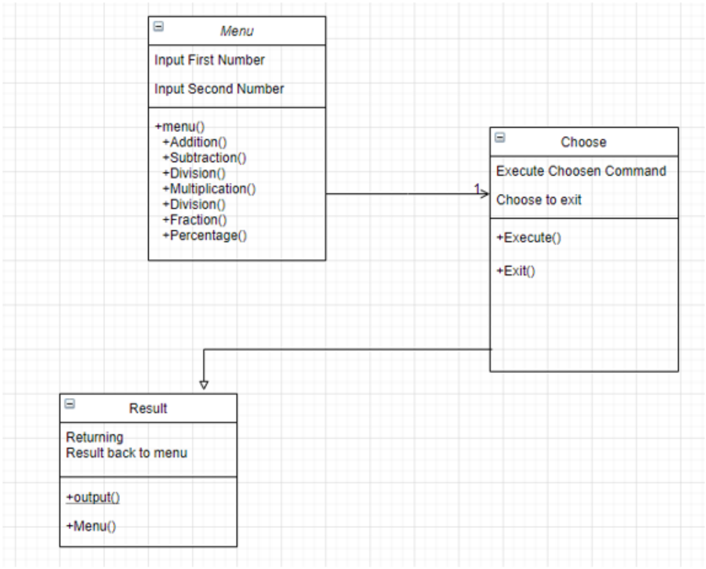


Figure 2: System Level Structural Diagram of Arithmetic Operation

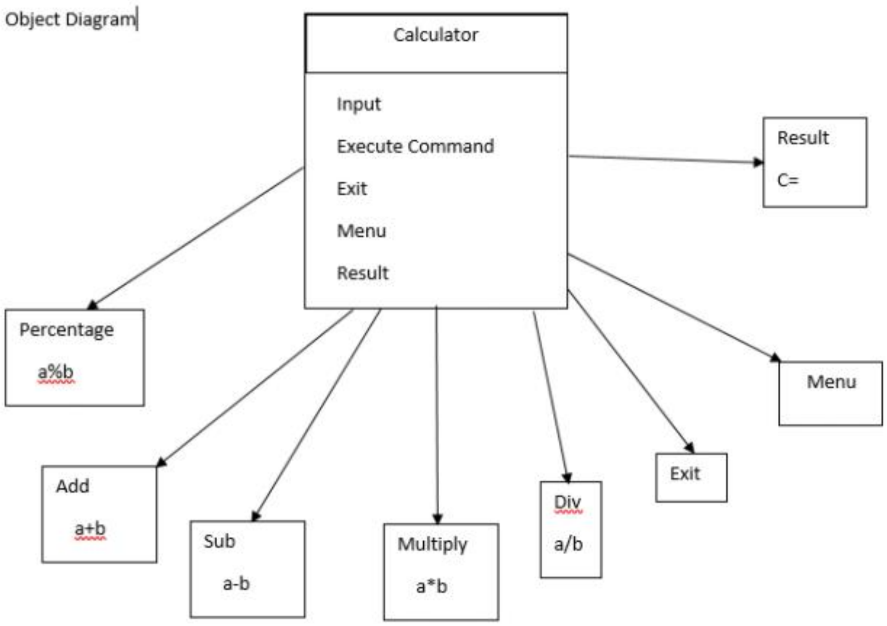


Figure 3: System Level Object Diagram of Arithmetic operation

Subsystem Level:

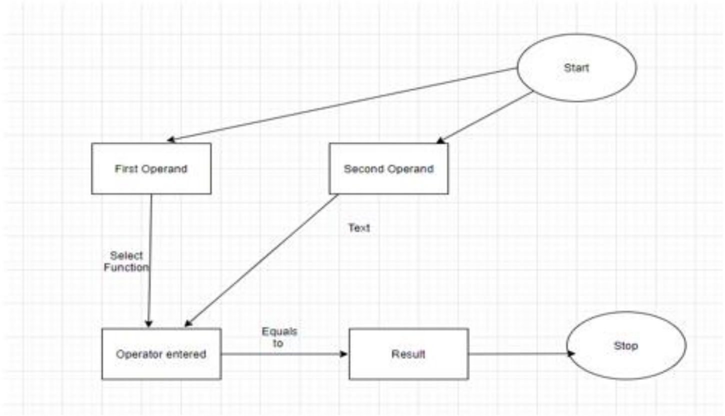


Figure 4: Subsystem level Structural Diagram of Arithmetic Operation

Trigonometry Function:

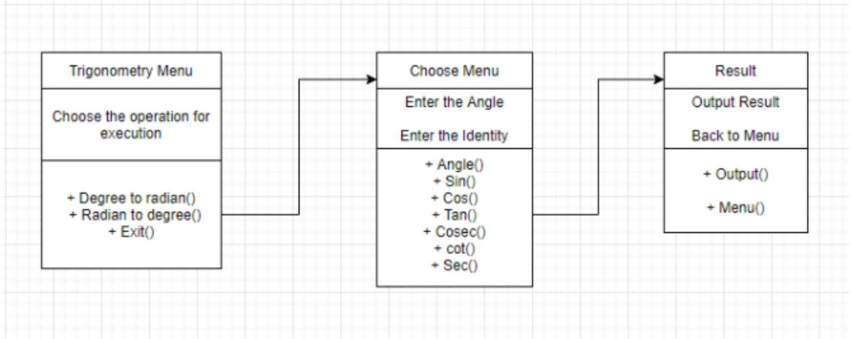


Figure 5: Structural Diagram of Trigonometric Function

Number Conversion:

System Level:

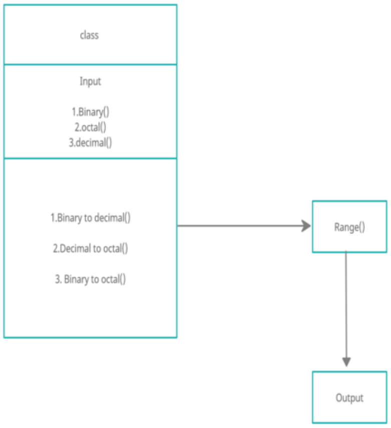


Figure 6: System Level Structural Diagram of Number conversion

Subsystem-level

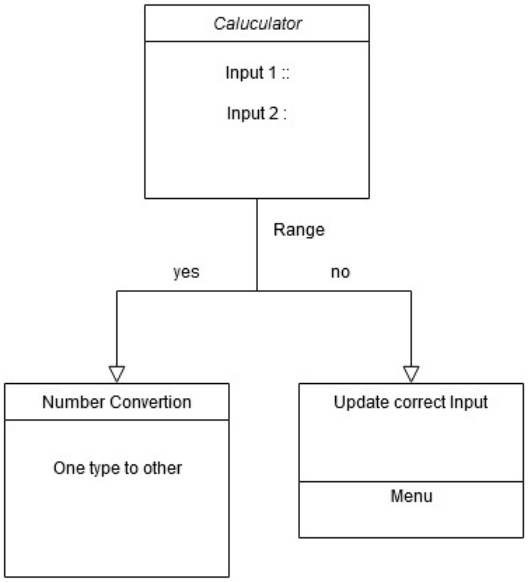


Figure 7: Subsystem Level Structural Diagram of Number Conversion

Area Calculator:

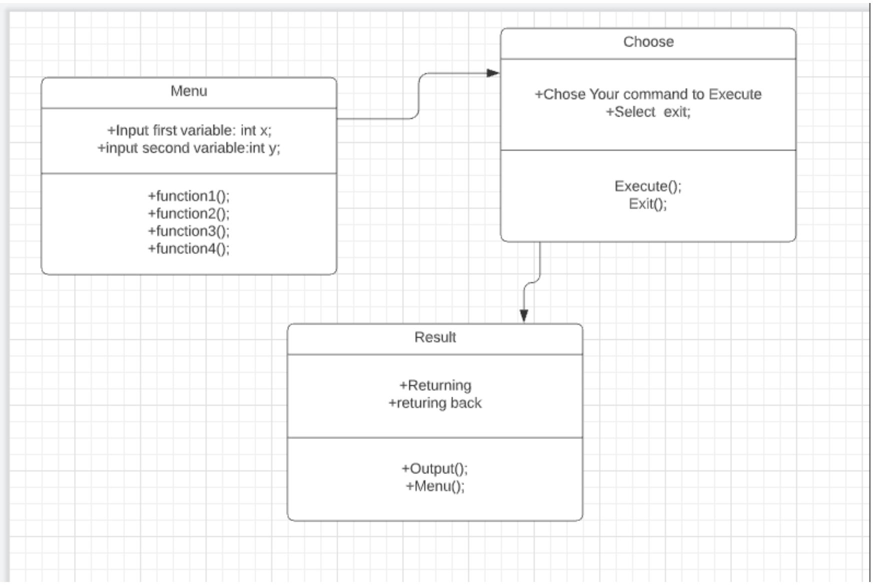


Figure 8: Structural Diagram of Area Calculator

Behavioral Diagram:

Arithmetic Operation:

System Level:

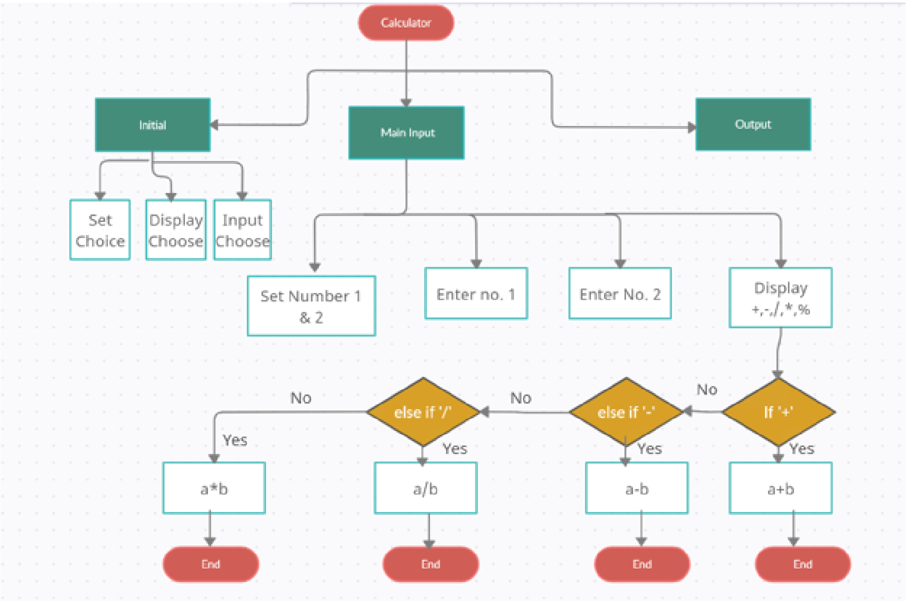


Figure 9: System Level Behavioural Diagram of Arithmetic Operation

Subsystem-Level:

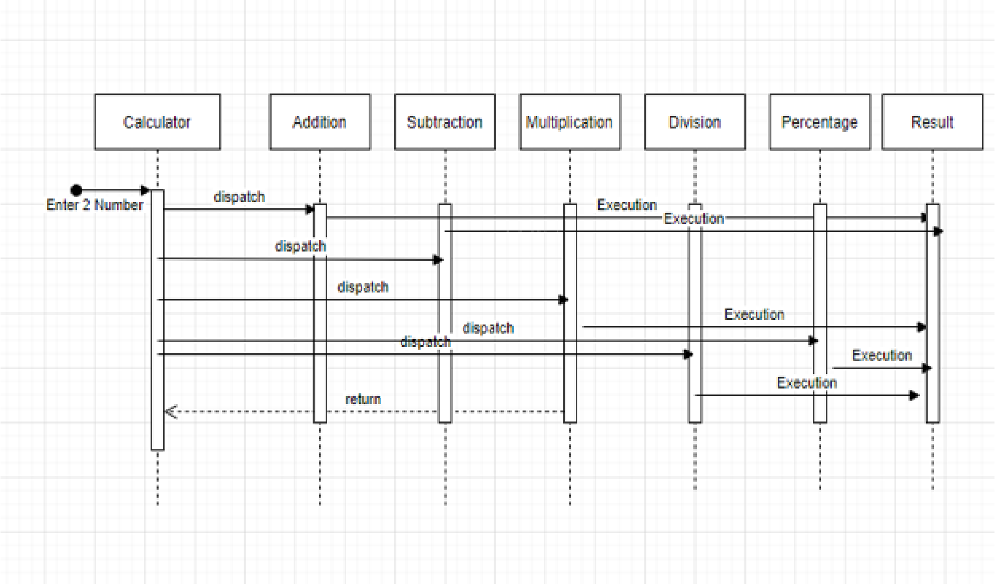


Figure 10: Subsystem Level Behavioral Diagram of Arithmetic Operation

Number Conversion:

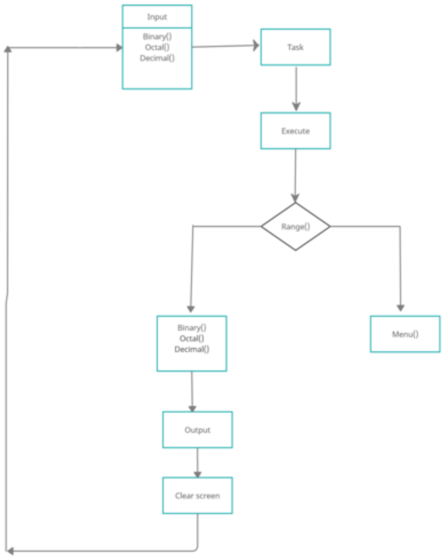


Figure 11: Behavioural Diagram of Number Conversion.

Test Plan

High Level Test Plan:

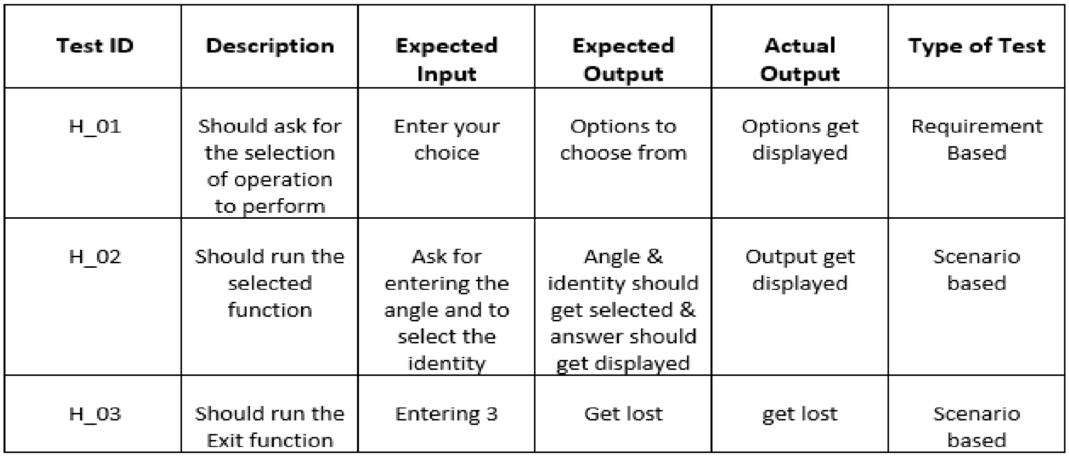
Arithmetic Operation

Table 7: Test Plan of Arithmetic Operation

Test ID	Description	Expected Input	Exp. Out.	Actual output	Type of Test
H_01	➢ Perform operations of 2 positive numbers	➢ 2 + 4.05 ➢ 4/6+2/3 ➢ 4-3 ➢ 3*4 ➢ 8/4	➢ 6.05 ➢ 8/6=1.333333 ➢ 1.0 ➢ 12.0 ➢ 2.0	➢ 6.0 ➢ 1.33 ➢ 1 ➢ 12.0 ➢ 2.0	Requirement based
H_02	➢ Perform operation of positive number and zero	➢ 2+0 ➢ 2-0 ➢ 2*0	➢ 2 ➢ 2 ➢ 0	➢ 2.0 ➢ 2.0 ➢ 0	Scenario based
H_03	➢ Perform operation of large numbers	➢ 345+567 ➢ 893.03-876.23 ➢ 45*34 ➢ 345/28	➢ 912 ➢ 16.8 ➢ 1530 ➢ 12.32	➢ 912.0 ➢ 16.80 ➢ 1530.0 ➢ 12.32	Boundary based

Trigonometric Operation

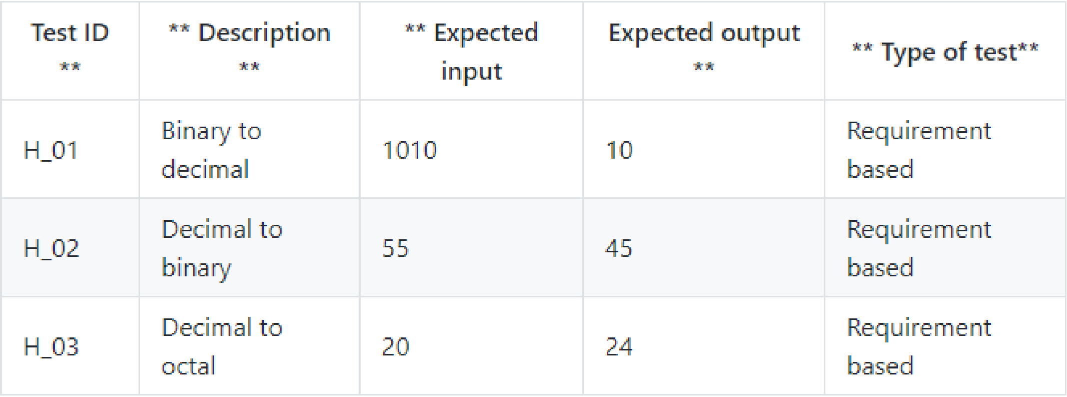
Table 8: Test Plan of Trigonometric Operation



Test ID	Description	Expected Input	Expected Output	Actual Output	Type of Test
H_01	Should ask for the selection of operation to perform	Enter your choice	Options to choose from	Options get displayed	Requirement Based
H_02	Should run the selected function	Ask for entering the angle and to select the identity	Angle & identity should get selected & answer should get displayed	Output get displayed	Scenario based
H_03	Should run the Exit function	Entering 3	Get lost	get lost	Scenario based

Number Conversion

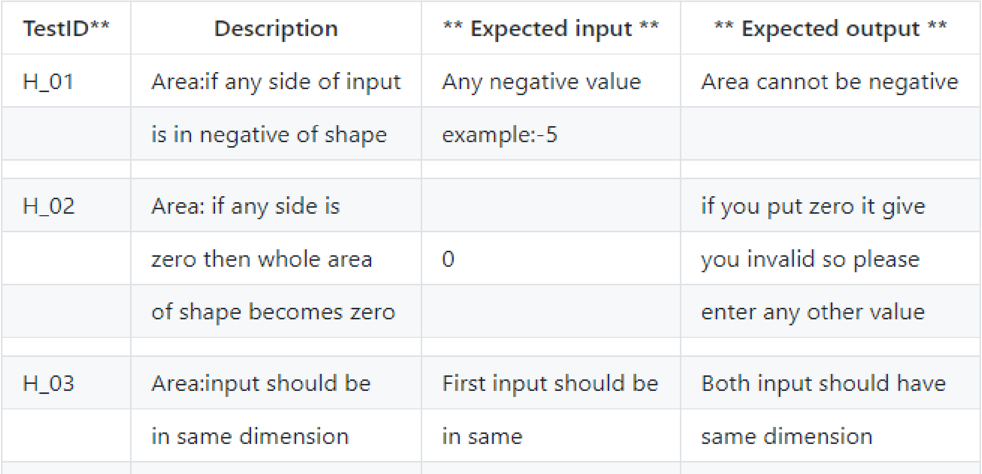
Table 9: Test Plan of Number Conversion



Test ID **	** Description **	** Expected input	Expected output **	** Type of test**
H_01	Binary to decimal	1010	10	Requirement based
H_02	Decimal to binary	55	45	Requirement based
H_03	Decimal to octal	20	24	Requirement based

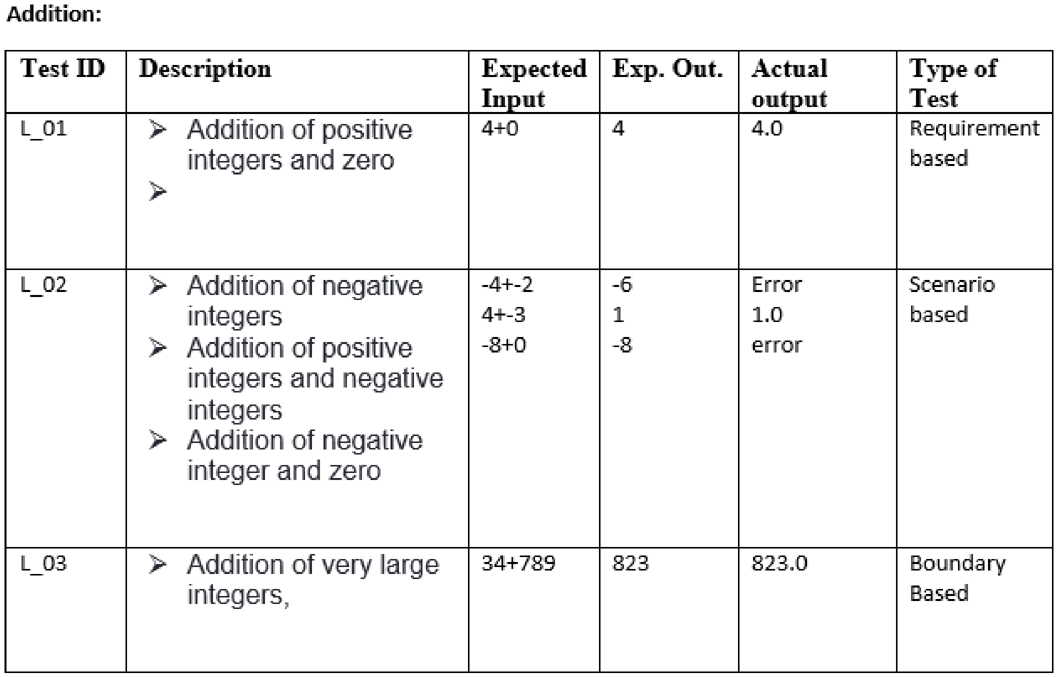
Area Calculation

Table 10: Test Plan of Area Calculator



TestID**	Description	** Expected input **	** Expected output **
H_01	Area:if any side of input is in negative of shape	Any negative value example:-5	Area cannot be negative
H_02	Area: if any side is zero then whole area of shape becomes zero	0	if you put zero it give you invalid so please enter any other value
H_03	Area:input should be in same dimension	First input should be in same	Both input should have same dimension

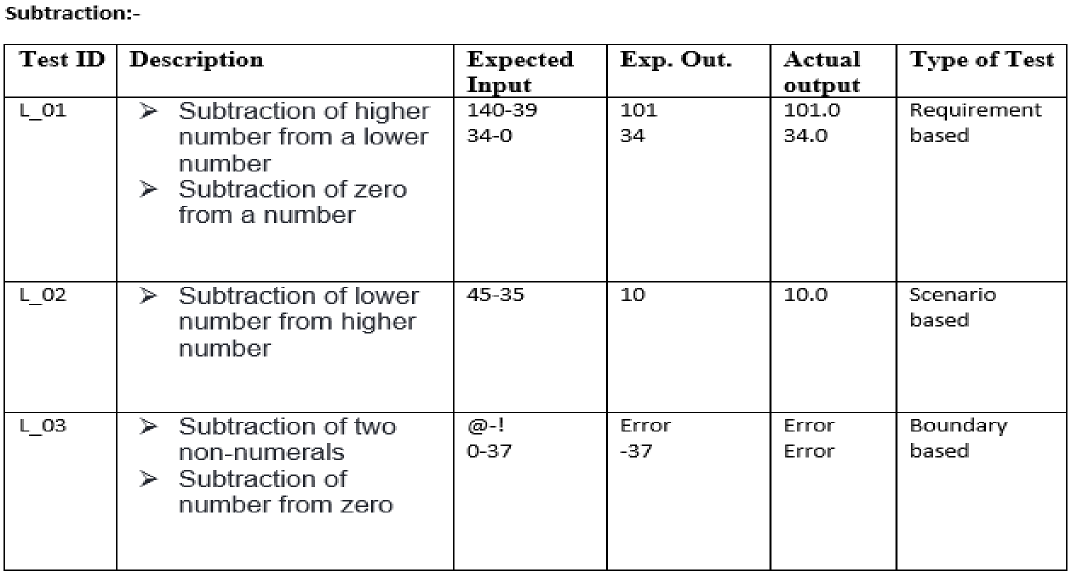
Low Level Test Plan:



Addition:

Test ID	Description	Expected Input	Exp. Out.	Actual output	Type of Test
L_01	➢ Addition of positive integers and zero ➢	4+0	4	4.0	Requirement based
L_02	➢ Addition of negative integers ➢ Addition of positive integers and negative integers ➢ Addition of negative integer and zero	-4+-2 4+-3 -8+0	-6 1 -8	Error 1.0 error	Scenario based
L_03	➢ Addition of very large integers,	34+789	823	823.0	Boundary Based

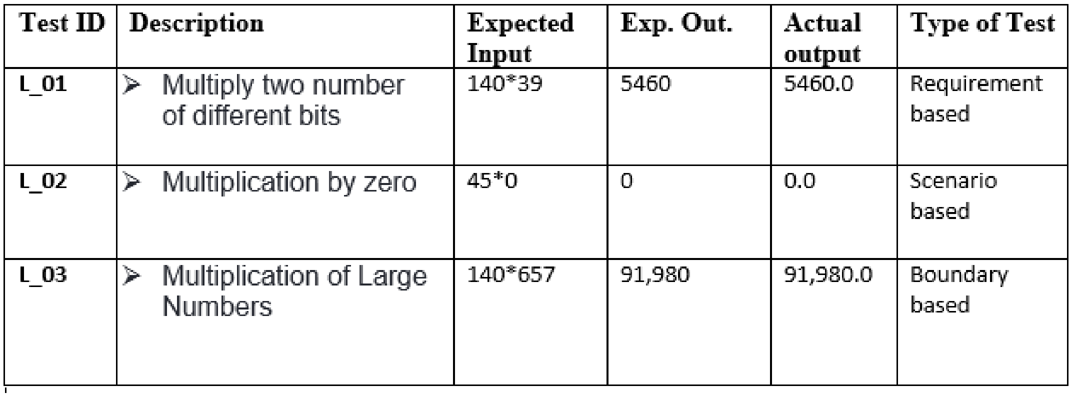
-



Subtraction:-

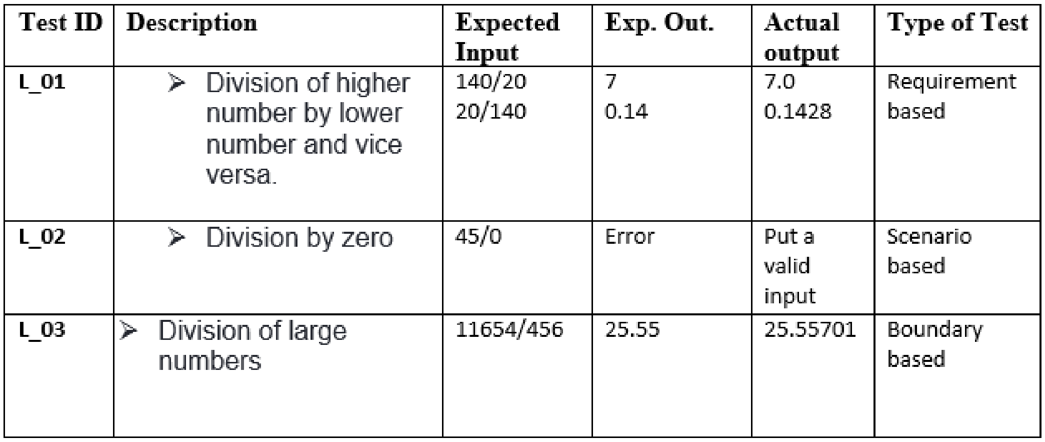
Test ID	Description	Expected Input	Exp. Out.	Actual output	Type of Test
L_01	➢ Subtraction of higher number from a lower number ➢ Subtraction of zero from a number	140-39 34-0	101 34	101.0 34.0	Requirement based
L_02	➢ Subtraction of lower number from higher number	45-35	10	10.0	Scenario based
L_03	➢ Subtraction of two non-numerals ➢ Subtraction of number from zero	@-! 0-37	Error -37	Error Error	Boundary based

Multiplication:-



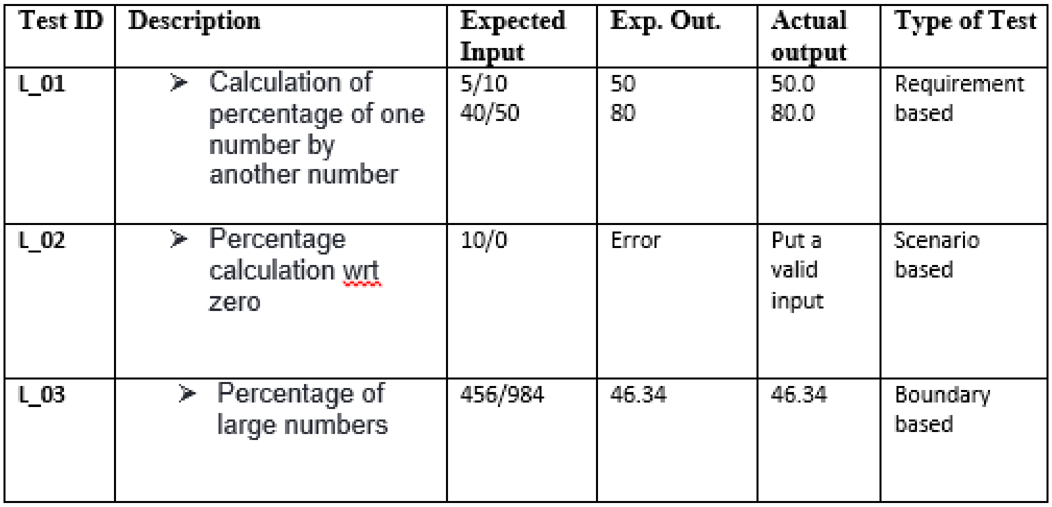
Test ID	Description	Expected Input	Exp. Out.	Actual output	Type of Test
L_01	➢ Multiply two number of different bits	140*39	5460	5460.0	Requirement based
L_02	➢ Multiplication by zero	45*0	0	0.0	Scenario based
L_03	➢ Multiplication of Large Numbers	140*657	91,980	91,980.0	Boundary based

Division:-

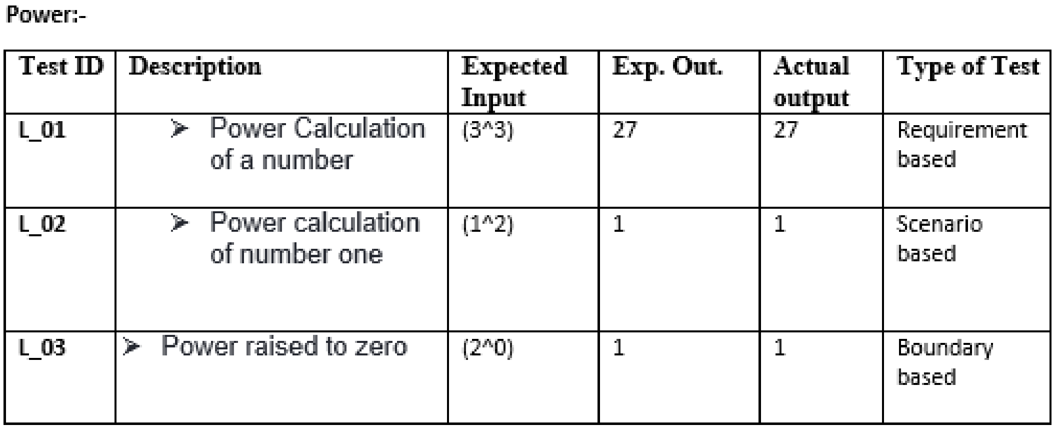


Test ID	Description	Expected Input	Exp. Out.	Actual output	Type of Test
L_01	➢ Division of higher number by lower number and vice versa.	140/20 20/140	7 0.14	7.0 0.1428	Requirement based
L_02	➢ Division by zero	45/0	Error	Put a valid input	Scenario based
L_03	➢ Division of large numbers	11654/456	25.55	25.55701	Boundary based

Percentage:-

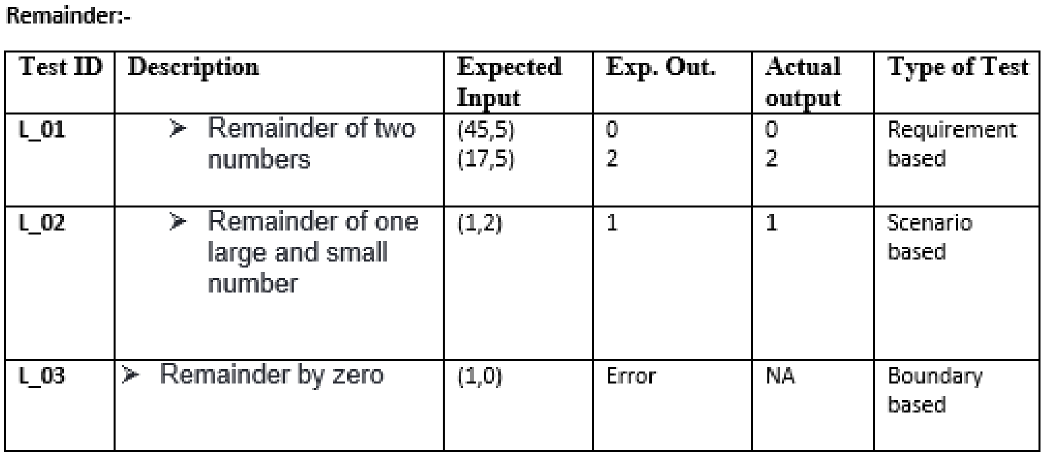


Test ID	Description	Expected Input	Exp. Out.	Actual output	Type of Test
L_01	➢ Calculation of percentage of one number by another number	5/10 40/50	50 80	50.0 80.0	Requirement based
L_02	➢ Percentage calculation wrt zero	10/0	Error	Put a valid input	Scenario based
L_03	➢ Percentage of large numbers	456/984	46.34	46.34	Boundary based



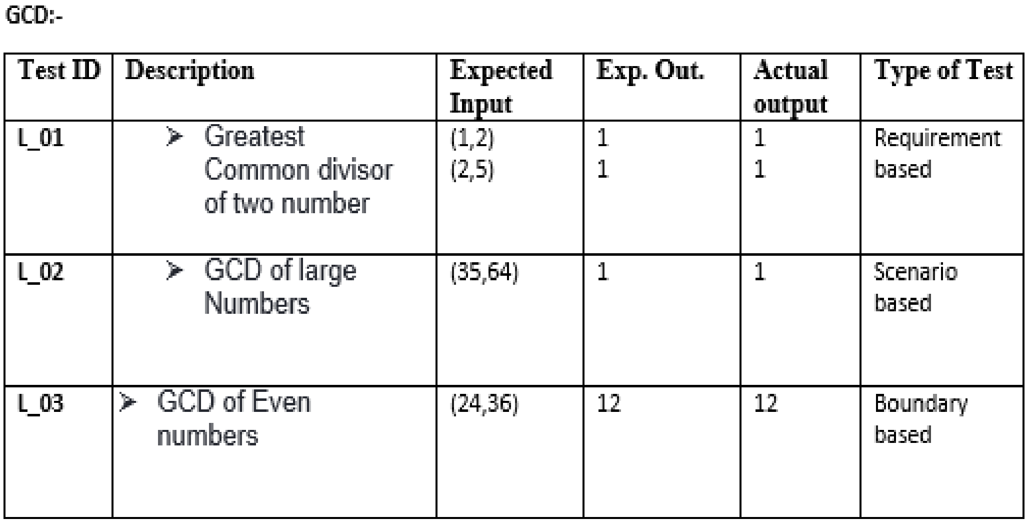
Power:-

Test ID	Description	Expected Input	Exp. Out.	Actual output	Type of Test
L_01	➢ Power Calculation of a number	(3^3)	27	27	Requirement based
L_02	➢ Power calculation of number one	(1^2)	1	1	Scenario based
L_03	➢ Power raised to zero	(2^0)	1	1	Boundary based



Remainder:-

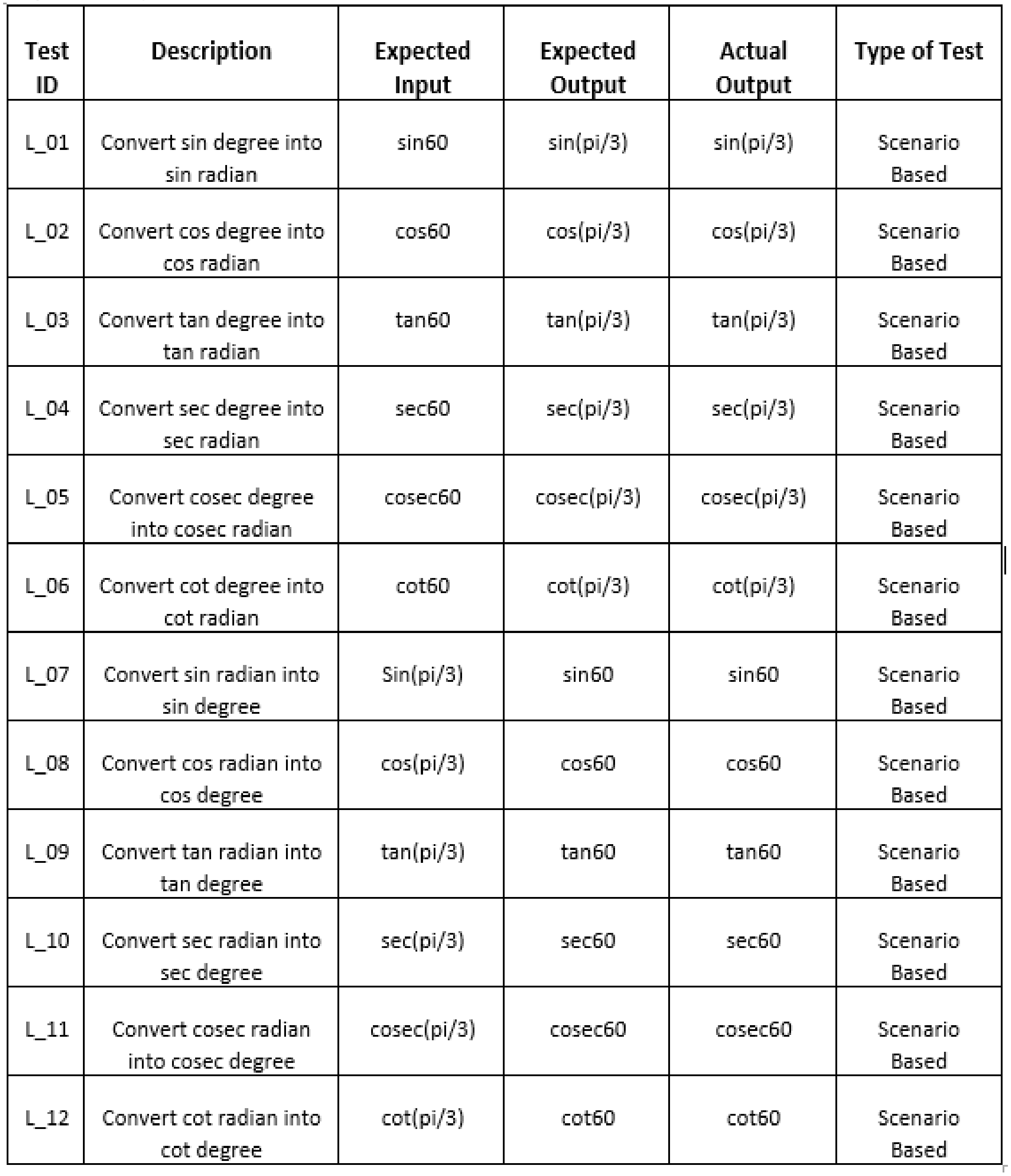
Test ID	Description	Expected Input	Exp. Out.	Actual output	Type of Test
L_01	➢ Remainder of two numbers	(45,5) (17,5)	0 2	0 2	Requirement based
L_02	➢ Remainder of one large and small number	(1,2)	1	1	Scenario based
L_03	➢ Remainder by zero	(1,0)	Error	NA	Boundary based



GCD:-

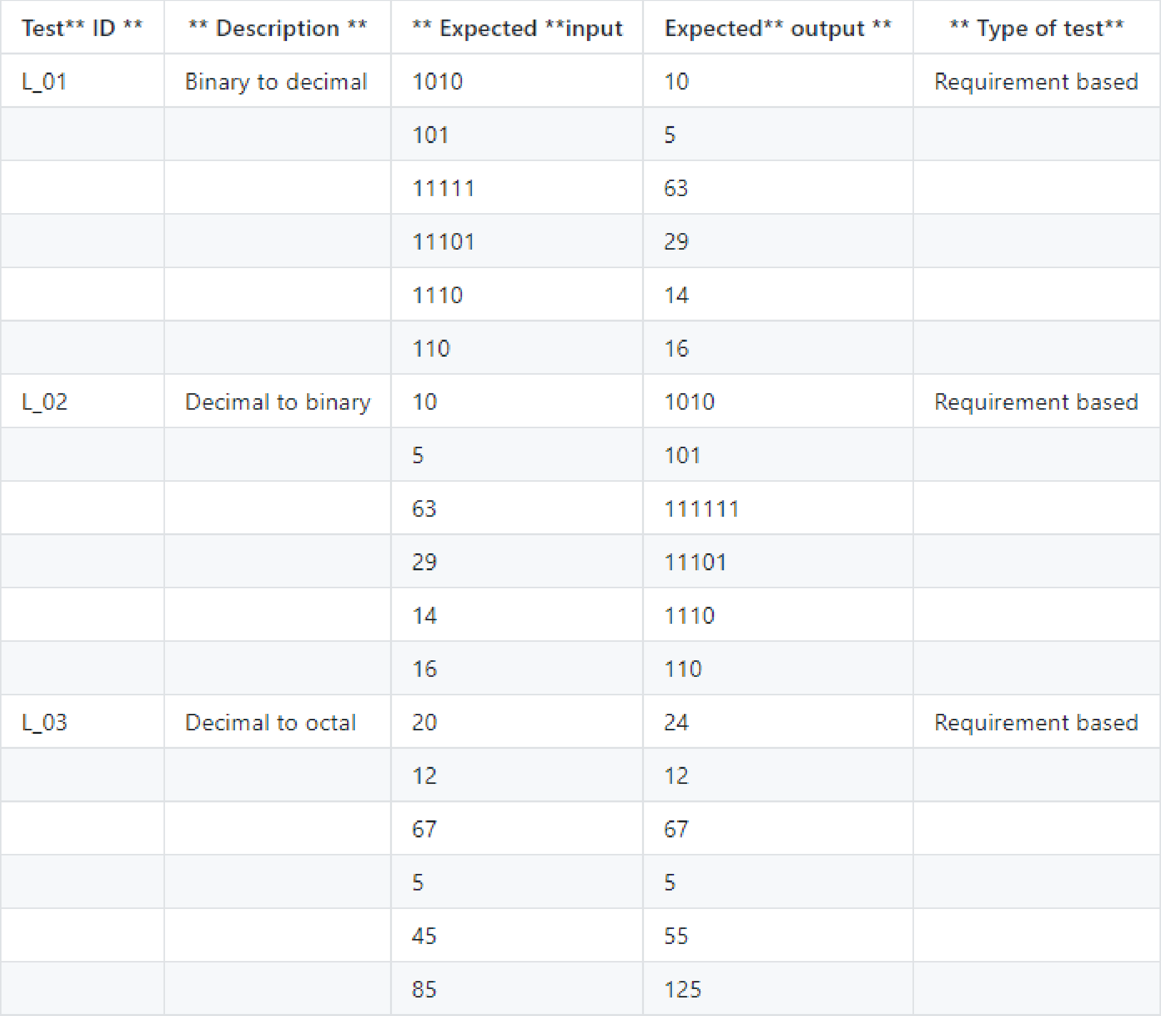
Test ID	Description	Expected Input	Exp. Out.	Actual output	Type of Test
L_01	➢ Greatest Common divisor of two number	(1,2) (2,5)	1 1	1 1	Requirement based
L_02	➢ GCD of large Numbers	(35,64)	1	1	Scenario based
L_03	➢ GCD of Even numbers	(24,36)	12	12	Boundary based

Trigonometry:-



Test ID	Description	Expected Input	Expected Output	Actual Output	Type of Test
L_01	Convert sin degree into sin radian	sin60	sin(pi/3)	sin(pi/3)	Scenario Based
L_02	Convert cos degree into cos radian	cos60	cos(pi/3)	cos(pi/3)	Scenario Based
L_03	Convert tan degree into tan radian	tan60	tan(pi/3)	tan(pi/3)	Scenario Based
L_04	Convert sec degree into sec radian	sec60	sec(pi/3)	sec(pi/3)	Scenario Based
L_05	Convert cosec degree into cosec radian	cosec60	cosec(pi/3)	cosec(pi/3)	Scenario Based
L_06	Convert cot degree into cot radian	cot60	cot(pi/3)	cot(pi/3)	Scenario Based
L_07	Convert sin radian into sin degree	Sin(pi/3)	sin60	sin60	Scenario Based
L_08	Convert cos radian into cos degree	cos(pi/3)	cos60	cos60	Scenario Based
L_09	Convert tan radian into tan degree	tan(pi/3)	tan60	tan60	Scenario Based
L_10	Convert sec radian into sec degree	sec(pi/3)	sec60	sec60	Scenario Based
L_11	Convert cosec radian into cosec degree	cosec(pi/3)	cosec60	cosec60	Scenario Based
L_12	Convert cot radian into cot degree	cot(pi/3)	cot60	cot60	Scenario Based

Number Conversion:-



Test** ID **	** Description **	** Expected **input	Expected** output **	** Type of test**
L_01	Binary to decimal	1010	10	Requirement based
		101	5	
		11111	63	
		11101	29	
		1110	14	
		110	16	
L_02	Decimal to binary	10	1010	Requirement based
		5	101	
		63	111111	
		29	11101	
		14	1110	
		16	110	
L_03	Decimal to octal	20	24	Requirement based
		12	12	
		67	67	
		5	5	
		45	55	
		85	125	

Implementation Summary

The implementation part consist of software implementation of the designed product known as calculator as per the specification mentioned in requirements and design. The folder consist of inc folder which consist of various header files (.h) of various section of designed calculator. It also contains src folder which is having the source files (i.e .c file). The test folder contains the test case implementation of the designed product. It also contains the Makefile which is used to build, run and clean all the multiple files and check for the test cases that are formulated according to the requirement specified.

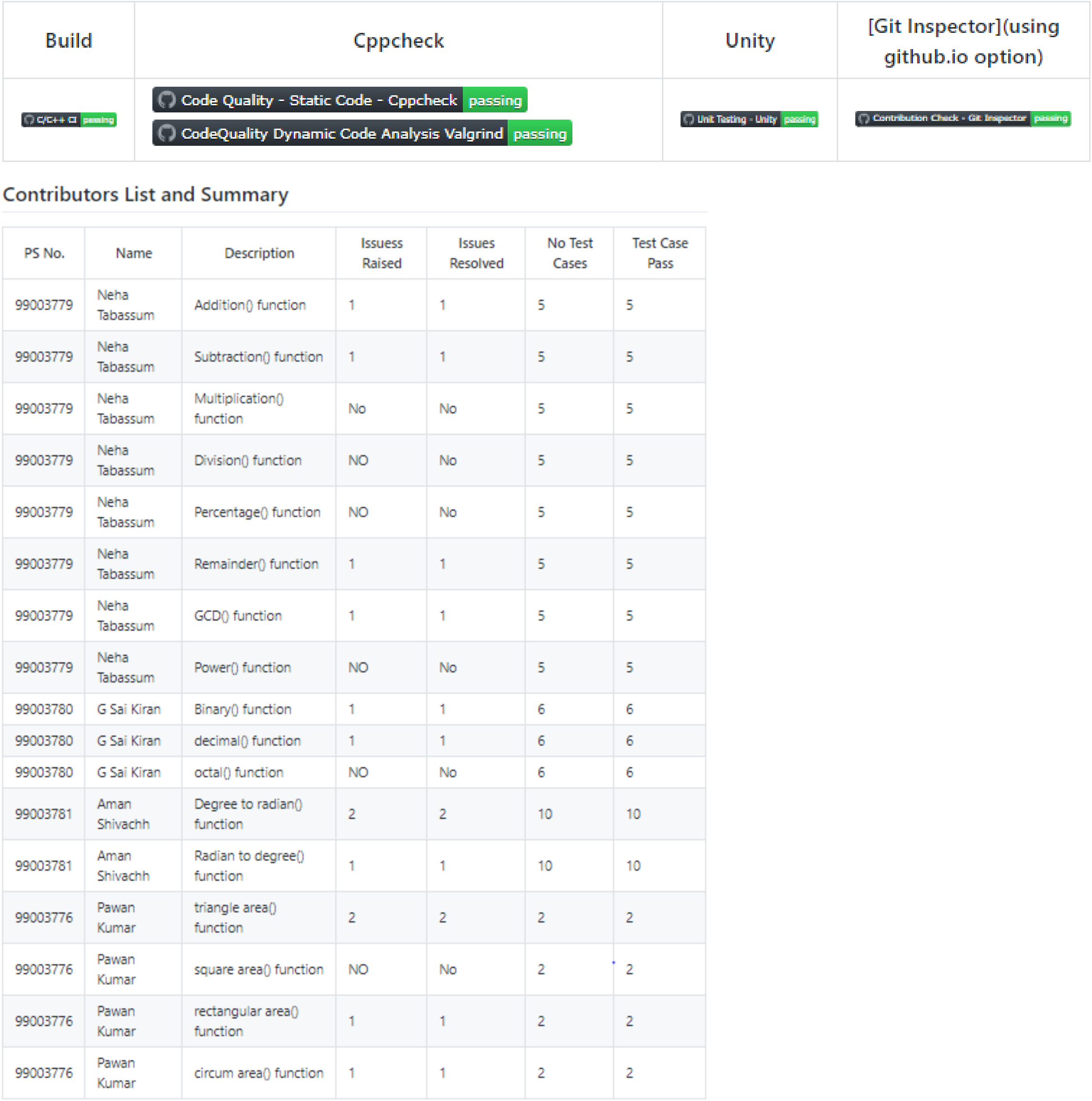
The source file contains the implementation of the following functions:

- Arithmetic operation (Addition, Subtraction, Multiplication, Division, Remainder, GCD)
- Percentage and power operation
- Trigonometric functions (Radian and Degree)
- Number conversion function (Binary, Decimal, Octal)
- Area Calculator operation (Rectangle, Square, Triangle, Circle)

Git Link

https://github.com/99003781/N8-Calculator.git

Git Dashboard



Build	Cppcheck	Unity	[Git Inspector](using github.io option)
C/C++ CI passing	Code Quality - Static Code - Cppcheck passing CodeQuality Dynamic Code Analysis Valgrind passing	Unit Testing - Unity passing	Contribution Check - Git Inspector passing

Contributors List and Summary

PS No.	Name	Description	Issuess Raised	Issues Resolved	No Test Cases	Test Case Pass
99003779	Neha Tabassum	Addition() function	1	1	5	5
99003779	Neha Tabassum	Subtraction() function	1	1	5	5
99003779	Neha Tabassum	Multiplication() function	No	No	5	5
99003779	Neha Tabassum	Division() function	NO	No	5	5
99003779	Neha Tabassum	Percentage() function	NO	No	5	5
99003779	Neha Tabassum	Remainder() function	1	1	5	5
99003779	Neha Tabassum	GCD() function	1	1	5	5
99003779	Neha Tabassum	Power() function	NO	No	5	5
99003780	G Sai Kiran	Binary() function	1	1	6	6
99003780	G Sai Kiran	decimal() function	1	1	6	6
99003780	G Sai Kiran	octal() function	NO	No	6	6
99003781	Aman Shivachh	Degree to radian() function	2	2	10	10
99003781	Aman Shivachh	Radian to degree() function	1	1	10	10
99003776	Pawan Kumar	triangle area() function	2	2	2	2
99003776	Pawan Kumar	square area() function	NO	No	2	2
99003776	Pawan Kumar	rectangular area() function	1	1	2	2
99003776	Pawan Kumar	circum area() function	1	1	2	2

Summary

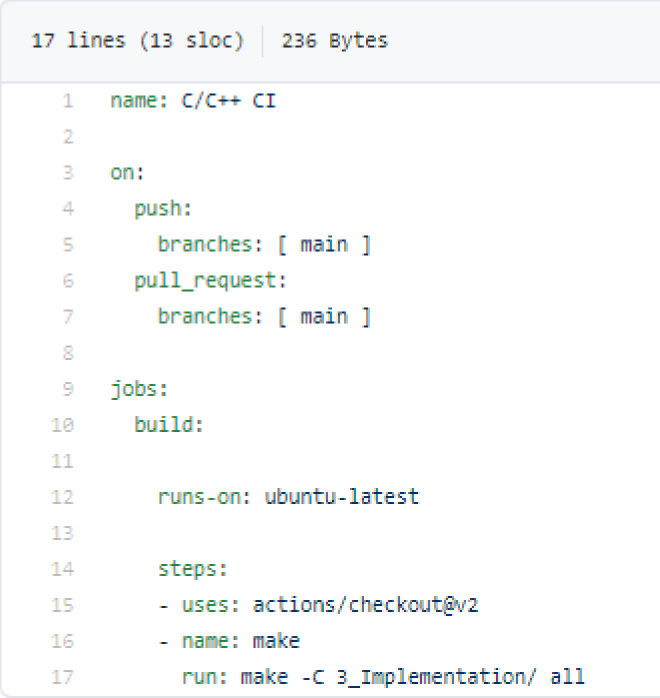
The whole project begins with the research on different calculators The Research has been divided on the basis of cost and features of different calculators. After the research the requirements for the customized calculator was documented in the form high level and low level requirements. Once the requirements were decided the same was implemented using UML diagrams. Once the UML diagrams were made for visual understanding of the design then the software implementaion was started using c code. Once the header files and source files were written then the same was tested using the test cases. Different files were combined using a single Makefile which build and run the code as a single code. After software implementation of codes CPP and unity check was done inn order to generate various badges for the correctness of the code.

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

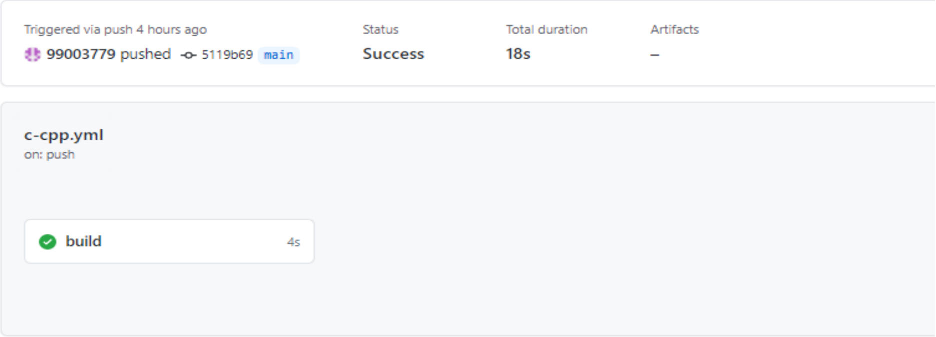
Setup Done:



17 lines (13 sloc) | 236 Bytes

```
name: C/C++ CI

on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

jobs:
  build:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v2
    - name: make
      run: make -C 3_Implementation/ all
```

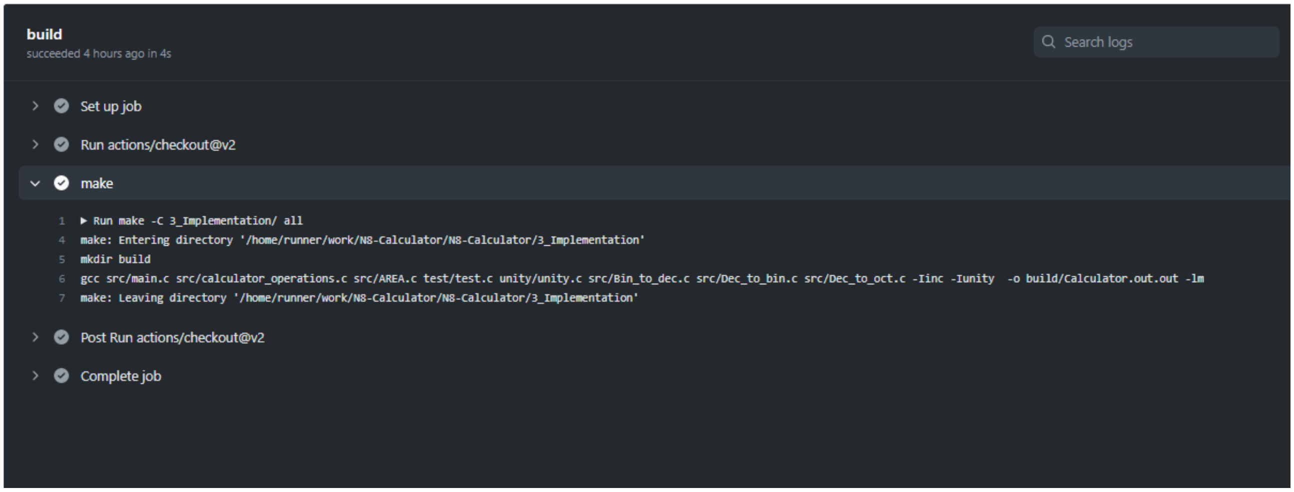
Outcome:



Triggered via push 4 hours ago	Status	Total duration	Artifacts
99003779 pushed -o- 5119b69 main	Success	18s	–

c-cpp.yml
on: push

✓ build 4s

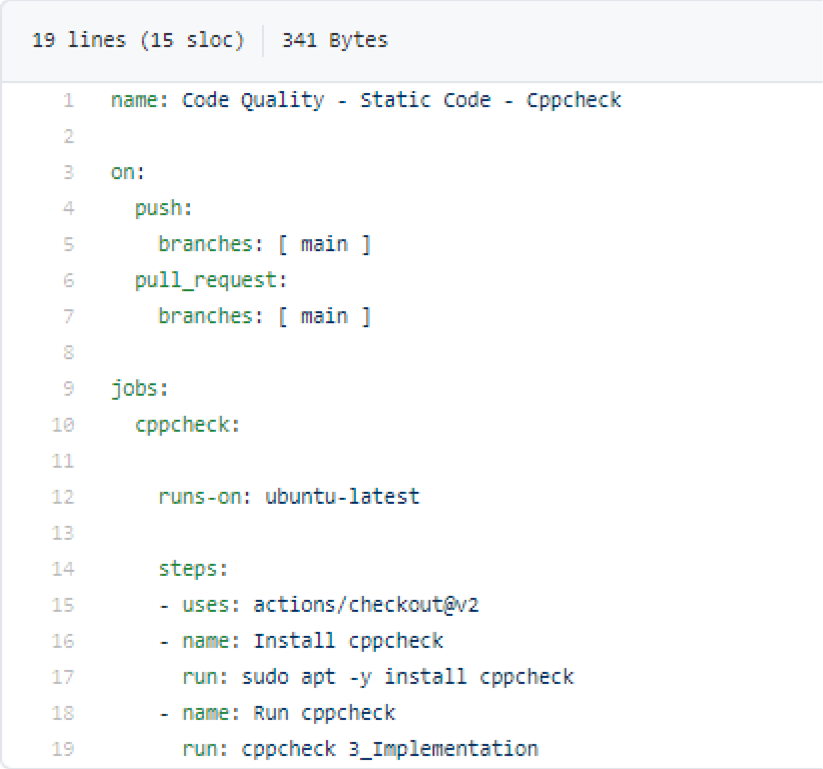


```
▶ Run make -C 3_Implementation/ all
make: Entering directory '/home/runner/work/N8-Calculator/N8-Calculator/3_Implementation'
mkdir build
gcc src/main.c src/calculator_operations.c src/AREA.c test/test.c unity/unity.c src/Bin_to_dec.c src/Dec_to_bin.c src/Dec_to_oct.c -Iinc -Iunity  -o build/Calculator.out.out -lm
make: Leaving directory '/home/runner/work/N8-Calculator/N8-Calculator/3_Implementation'
```

Code quality and Issues or Bug Tracking

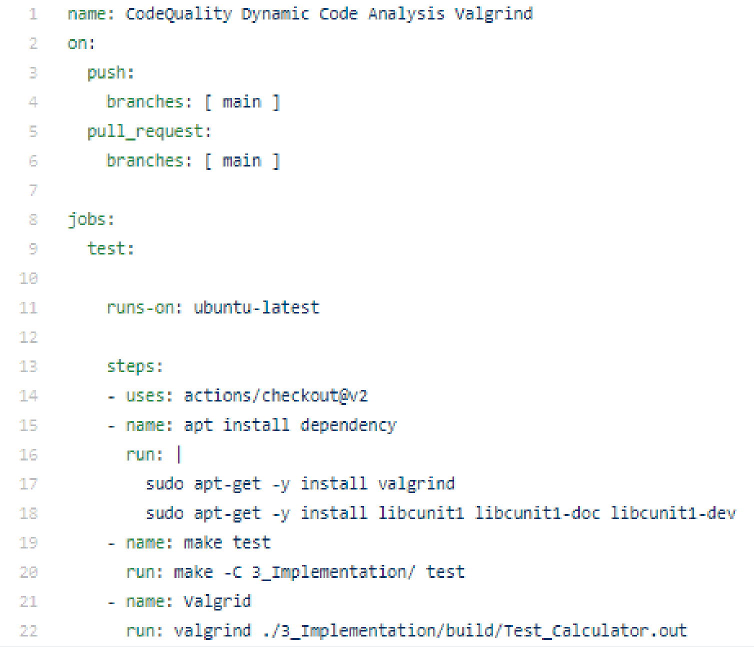
Setup:-

Static Analysis:



```yaml
name: Code Quality - Static Code - Cppcheck

on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

jobs:
  cppcheck:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v2
    - name: Install cppcheck
      run: sudo apt -y install cppcheck
    - name: Run cppcheck
      run: cppcheck 3_Implementation
```

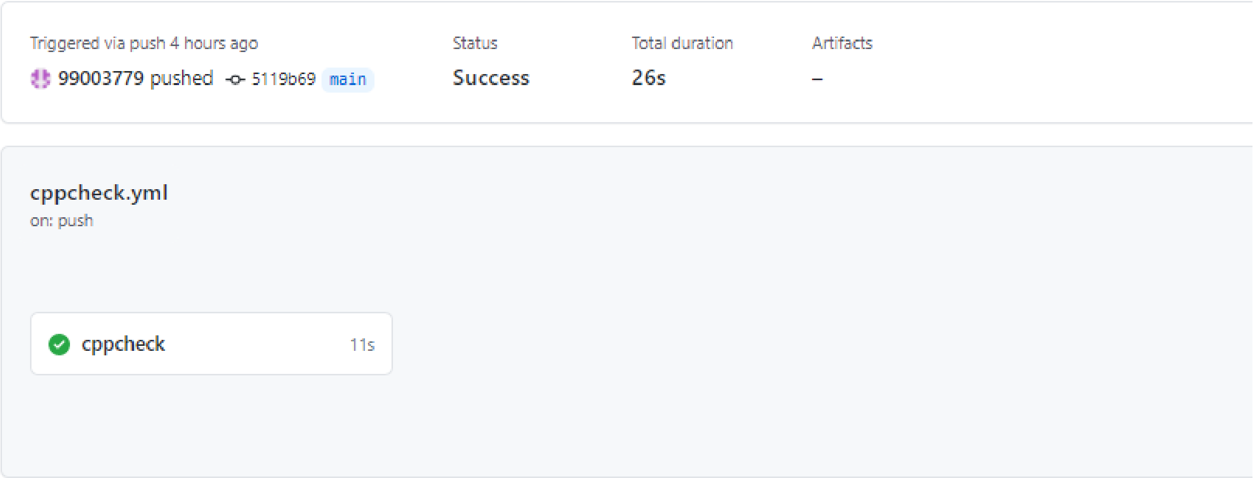
Dynamic Analysis:



```
name: CodeQuality Dynamic Code Analysis Valgrind
on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

jobs:
  test:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v2
    - name: apt install dependency
      run: |
        sudo apt-get -y install valgrind
        sudo apt-get -y install libcunit1 libcunit1-doc libcunit1-dev
    - name: make test
      run: make -C 3_Implementation/ test
    - name: Valgrid
      run: valgrind ./3_Implementation/build/Test_Calculator.out
```

Outcome:

Static Analysis:



Triggered via push 4 hours ago
99003779 pushed -o- 5119b69 main

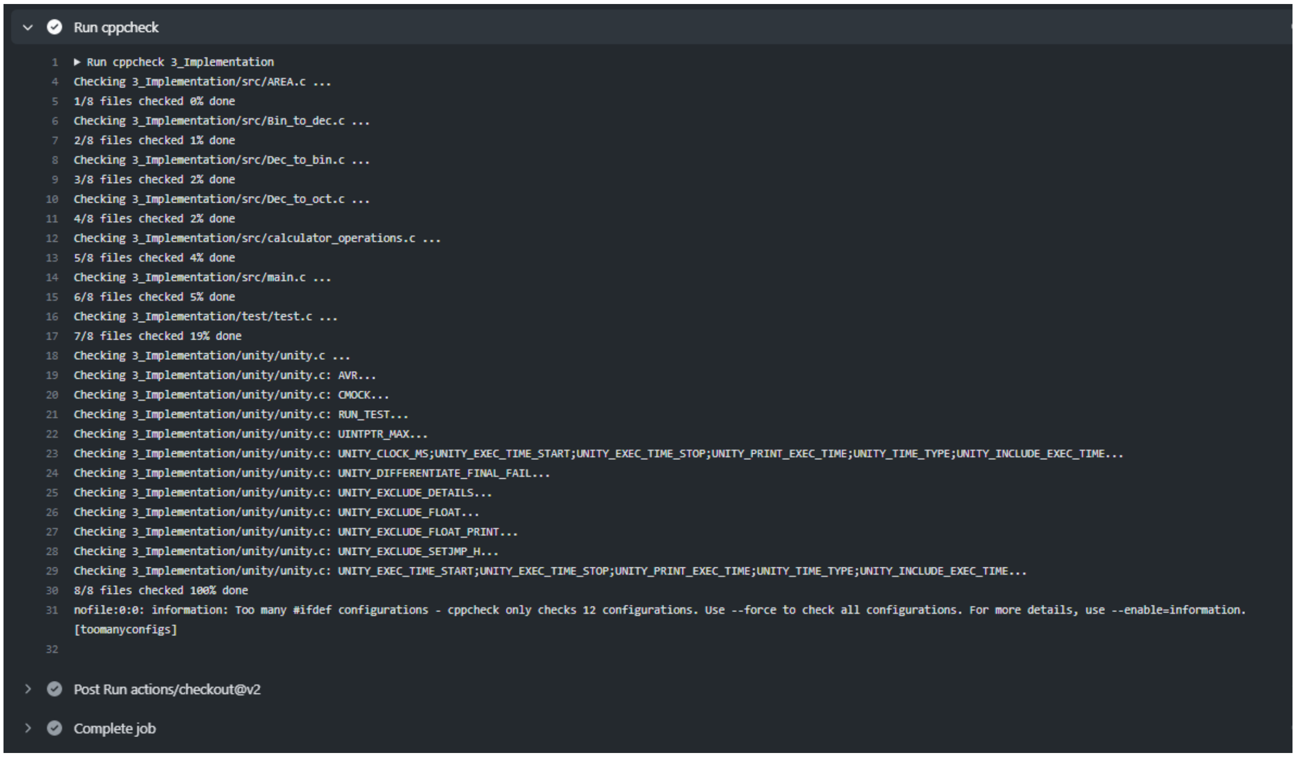
Status
Success

Total duration
26s

Artifacts
–

cppcheck.yml
on: push

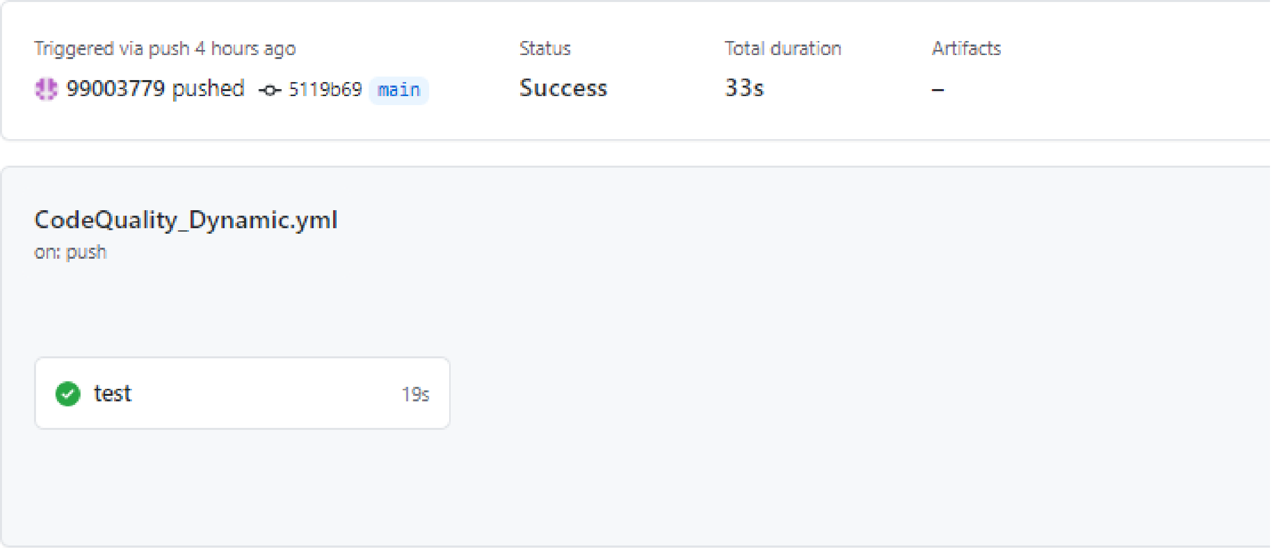
✓ cppcheck 11s



Run cppcheck

```
▶ Run cppcheck 3_Implementation
Checking 3_Implementation/src/AREA.c ...
1/8 files checked 0% done
Checking 3_Implementation/src/Bin_to_dec.c ...
2/8 files checked 1% done
Checking 3_Implementation/src/Dec_to_bin.c ...
3/8 files checked 2% done
Checking 3_Implementation/src/Dec_to_oct.c ...
4/8 files checked 2% done
Checking 3_Implementation/src/calculator_operations.c ...
5/8 files checked 4% done
Checking 3_Implementation/src/main.c ...
6/8 files checked 5% done
Checking 3_Implementation/test/test.c ...
7/8 files checked 19% done
Checking 3_Implementation/unity/unity.c ...
Checking 3_Implementation/unity/unity.c: AVR...
Checking 3_Implementation/unity/unity.c: CMOCK...
Checking 3_Implementation/unity/unity.c: RUN_TEST...
Checking 3_Implementation/unity/unity.c: UINTPTR_MAX...
Checking 3_Implementation/unity/unity.c: UNITY_CLOCK_MS;UNITY_EXEC_TIME_START;UNITY_EXEC_TIME_STOP;UNITY_PRINT_EXEC_TIME;UNITY_TIME_TYPE;UNITY_INCLUDE_EXEC_TIME...
Checking 3_Implementation/unity/unity.c: UNITY_DIFFERENTIATE_FINAL_FAIL...
Checking 3_Implementation/unity/unity.c: UNITY_EXCLUDE_DETAILS...
Checking 3_Implementation/unity/unity.c: UNITY_EXCLUDE_FLOAT...
Checking 3_Implementation/unity/unity.c: UNITY_EXCLUDE_FLOAT_PRINT...
Checking 3_Implementation/unity/unity.c: UNITY_EXCLUDE_SETJMP_H...
Checking 3_Implementation/unity/unity.c: UNITY_EXEC_TIME_START;UNITY_EXEC_TIME_STOP;UNITY_PRINT_EXEC_TIME;UNITY_TIME_TYPE;UNITY_INCLUDE_EXEC_TIME...
8/8 files checked 100% done
nofile:0:0: information: Too many #ifdef configurations - cppcheck only checks 12 configurations. Use --force to check all configurations. For more details, use --enable=information.
[toomanyconfigs]
```

Dynamic Analysis:



Triggered via push 4 hours ago: 99003779 pushed -o- 5119b69 main

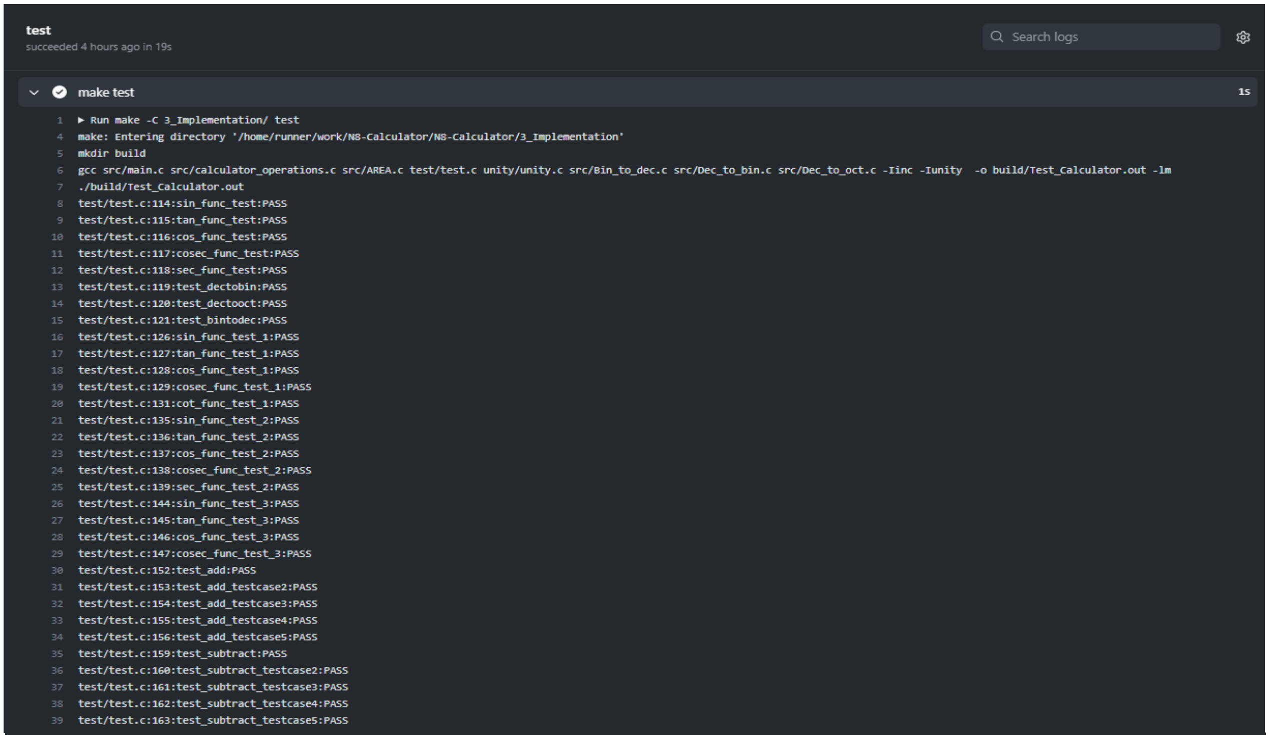
Status: Success

Total duration: 33s

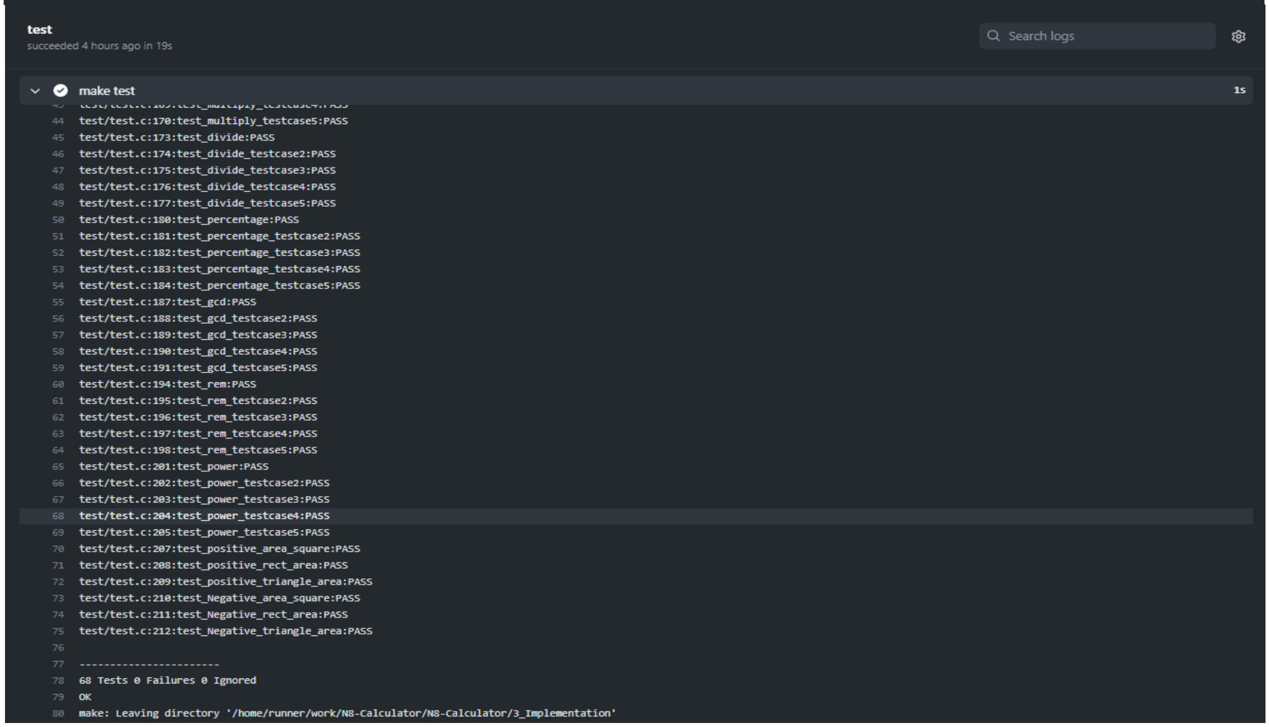
Artifacts: –

CodeQuality_Dynamic.yml
on: push

✓ test 19s



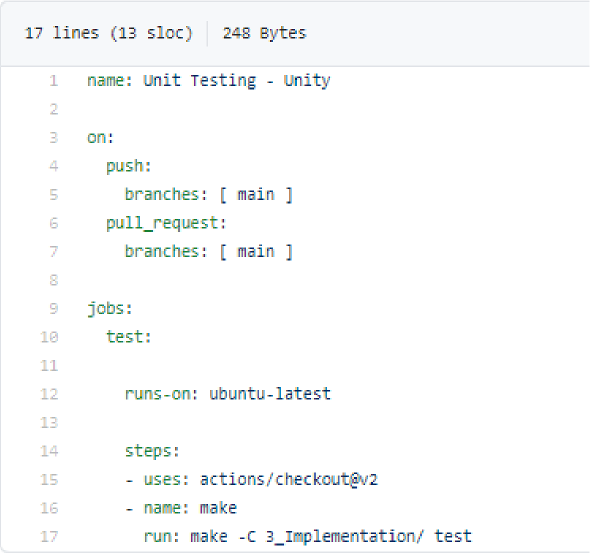
```
▶ Run make -C 3_Implementation/ test
make: Entering directory '/home/runner/work/N8-Calculator/N8-Calculator/3_Implementation'
mkdir build
gcc src/main.c src/calculator_operations.c src/AREA.c test/test.c unity/unity.c src/Bin_to_dec.c src/Dec_to_bin.c src/Dec_to_oct.c -Iinc -Iunity  -o build/Test_Calculator.out -lm
./build/Test_Calculator.out
test/test.c:114:sin_func_test:PASS
test/test.c:115:tan_func_test:PASS
test/test.c:116:cos_func_test:PASS
test/test.c:117:cosec_func_test:PASS
test/test.c:118:sec_func_test:PASS
test/test.c:119:test_dectobin:PASS
test/test.c:120:test_dectooct:PASS
test/test.c:121:test_bintodec:PASS
test/test.c:126:sin_func_test_1:PASS
test/test.c:127:tan_func_test_1:PASS
test/test.c:128:cos_func_test_1:PASS
test/test.c:129:cosec_func_test_1:PASS
test/test.c:131:cot_func_test_1:PASS
test/test.c:135:sin_func_test_2:PASS
test/test.c:136:tan_func_test_2:PASS
test/test.c:137:cos_func_test_2:PASS
test/test.c:138:cosec_func_test_2:PASS
test/test.c:139:sec_func_test_2:PASS
test/test.c:144:sin_func_test_3:PASS
test/test.c:145:tan_func_test_3:PASS
test/test.c:146:cos_func_test_3:PASS
test/test.c:147:cosec_func_test_3:PASS
test/test.c:152:test_add:PASS
test/test.c:153:test_add_testcase2:PASS
test/test.c:154:test_add_testcase3:PASS
test/test.c:155:test_add_testcase4:PASS
test/test.c:156:test_add_testcase5:PASS
test/test.c:159:test_subtract:PASS
test/test.c:160:test_subtract_testcase2:PASS
test/test.c:161:test_subtract_testcase3:PASS
test/test.c:162:test_subtract_testcase4:PASS
test/test.c:163:test_subtract_testcase5:PASS
```



```
test/test.c:169:test_multiply_testcase4:PASS
test/test.c:170:test_multiply_testcase5:PASS
test/test.c:173:test_divide:PASS
test/test.c:174:test_divide_testcase2:PASS
test/test.c:175:test_divide_testcase3:PASS
test/test.c:176:test_divide_testcase4:PASS
test/test.c:177:test_divide_testcase5:PASS
test/test.c:180:test_percentage:PASS
test/test.c:181:test_percentage_testcase2:PASS
test/test.c:182:test_percentage_testcase3:PASS
test/test.c:183:test_percentage_testcase4:PASS
test/test.c:184:test_percentage_testcase5:PASS
test/test.c:187:test_gcd:PASS
test/test.c:188:test_gcd_testcase2:PASS
test/test.c:189:test_gcd_testcase3:PASS
test/test.c:190:test_gcd_testcase4:PASS
test/test.c:191:test_gcd_testcase5:PASS
test/test.c:194:test_rem:PASS
test/test.c:195:test_rem_testcase2:PASS
test/test.c:196:test_rem_testcase3:PASS
test/test.c:197:test_rem_testcase4:PASS
test/test.c:198:test_rem_testcase5:PASS
test/test.c:201:test_power:PASS
test/test.c:202:test_power_testcase2:PASS
test/test.c:203:test_power_testcase3:PASS
test/test.c:204:test_power_testcase4:PASS
test/test.c:205:test_power_testcase5:PASS
test/test.c:207:test_positive_area_square:PASS
test/test.c:208:test_positive_rect_area:PASS
test/test.c:209:test_positive_triangle_area:PASS
test/test.c:210:test_Negative_area_square:PASS
test/test.c:211:test_Negative_rect_area:PASS
test/test.c:212:test_Negative_triangle_area:PASS

-----------------------
68 Tests 0 Failures 0 Ignored
OK
make: Leaving directory '/home/runner/work/N8-Calculator/N8-Calculator/3_Implementation'
```

Unit Testing

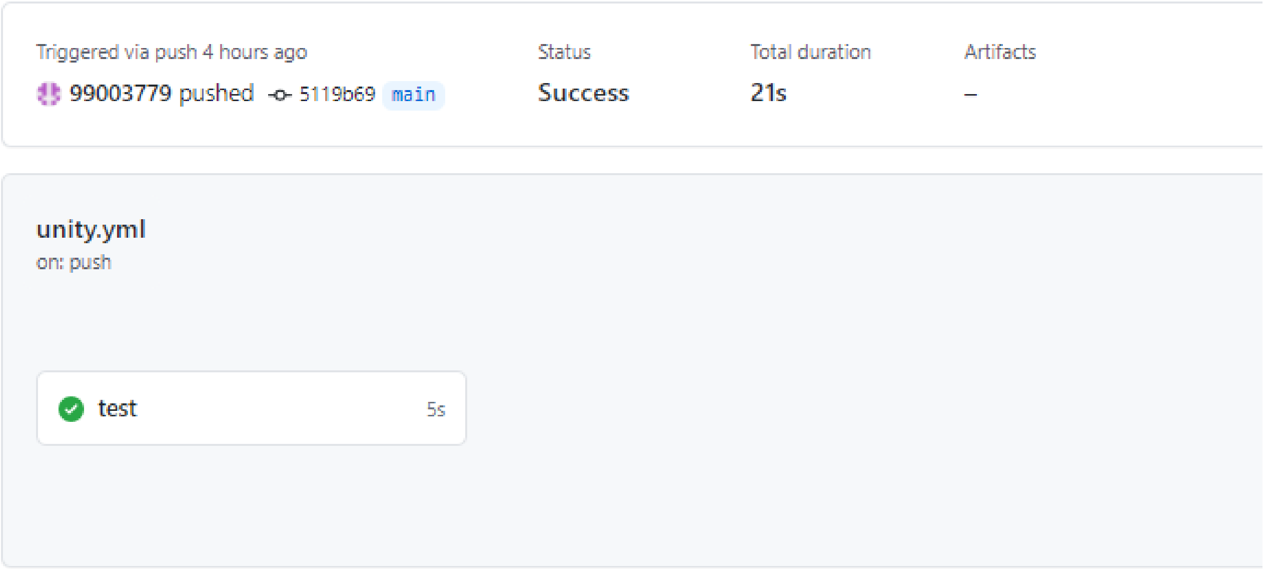
Setup:



17 lines (13 sloc) | 248 Bytes

```yaml
name: Unit Testing - Unity

on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

jobs:
  test:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v2
    - name: make
      run: make -C 3_Implementation/ test
```

Outcome:



Triggered via push 4 hours ago
99003779 pushed 5119b69 main

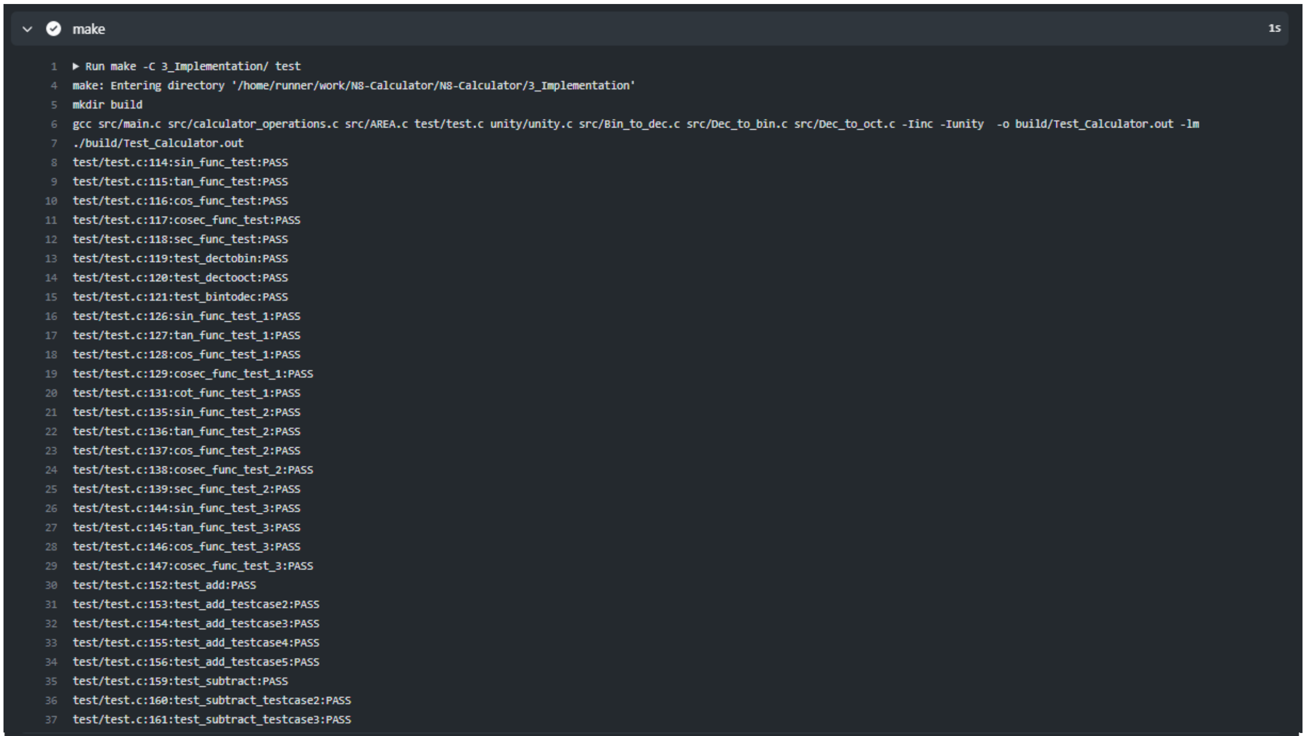
Status
Success

Total duration
21s

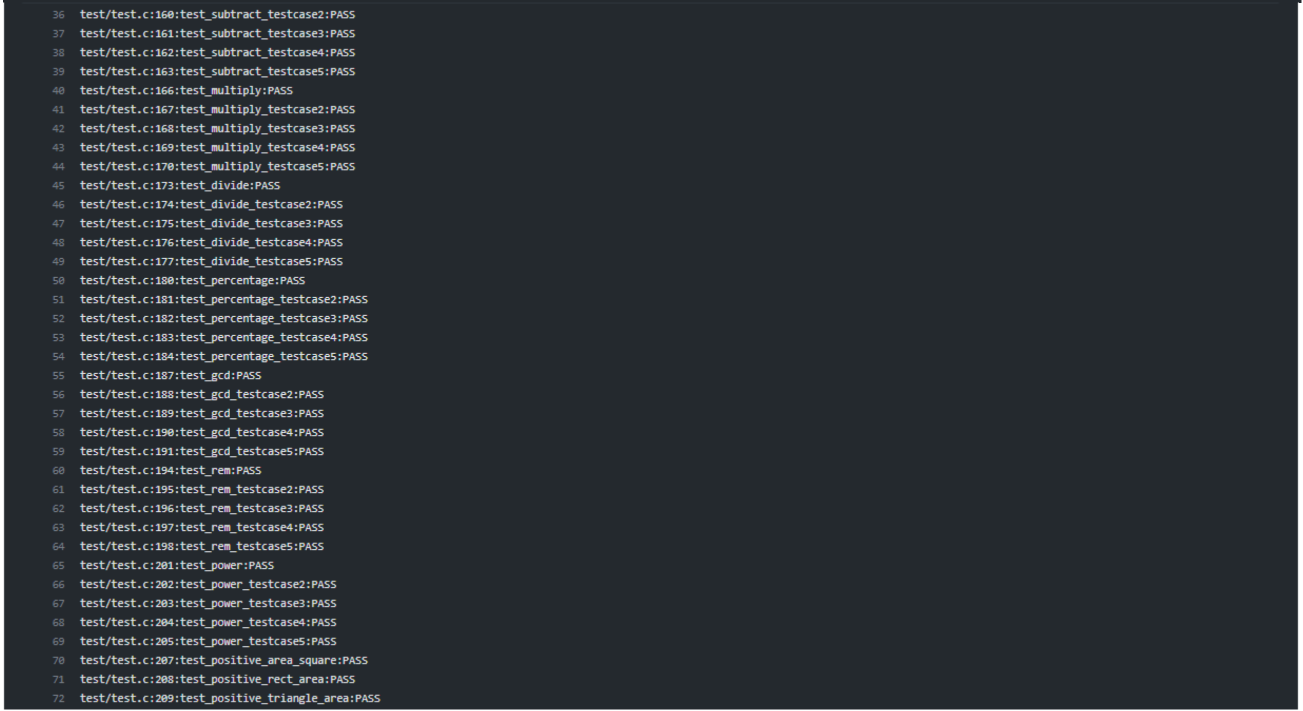
Artifacts
–

unity.yml
on: push

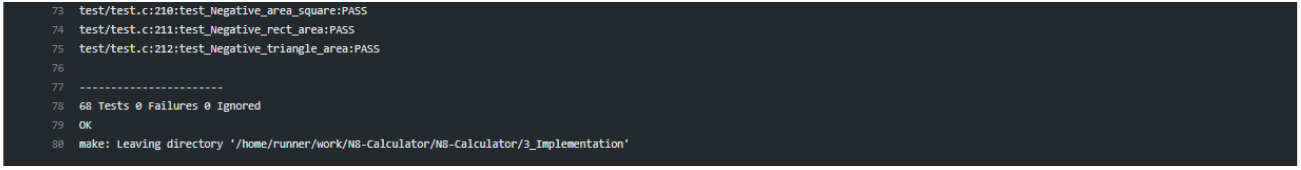
✓ test 5s



```
▶ Run make -C 3_Implementation/ test
make: Entering directory '/home/runner/work/N8-Calculator/N8-Calculator/3_Implementation'
mkdir build
gcc src/main.c src/calculator_operations.c src/AREA.c test/test.c unity/unity.c src/Bin_to_dec.c src/Dec_to_bin.c src/Dec_to_oct.c -Iinc -Iunity  -o build/Test_Calculator.out -lm
./build/Test_Calculator.out
test/test.c:114:sin_func_test:PASS
test/test.c:115:tan_func_test:PASS
test/test.c:116:cos_func_test:PASS
test/test.c:117:cosec_func_test:PASS
test/test.c:118:sec_func_test:PASS
test/test.c:119:test_dectobin:PASS
test/test.c:120:test_dectooct:PASS
test/test.c:121:test_bintodec:PASS
test/test.c:126:sin_func_test_1:PASS
test/test.c:127:tan_func_test_1:PASS
test/test.c:128:cos_func_test_1:PASS
test/test.c:129:cosec_func_test_1:PASS
test/test.c:131:cot_func_test_1:PASS
test/test.c:135:sin_func_test_2:PASS
test/test.c:136:tan_func_test_2:PASS
test/test.c:137:cos_func_test_2:PASS
test/test.c:138:cosec_func_test_2:PASS
test/test.c:139:sec_func_test_2:PASS
test/test.c:144:sin_func_test_3:PASS
test/test.c:145:tan_func_test_3:PASS
test/test.c:146:cos_func_test_3:PASS
test/test.c:147:cosec_func_test_3:PASS
test/test.c:152:test_add:PASS
test/test.c:153:test_add_testcase2:PASS
test/test.c:154:test_add_testcase3:PASS
test/test.c:155:test_add_testcase4:PASS
test/test.c:156:test_add_testcase5:PASS
test/test.c:159:test_subtract:PASS
test/test.c:160:test_subtract_testcase2:PASS
test/test.c:161:test_subtract_testcase3:PASS
```



```
test/test.c:160:test_subtract_testcase2:PASS
test/test.c:161:test_subtract_testcase3:PASS
test/test.c:162:test_subtract_testcase4:PASS
test/test.c:163:test_subtract_testcase5:PASS
test/test.c:166:test_multiply:PASS
test/test.c:167:test_multiply_testcase2:PASS
test/test.c:168:test_multiply_testcase3:PASS
test/test.c:169:test_multiply_testcase4:PASS
test/test.c:170:test_multiply_testcase5:PASS
test/test.c:173:test_divide:PASS
test/test.c:174:test_divide_testcase2:PASS
test/test.c:175:test_divide_testcase3:PASS
test/test.c:176:test_divide_testcase4:PASS
test/test.c:177:test_divide_testcase5:PASS
test/test.c:180:test_percentage:PASS
test/test.c:181:test_percentage_testcase2:PASS
test/test.c:182:test_percentage_testcase3:PASS
test/test.c:183:test_percentage_testcase4:PASS
test/test.c:184:test_percentage_testcase5:PASS
test/test.c:187:test_gcd:PASS
test/test.c:188:test_gcd_testcase2:PASS
test/test.c:189:test_gcd_testcase3:PASS
test/test.c:190:test_gcd_testcase4:PASS
test/test.c:191:test_gcd_testcase5:PASS
test/test.c:194:test_rem:PASS
test/test.c:195:test_rem_testcase2:PASS
test/test.c:196:test_rem_testcase3:PASS
test/test.c:197:test_rem_testcase4:PASS
test/test.c:198:test_rem_testcase5:PASS
test/test.c:201:test_power:PASS
test/test.c:202:test_power_testcase2:PASS
test/test.c:203:test_power_testcase3:PASS
test/test.c:204:test_power_testcase4:PASS
test/test.c:205:test_power_testcase5:PASS
test/test.c:207:test_positive_area_square:PASS
test/test.c:208:test_positive_rect_area:PASS
test/test.c:209:test_positive_triangle_area:PASS
```



```
test/test.c:210:test_Negative_area_square:PASS
test/test.c:211:test_Negative_rect_area:PASS
test/test.c:212:test_Negative_triangle_area:PASS

-----------------------
68 Tests 0 Failures 0 Ignored
OK
make: Leaving directory '/home/runner/work/N8-Calculator/N8-Calculator/3_Implementation'
```

Individual Contribution & Highlights

Contributions by me for the Team:

Arithmetic Operations:

- Implements various arithmetic operations which includes addition, subtraction, multiplication, division, remainder and GCD
- Implements low-level and high-level designs
- Contributed for low-level and high-level requirements in arithmetic operations
- Write various test cases for the same.

Percentage and Power:

- Implements percentage and power operation functions.
- Implements low-level and high-level designs
- Wrote various test cases for the same.

Summary

Hardware is not implemented yet. Only software portion has been implemented.

Challenges faced and how were they overcome

1. Found difficulty in writing the make file but with the help of collegues we are able to make our own make file
2. Initially it was difficult to understand the functions in test cases, but with self and group study we resolved the problems
3. Code debugging was a bit time consuming, but with brainstorming we excel it.
4. Integration of individual codes was difficult to understand, but with peer support, we did it.

Future Scope (If applicable):

NA

Miniproject -2 [Individual]: Python

Reading Excel Sheet and summarizing the data in the mastersheet

Module/s:

Modules linked to this mini-project is Python and SDLC

Topic and Subtopics

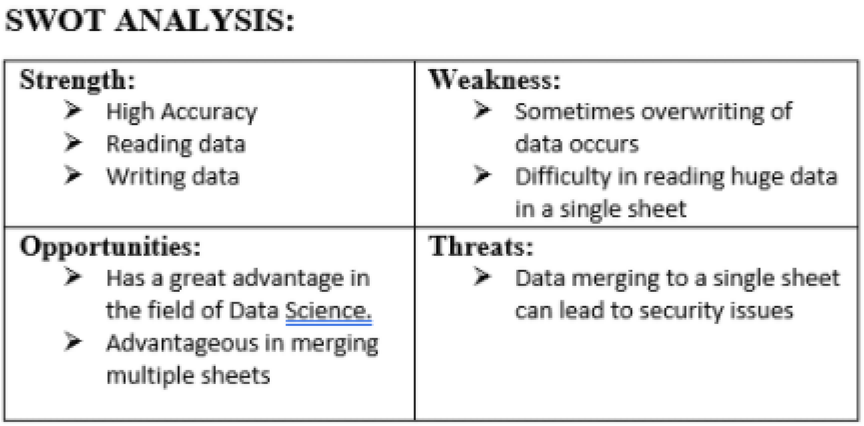
Objectives & Requirements

Objective:

The main objective of the designed code is to read and write data. The code implemented, has considered the basic requirement of reading and writing data with user friendly environment.
An excel sheet has been made manually which consists of 5 sheets, 1 master sheet and 1 summary sheet. Here we are searching details of an individual in all the 5 sub-sheets corresponding to :

Name:
Registration number:
Email ID:

Once the data has been fetched from the sub sheets then it will be printed to the master sheet. The excel sheet also consists of a summary sheet which indicates the count number of data fetched from each sheet. The whole implementation is used to read a file for better searching and writing. The code makes the study easier in the field of data science where lots and lots of data needs extraction.



SWOT ANALYSIS:

Strength: ➢ High Accuracy ➢ Reading data ➢ Writing data	**Weakness:** ➢ Sometimes overwriting of data occurs ➢ Difficulty in reading huge data in a single sheet
Opportunities: ➢ Has a great advantage in the field of Data Science. ➢ Advantageous in merging multiple sheets	**Threats:** ➢ Data merging to a single sheet can lead to security issues

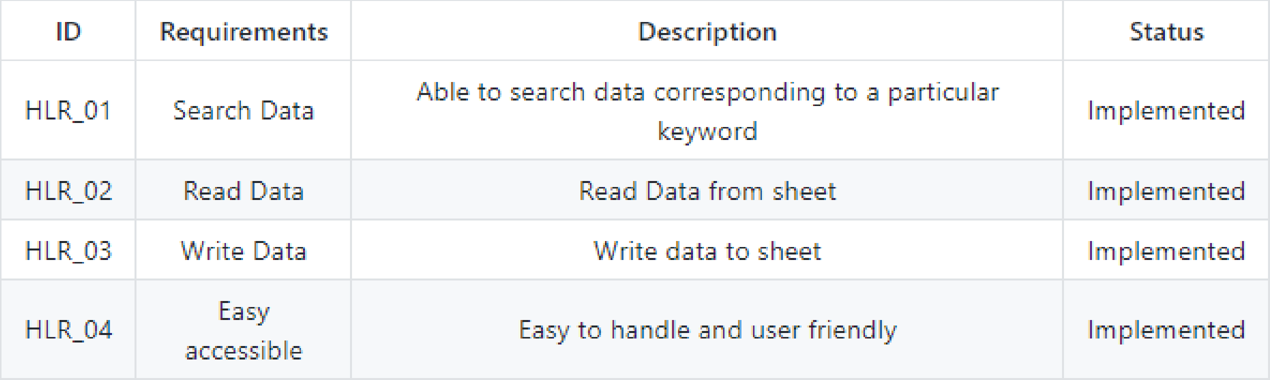
4W's and 1H:-
Who:- Basically used in research field and data science fields where large data needs manipulation and extraction.
What:- Xlsx file or CSV file with python code to read and write data
When:- Used when large data needs to manipulated and extracted.
Where:- In Research and technical fields

How:- User friendly and easily accessible.

Requirements:

High-Level:

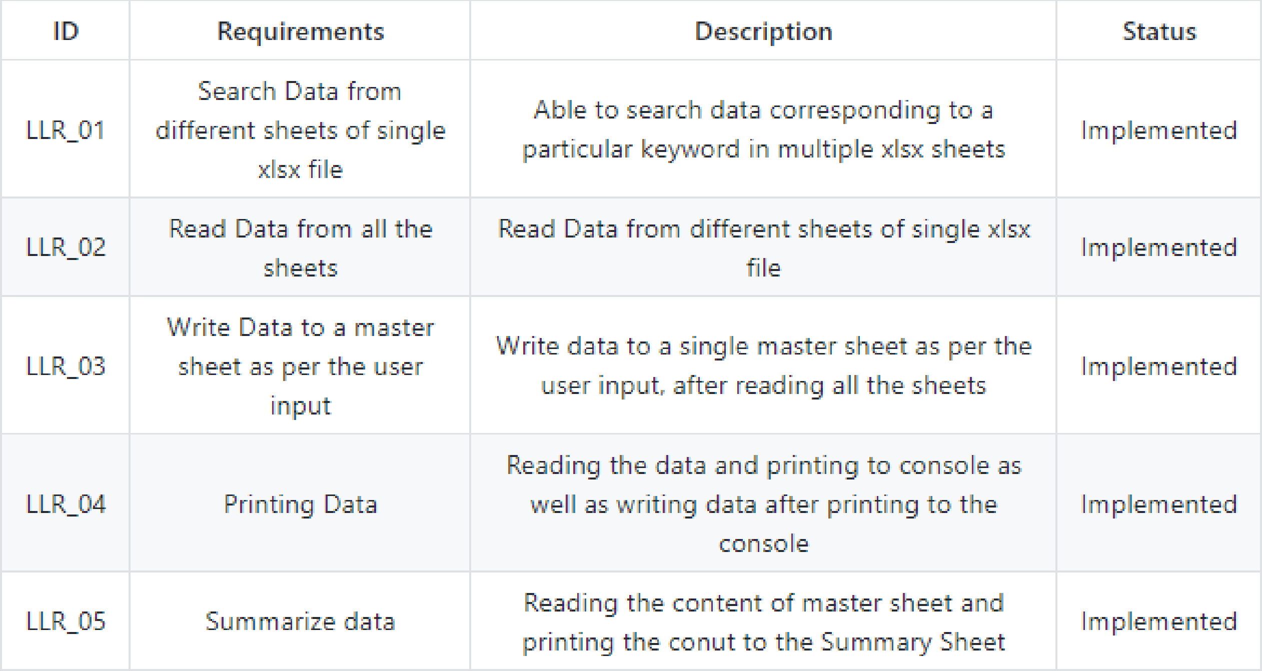
Table 11: HLR of data read/write



ID	Requirements	Description	Status
HLR_01	Search Data	Able to search data corresponding to a particular keyword	Implemented
HLR_02	Read Data	Read Data from sheet	Implemented
HLR_03	Write Data	Write data to sheet	Implemented
HLR_04	Easy accessible	Easy to handle and user friendly	Implemented

Low-Level:

Table 12: LLR requirement of data read/write



ID	Requirements	Description	Status
LLR_01	Search Data from different sheets of single xlsx file	Able to search data corresponding to a particular keyword in multiple xlsx sheets	Implemented
LLR_02	Read Data from all the sheets	Read Data from different sheets of single xlsx file	Implemented
LLR_03	Write Data to a master sheet as per the user input	Write data to a single master sheet as per the user input, after reading all the sheets	Implemented
LLR_04	Printing Data	Reading the data and printing to console as well as writing data after printing to the console	Implemented
LLR_05	Summarize data	Reading the content of master sheet and printing the conut to the Summary Sheet	Implemented

Design

Structural Diagram: Class Diagram

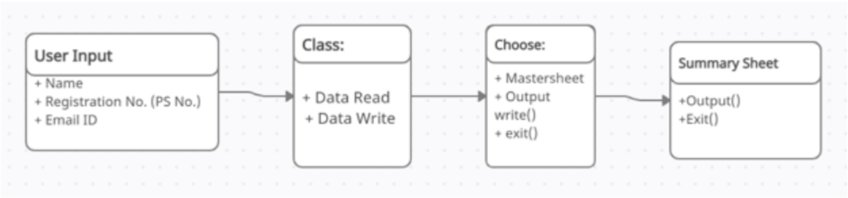


Figure 12: Structural Diagram of data read/write

Behavioral Diagram: Activity Diagram

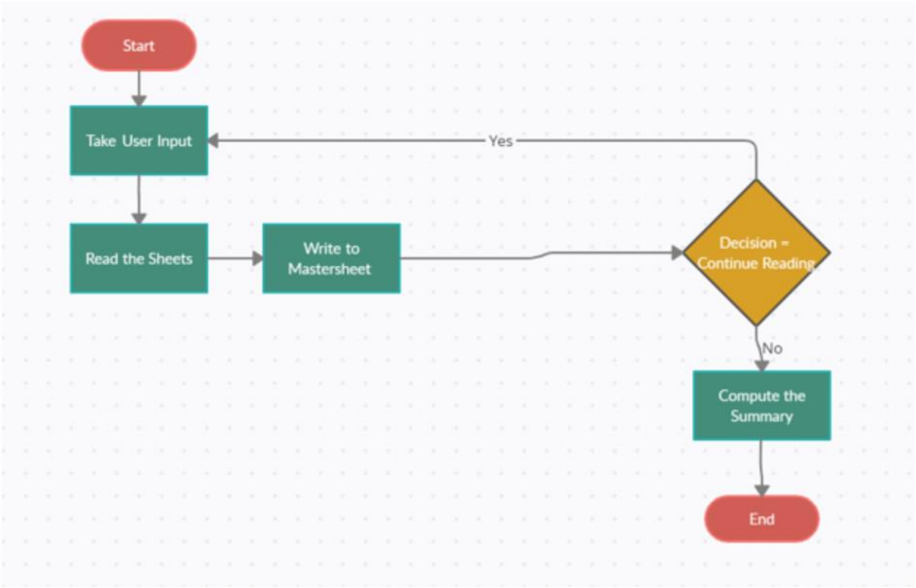


Figure 13: Behavioural Diagram of Data read/write

V – Model:

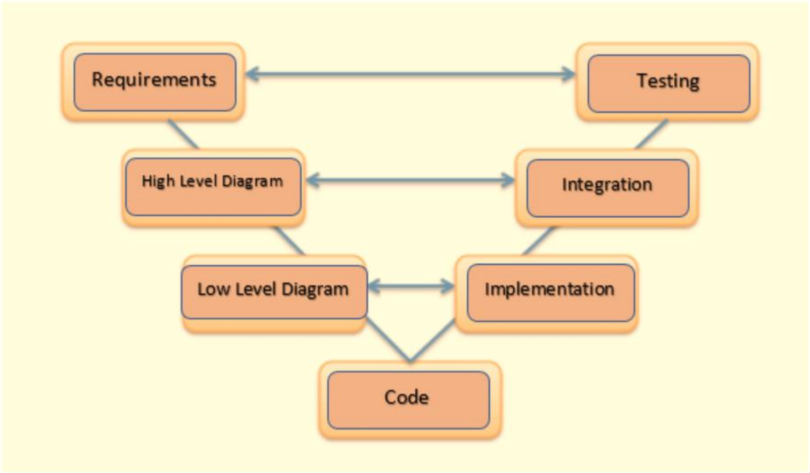


Figure 14: V-model Implementation of data read/write

Test Plan

Table 13: Test Plan of Data read/write

Test_ID	Description	Expected Input	Expected Output	Actual Output
TP_01	User enters the registration number, name and Email ID of the person being searched	99673798 Ali Adibi Ali.adibi@ece.gatech.edu	The data corresponding to the given input will be searched in all the sub-sheets and printed to mastersheet.	The data is printed to mastersheet and summary count has been incremented.
TP_02	User enters the registration number, name and Email ID of the person being searched. Now it takes multiple input of data by selecting Yes/No in the terminal window.	99673798 Ali Adibi Ali.adibi@ece.gatech.edu Continue (y/n) ? y 99673786 Professor Peter Y. K. Cheung p.cheung@imperial.ac.uk	The data corresponding to the given multiple inputs will be searched in all the sub-sheets and printed to mastersheet.	The data is printed to mastersheet and summary count has been incremented.
TP_03	User enters the registration number, name and Email ID of the person being searched. Now it takes multiple input of data by selecting Yes/No in the terminal window. Now user select no for further data intake.	99673798 Ali Adibi Ali.adibi@ece.gatech.edu Continue (y/n) ? y 99673786 Professor Peter Y. K. Cheung p.cheung@imperial.ac.uk Continue (y/n) ? n	The data corresponding to the given multiple inputs will be searched in all the sub-sheets and printed to mastersheet. Data has been printed to Mastersheet	The data is printed to mastersheet and summary count has been incremented.

Implementation Summary:

The user provides data of the person being searched which includes:

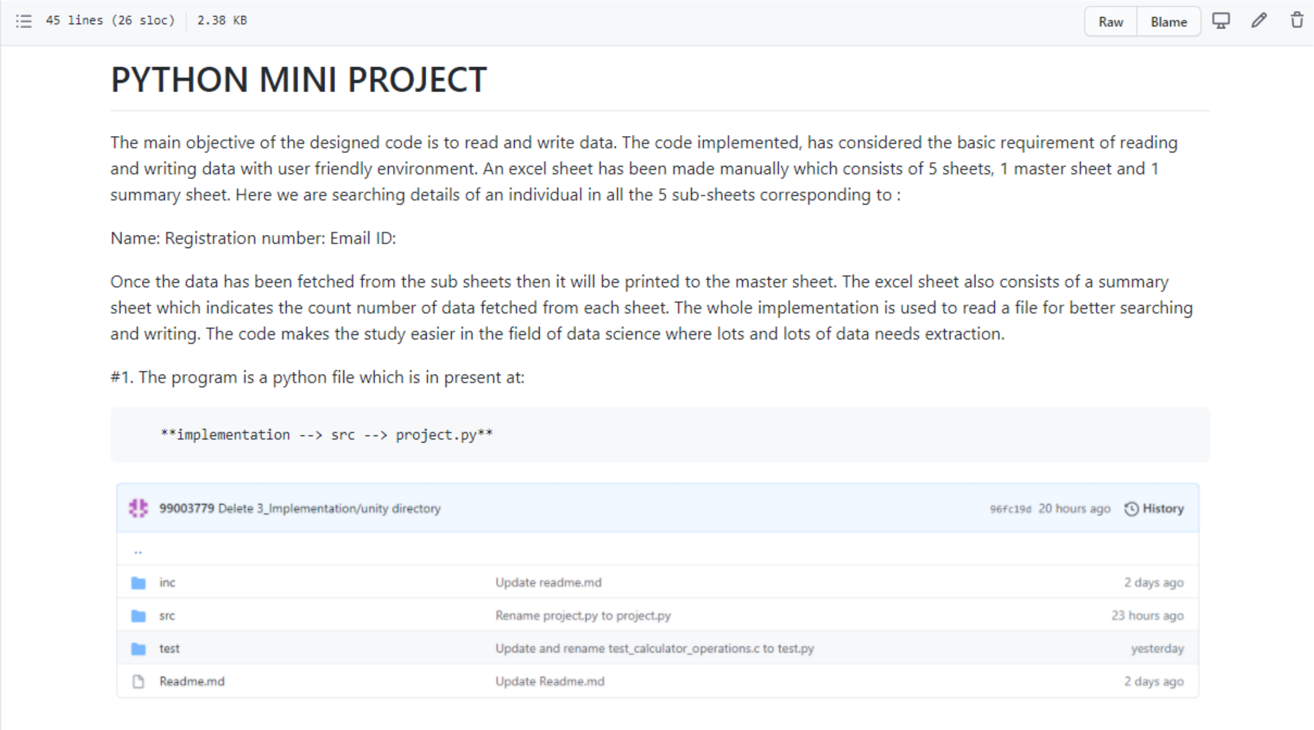
- Registration number:
- Name:
- Email ID:

The designed python code will search for the provided data in all the sheets of a single excel file and summarize the complete data of that person in a single excel Mastersheet. Once all the data of a single person or multiple person has been printed to master sheet then the summary sheet will summarize the total number count of the mastersheet. The detailed steps are on Git Link shown below.

Git Link

https://github.com/99003779/Python_Mini_Project.git

Git Dashboard



PYTHON MINI PROJECT

The main objective of the designed code is to read and write data. The code implemented, has considered the basic requirement of reading and writing data with user friendly environment. An excel sheet has been made manually which consists of 5 sheets, 1 master sheet and 1 summary sheet. Here we are searching details of an individual in all the 5 sub-sheets corresponding to :

Name: Registration number: Email ID:

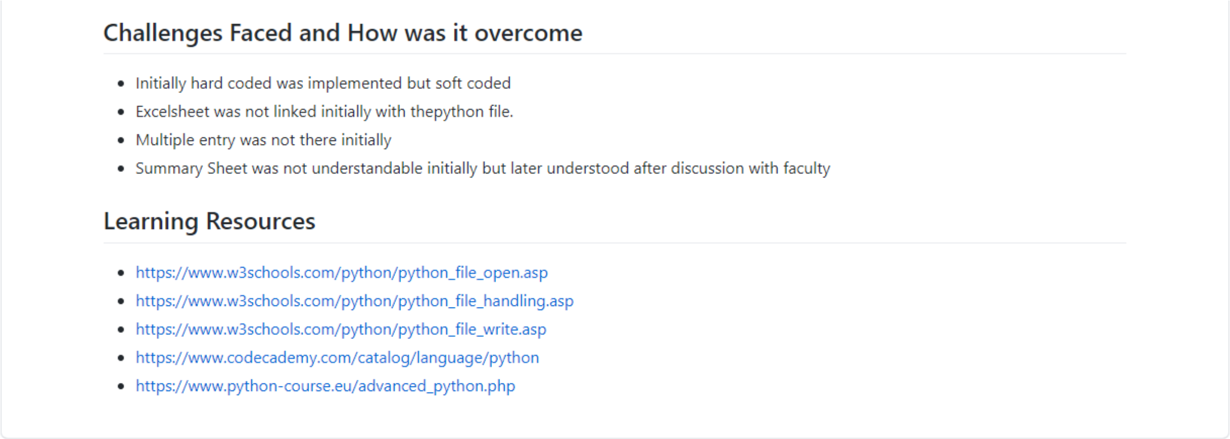
Once the data has been fetched from the sub sheets then it will be printed to the master sheet. The excel sheet also consists of a summary sheet which indicates the count number of data fetched from each sheet. The whole implementation is used to read a file for better searching and writing. The code makes the study easier in the field of data science where lots and lots of data needs extraction.

#1. The program is a python file which is in present at:

```
    **implementation --> src --> project.py**
```

99003779 Delete 3_Implementation/unity directory — 96fc19d 20 hours ago — History

..		
inc	Update readme.md	2 days ago
src	Rename project.py to project.py	23 hours ago
test	Update and rename test_calculator_operations.c to test.py	yesterday
Readme.md	Update Readme.md	2 days ago



Challenges Faced and How was it overcome

- Initially hard coded was implemented but soft coded
- Excelsheet was not linked initially with thepython file.
- Multiple entry was not there initially
- Summary Sheet was not understandable initially but later understood after discussion with faculty

Learning Resources

- https://www.w3schools.com/python/python_file_open.asp
- https://www.w3schools.com/python/python_file_handling.asp
- https://www.w3schools.com/python/python_file_write.asp
- https://www.codecademy.com/catalog/language/python
- https://www.python-course.eu/advanced_python.php

Summary

In this python mini-project, 5 sheets were created in a single excel file which contains data of a delegates. It sheet contains 40 rows with 10 columns. The user provides data of the person being searched which includes:

- Registration number:
- Name:
- Email ID:

The designed python code will search for the provided data in all the sheets of a single excel file and summarize the complete data of that person in a single excel Mastersheet. Once all the data of a single person or multiple person has been printed to master sheet then the summary sheet will summarize the total number count of the mastersheet.

Individual Contribution & Highlights:

The whole python mini project is a single and individual project .

Challenges faced and how were they overcome

- Initially hard coded was implemented but soft coded
- Excel sheet was not linked initially with the python file.
- Multiple entry was not there initially
- Summary Sheet was not understandable initially but later understood after discussion with faculty

Miniproject -3 [Team] – Embedded C

Module/s

Modules linked to this miniproject is Embedded C

Topic and Subtopics

Driver Designing
GPIO Programming
I2C, SPI, UART and ADC Protocols
Hardware Abstraction Layer (HAL) Application interface

Objectives & Requirements

Objective:

The main objective of the project is design a Body Control Module (BCM) with some features using STM32F407VG microcontroller with 32-bit ARM cortex-M4 FPU core processor.

A control module is a computer component in a car that monitors, controls and uses electrical devices throughout the vehicle. When electric devices began to be used in cars and trucks, each tool was controlled by a different electrical component. There was a cooling module, an interior lighting module, a door lock module, and so on. The body control module combines all these different modules under a single system to work together instead of partitioning, a simple format for both production and problem solving. While there are many different types of body control modules, they are generally the same, the sensors for connecting, switching and automatic switching together in a single computer system. These components are subdivided into inputs, such as sensor data about temperature or speed, and the effects, or the way a computer control system responds to control motor performance. Inputs and outputs are further divided into analog and digital information types - analog signals used by modular continuous modules (such as oil pressure) and digital signals used for modules that can either turn on or off (such as headlights or oil indicator light).

Requirement:

In our project we have implemented six features on a microcontroller STM32f4 discovery board. In the project we integrated IR Sensor, thermal sensor, Gas sensor (mq7), LDR sensor, Moisture sensor, ignition sensor with microcontroller STM32f4 using GPIO protocol.

- IR Sensor:-An IR sensor is an electronic device, which illuminates light to detect the surrounding environment. The IR sensor can detect movement. Usually, in the infrared spectrum, all substances emit a certain type of radiation. These types of radiation are invisible to our eyes, but the infrared sensor can detect these rays.
- The gas sensor receives the attention of the liquor gas in the air and the analog voltage reading output. The sensor can operate at temperatures ranging from -10 to 50 ° C with an electrical power of less than 150 Ma to 5V. Sensitivity range ranges from 0.04 mg / L to 4 mg / L, suitable for breathalyzers.
- LDR or light-resistant light resistor is also known as photo resistor, photocell, photoconductor. One type of opposition its resistance varies depending on the amount of light falling on its surface. When light falls on an opponent, the resistance changes.
- Moisture sensors measure the amount of water in the soil. Since direct gravimetric measurement of free soil moisture requires removal, drying, and weight of the sample, soil moisture sensors measure water content indirectly using other earth materials, such as electricity resistance, dielectric constant, or neutron contact, as representative of moisture content.

- The ignition system produces sparks or burns the electrode at high temperatures to heat a mixture of gasoline gas in combustion engines fire engines, oil-fired boilers and gas, rocket engines, etc.
- Thermal sensors detect a change in a physical parameter such as resistance or output voltage that corresponds to a temperature change.

High Level Requirements:

Table 14: HLR of hardware sensor project

ID	Requirements	Description	Status
HLR_01	Enable control Module	Control Module should be enabled in order to detect different sensors.	Implemented
HLR_02	Temperature sensor	Able to detect temperature	Implemented
HLR_03	Gas Sensor	Able to detect gas leakage	Implemented
HLR_04	LDR (light dependent resistor) Sensor	Able to detect higher intensity of light	Implemented
HLR_05	Moisture Sensor	Able to detect moisture	Implemented
HLR_06	IR Sensor	Able to detect obstacles	Implemented

Low Level Requirements:

Table 15: LLR of hardware sensor project

ID	Requirements	Description	Status
LLR_01	Implementation of air Conditioning Control module	Thermal sensor detects the high temperature and AC switches ON.	Implemented
LLR_02	Implementation of gas leakage control module	Gas sensor detects gas leakage and glows LED which prevents spontaneous car explosion.	Implemented
LLR_03	Implementation of shed controller module	LDR sensor enables the intensity of sunlight and the shed is enabled when the sunlight is too bright	Implemented
LLR_04	Implementation of Headlight control module	LDR sensor senses darkness and headlight turns ON.	Implemented
LLR_05	Implementation of Wiper control module	Moisture sensor senses water in wiper and LED switches ON	Implemented
LLR_06	Implementation of obstacle detector control module	LED glows when some obstacle is detected	Implemented

Design

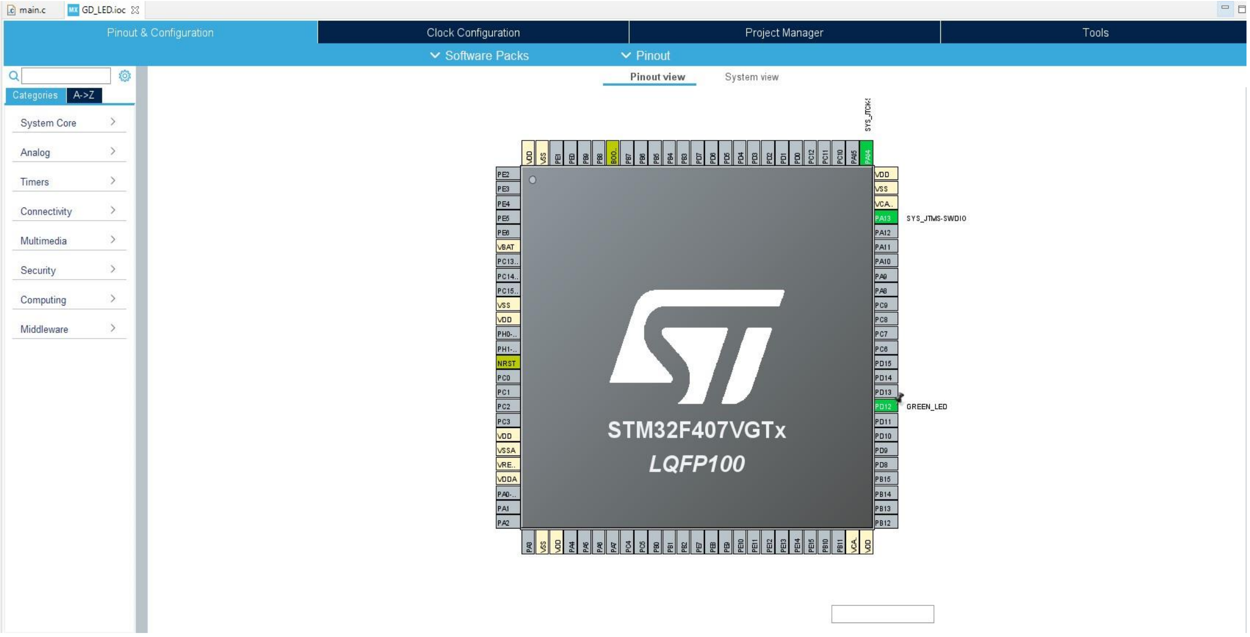


Figure 15: High Level Design of hardware

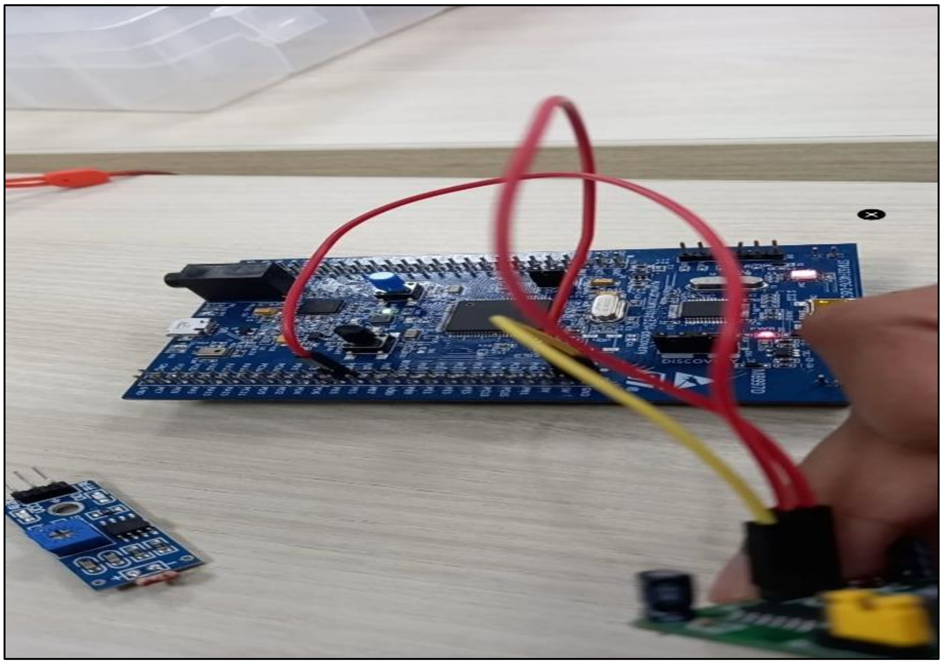


Figure 16:Air Conditioner Control Module

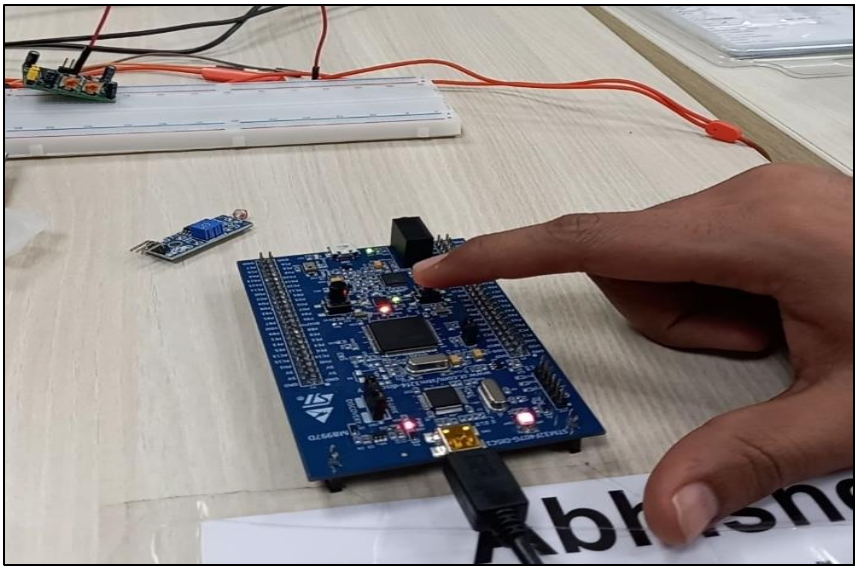


Figure 17: Sun Shed Control Module

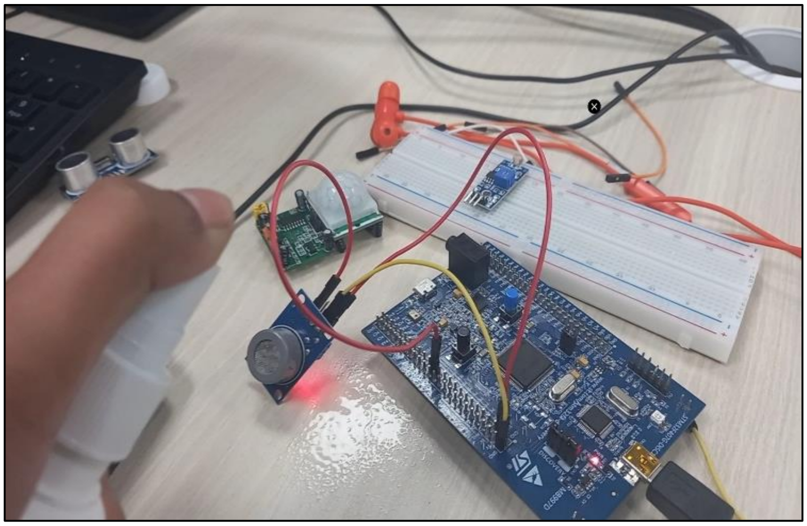


Figure 18: Gas Leakage Control Module

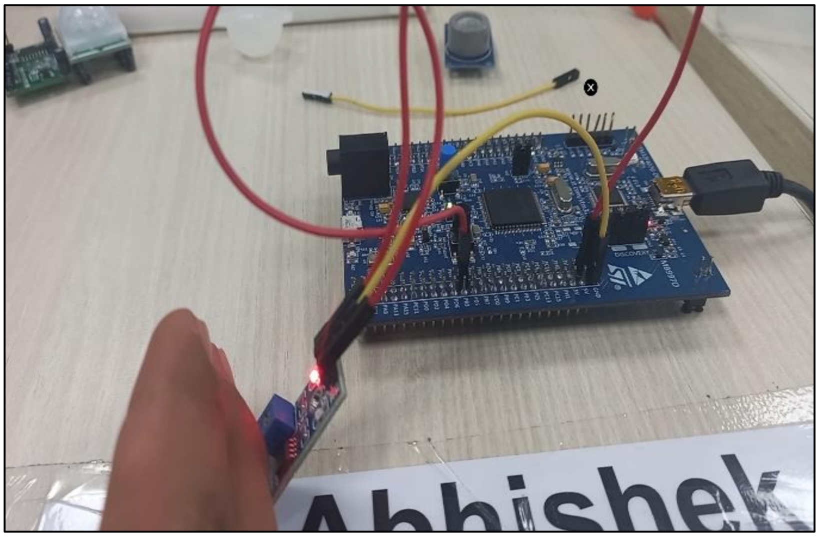


Figure 19: Obstacle Detector control module

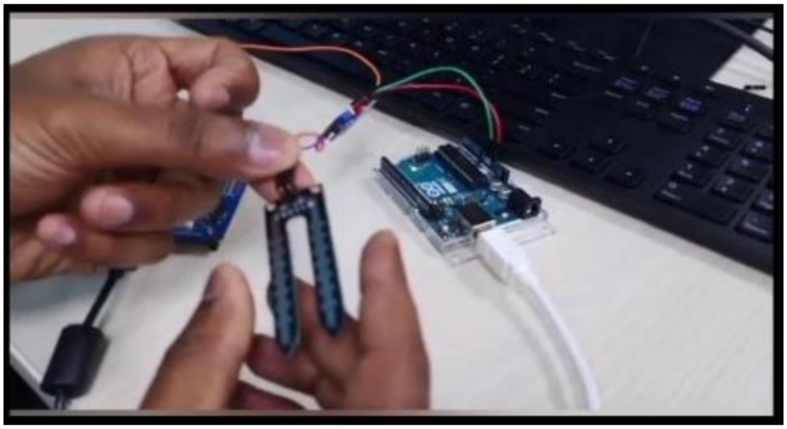


Figure 20:Wiper Control Module

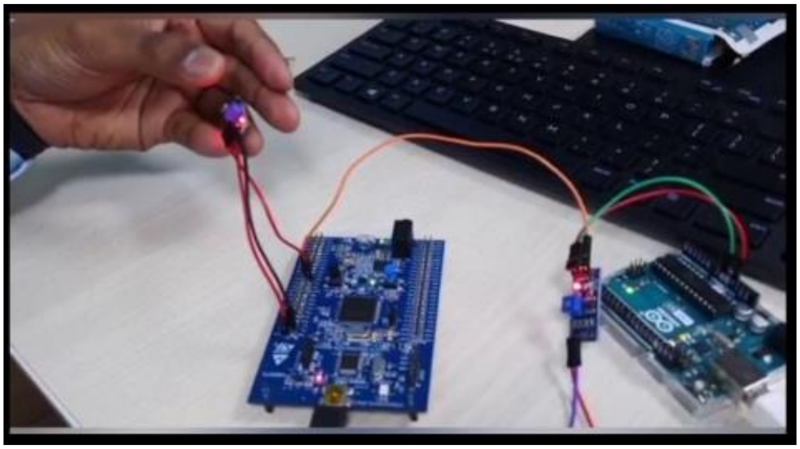


Figure 21: Headlight Control Module

Test Plan

High Level Plan

Table 16: High Level Test Plan of Hardware Project

ID	Description	Expected Input	Expected Output	Actual Output
HLP_01	Enable control Module should be enabled in order to detect different sensors.	Module is turned on	Module gets enabled	Module enabled
HLP_02	Temperature sensor should be able to detect temperature	High Temperature is provided	LED glows	LED glows indicating AC switches ON.
HLP_03	Gas Sensor should be able to detect gas leakage	Excess gas leakage	LED glows and system shut down	LED glows and car shut down
HLP_04	LDR Sensor should be able to detect higher intensity of light	Excess/low sunlight/light	Sun shed enabled or headlight switches ON	Sun shed enabled or headlight switched ON
HLP_05	Moisture sensor should be able to detect moisture	Water droplets	Moisture sensor senses water droplets and wiper starts moving	Moisture sensor senses water droplets and wiper starts moving
HLP_06	IR sensor should be able to detect obstacles	Sensor occupied by any obstacle	Red LED blinks	Red LED blinks

Low Level Plan:

Table 17: Low Level Test Plan of Hardware project

ID	Description	Expected Input	Expected Output	Actual Output
LLP_01	Thermal sensor detects the high temperature and AC switches ON.	High Temperature provided at the input of thermal sensor	AC switches ON	AC Switched ON
LLP_02	Gas sensor detects gas leakage and glows LED which prevents spontaneous car explosion.	Hand sanitizer containing alcohol	LED is switched ON and Buzzer starts beeping.	LED is ON
LLP_03	LDR sensor senses the intensity of sunlight and the shed is enabled when the sunlight is too bright	Torch Light (at small scale) or sunlight	Red LED turns ON	Red LED turns ON indicating Shed has been open
LLP_04	LDR sensor senses darkness and headlight turns ON.	Push button is pressed indicating seat shifts front or back	Green LED Glows.	Green LED Glows indicating headlight is ON
LLP_05	Moisture sensor senses water in wiper and LED switches ON	Water droplets	LED turned ON	LED turned ON indicating wiper is active
LLP_06	LED glows when some obstacle is detected	IR sensor detects obstacle	LED glows	Green LED Glows.

Implementation Summary

In this project, six functions of a Body Control Module has been implemented on a STM microcontroller using different sensors and Arduino board.

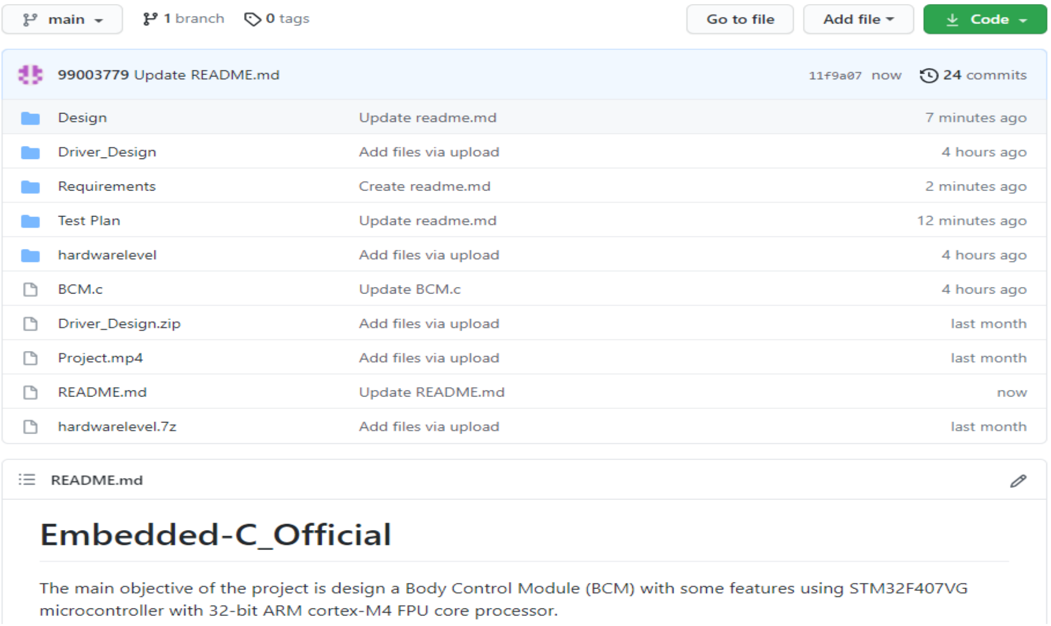
Following six features has been implemented:

1. **Air Conditioner Control Module:** In this module, temperature sensor has been used. Thermal sensor detects the high temperature and AC switches ON. Thermal sensors detect a change in a physical parameter such as resistance or output voltage that corresponds to a temperature change.
2. **Gas Leakage control Module/ Car shut down control Module:** In this module, Gas sensor has been used. Gas sensor detects gas leakage and glows LED which shuts down the car and prevents spontaneous car explosion. The gas sensor receives the attention of the liquor gas in the air and the analog voltage reading output.
3. **Sun Shed Control Module:** In this module, LDR sensor has been used. LDR sensor senses the intensity of sunlight and the shed is enabled when the sunlight is too bright. When light falls on it, the resistance changes and hence it works.
4. **Headlight Control Module:** In this module, again LDR sensor has been used. LDR sensor senses darkness and headlight turns ON.
5. **Wiper Control Module:** In this module, moisture sensor has been used. Moisture sensor senses water in wiper and LED switches ON indicating that wiper is working. Moisture sensor measures water content indirectly with the help of electricity resistance, dielectric constant, or neutron contact, as representative of moisture content.
6. **Obstacle Detector Control Module:** In this module, IR sensor has been used. LED glows when some obstacle is detected by the sensor. An IR sensor is an electronic device, which illuminates light to detect the surrounding environment. The IR sensor can detect movement.

Git Link

https://github.com/99003779/Embedded-C_Official.git

Git Dashboard



99003779 Update README.md 11f9a07 now 24 commits

Design	Update readme.md	7 minutes ago
Driver_Design	Add files via upload	4 hours ago
Requirements	Create readme.md	2 minutes ago
Test Plan	Update readme.md	12 minutes ago
hardwarelevel	Add files via upload	4 hours ago
BCM.c	Update BCM.c	4 hours ago
Driver_Design.zip	Add files via upload	last month
Project.mp4	Add files via upload	last month
README.md	Update README.md	now
hardwarelevel.7z	Add files via upload	last month

README.md

Embedded-C_Official

The main objective of the project is design a Body Control Module (BCM) with some features using STM32F407VG microcontroller with 32-bit ARM cortex-M4 FPU core processor.

Summary

In this project, we have implemented a body control module showing six features on a STM32 microcontroller board having a core processor of 32-bit ARM cortex-M4 FPU. Following six features has been implemented:

- Air Conditioner Control Module
- Gas Leakage Control Module/ Car shut down control Module
- Sun Shed Control Module
- Headlight Control Module
- Wiper Control Module
- Obstacle Detector Control Module

Individual Contribution & Highlights:

Features implemented by me in the team hardware project are:

- **Air Conditioner Control Module:** Thermal sensor detects the high temperature and AC switches ON.
- **Gas Leakage control Module/ Car shut down control Module:** The gas sensor receives the attention of the liquor gas in the air and the analog voltage reading output.

Challenges faced and how were they overcome

- Integration of all the features on a single STM32 microcontroller board.
- Hardware implementation on STM32 IDE platform.

Future Scope (If applicable):

NA

Miniproject -4 [Individual]: Embedded linux and Kernel Programming

Module/s

Modules linked to this miniproject is Ex – Linux, Embedded C, OS programming and Kernel Device Drivers.

Topic and Subtopics

1. Working with QEMU:-
   - Setting up Qemu
   - Building Custom Kernel
   - Cross Compilation Techniques
   - Booting Techniques

2. Kernel Programming and Device Drivers
   - Introduction to Kernel
   - Static and Dynamic Modules
   - Kconfig entries
   - Adding System Calls
   - Simple Drivers
   - File Operations
   - Kernel Data structures
   - Concurrency and Inter Process Communication (IPC)
   - IOCTL Operations

Objectives & Requirements

Objective:
The objective of this project is to implement the learning of the module in the following activities:

- Making a System call to echo back the given string
- Making a System Call to pass the Arguments
- Making a System call to traverse process list and print PID and PPID of a process and making a System call to retrieve the attributes of calling process
- Implementing IOCTL to echo back the string, implement two operations such that the string passed by one operation is retrieved by the other operation
- Applying mutual exclusion between first two threads by implementing three kernel threads. First two threads will write N nodes into a kernel list. The third thread waits for the first two threads and traverses the list while mutual exclusion is applied between the first two threads. Exit method cleanup the list nodes

Requirements:

High Level Requirement:

Table 18: HLR of Kernel Activity

ID	Requirements	Description	Status
HLR_01	System Call to echo back string	The system call echoes back the given string on target from user space	Implemented
HLR_02	System Call to pass the argument	The System Call passes the argument from user space to Kernel Space	Implemented
HLR_03	System Call to traverse process list and print PID and PPID and retrieve attributes of the calling process	The System Call traverse the process list, prints PID and PPID and retrieve attributes of the calling process on the target window	Implemented
HLR_04	IOCTL to echo back string using two operations	Use IOCTL to echo back string by implementing 2 operations, one for passing string and other of for retrieving it.	Implemented
HLR_05	Threads implementation of 3 threads where 3rd third retrieves data for first 2 threads with list of N nodes and its traversal	Applying mutual exclusion to implement 3 threads where 2 write N nodes to threads and 3rd traverses the list. Exit module cleans up the node.	Implemented

Low Level Requirement:

Table 19: LLR of Kernel Activity

ID	Requirements	Description	Status
LLR_01	The string is echoed back to Target or the kernel space by system call	The string "abcdef" gets echoed to Qemu by creating a system call invoked by the user space.	Implemented
LLR_02	The Argument is passed back to the target	The Argument is passed back to the kernel space from user space	Implemented
LLR_03	The System Call traverse the process list, prints PID and PPID and retrieve attributes of the calling process on the target window	User invokes a system call to get PID, PPID and attributes of the calling process	Implemented
LLR_04	IOCTL used to echo back string with 2 operations	IOCTL to echo back string using 2 operations one to pass the argument and other to retrieve data and display it on Qemu.	Implemented
LLR_05	Mutual exclusion of three threads to traverse, list and clean up nodes	First two threads will write N nodes into a kernel list. The third thread waits for it and traverses the list. Mutual exclusion applied between the first two threads. Exit method cleanup the list nodes	Implemented

Test Plan

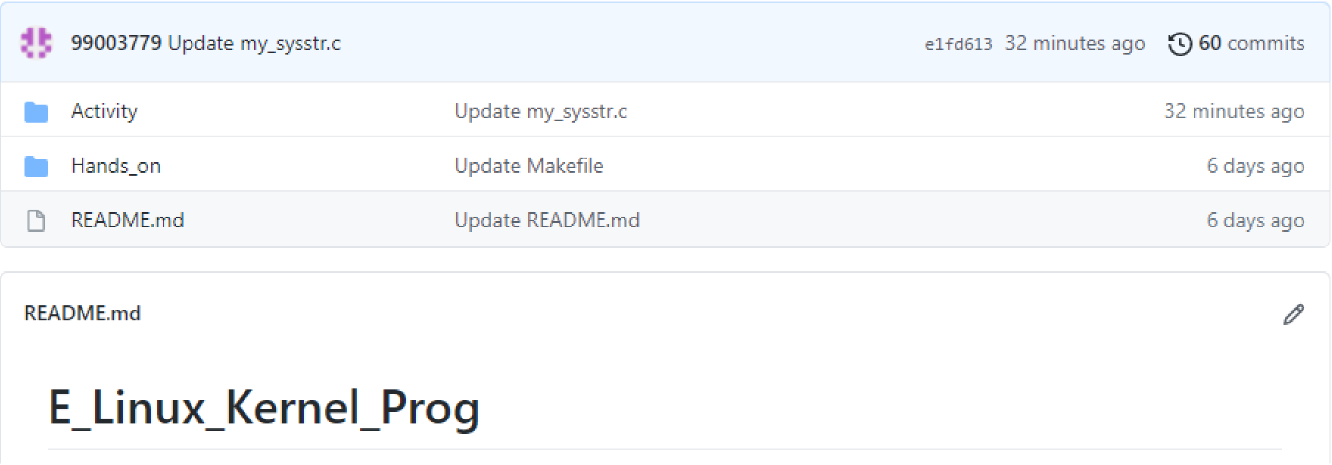
Table 20: Test Plan of Kernel Activity Implemented

Test ID	Description	Expected Input	Expected Output	Actual Output
TP_01	The string passed by the user is echoed to target space when invoked	"abcdef" as string passed in userspace	"abcdef" is printed in target space	"abcdef" is printed in target space using command dmesg
TP_02	The argument passed by user is printed to the kernel space	1,2,3 as argument is passed by user	1,2,3 is passed to the kernel space	1,2,3 is printed in kernel space using command dmesg in kernel space only
TP_03	The PID, PPID and the attributes of the process is retrieved in target space when user space code is invoked	User space invocation for the system call	PID, PPID, Process State, Priority, RT Priority, Static Priority and Normal priority are printed in kernel space.	PID, PPID, process State, Priority, RT Priority, Static Priority and Normal priority are printed in kernel space.
TP_04	3 threads are implemented. 1st and 2nd thread creates a list of N nodes, 3rd traverses the list after waiting for 2 threads to complete.	Invocation of the code	3rd thread waits for 1st 2 threads to create a list through N nodes. 3rd thread then traverses the list and adds in Kfifo.	3rd thread waits for 1st 2 threads to create a list through N nodes. 3rd thread then traverses the list and adds in Kfifo.
TP_05	3 threads are implemented. 1st and 2nd thread creates a list of N nodes, 3rd traverses the list after waiting for 2 threads to complete by mutual exclusion.	Invocation of the code	3rd thread waits for 1st 2 threads to create a list through N nodes by mutual exclusion. 3rd thread then traverses the list. Exit method cleans up all the nodes.	3rd thread waits for 1st 2 threads to create a list through N nodes by mutual exclusion. 3rd thread then traverses the list. Exit method cleans up all the nodes.

Git Link

https://github.com/99003779/Embedded_linux_and_kernel_programming.git

Git Dashboard



99003779 Update my_sysstr.c e1fd613 32 minutes ago 60 commits

Activity	Update my_sysstr.c	32 minutes ago
Hands_on	Update Makefile	6 days ago
README.md	Update README.md	6 days ago

README.md

E_Linux_Kernel_Prog

Individual Contribution & Highlights

This whole project is an individual project. So, all the activities performed in this project was done by me only.

Summary

The whole project is based on kernel programming with understanding of embedded linux, device drivers, concept of system call and IOCTL i.e input output control.

The Linux Kernel Driver model is a combination of all the different types of drivers that were previously used in the kernel. It aims to add bus-specific drivers to bridges and devices by integrating data collection and operating in data that are globally accessible.

System calling is a system configuration program that asks for applications from the kernel, and strace is a powerful tool that allows you to follow a small layer between user processes and Linux kernel.

Ioctl, which means "input output control" is a type of system call to a specific device. There are only a few system calls on Linux (300-400), which are not enough to display all the functions of unique devices. The driver can then specify ioctl which allows the user app to send orders.

Individual Contribution & Highlights:

The whole project was the individual project.

Challenges faced and how were they overcome

- System call implementation.
- Depth understanding of user space and kernel space

Future Scope (If applicable):-

NA

Miniproject -5 [Individual]: Networking

Module/s

Modules linked to this miniproject is Data Communication and Networking

Topic and Subtopics

IPv4 and IPv6 addresses
Routers
Routing Protocols
Networking
Packet Tracer Tool

Objectives & Requirements

Objective:
The objective of this project is to design a network consisting of various servers connecting two or more sub networks and is able to send data packets throughout the network. Cisco Packet tracer is the tool that has been used in network designing
Requirements:

Cisco Packet Tracer: It is a strong network simulation program or the software that allows many esearchers, scholars and students across the globe to experiment with the behavior of network. It supplements physical equipments so that students will be able to create networks immense number of devices. Following is a small network demonstrating many networks interconnected by routers and switch.

Each server should be able to send packet to every other server. Whether the server is in the same network or different network.

Design

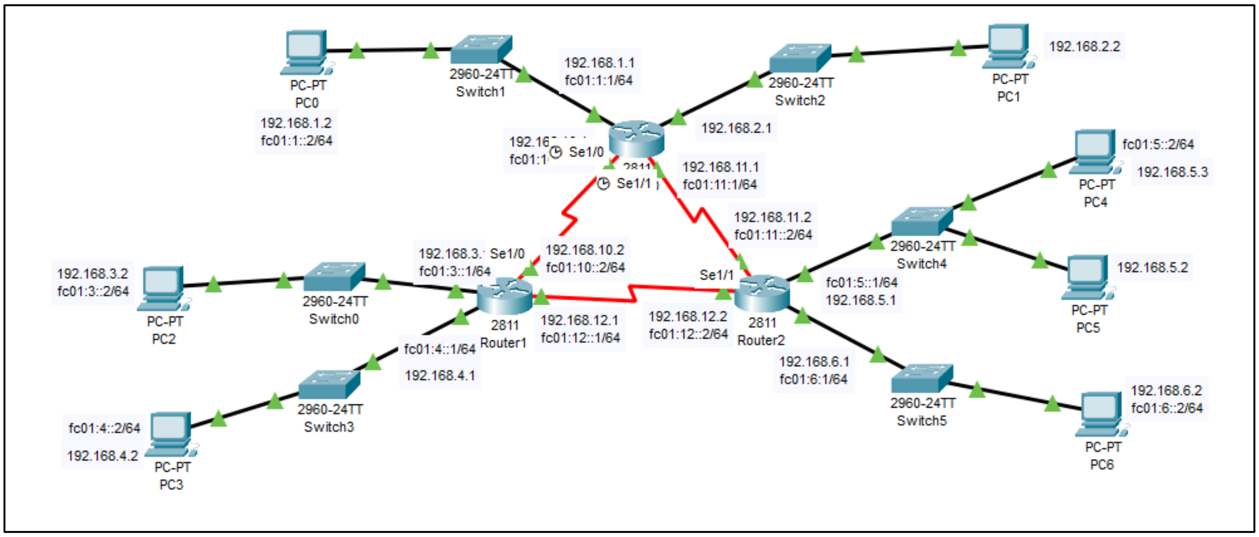
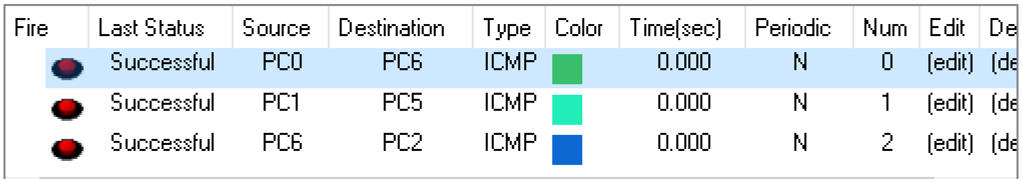


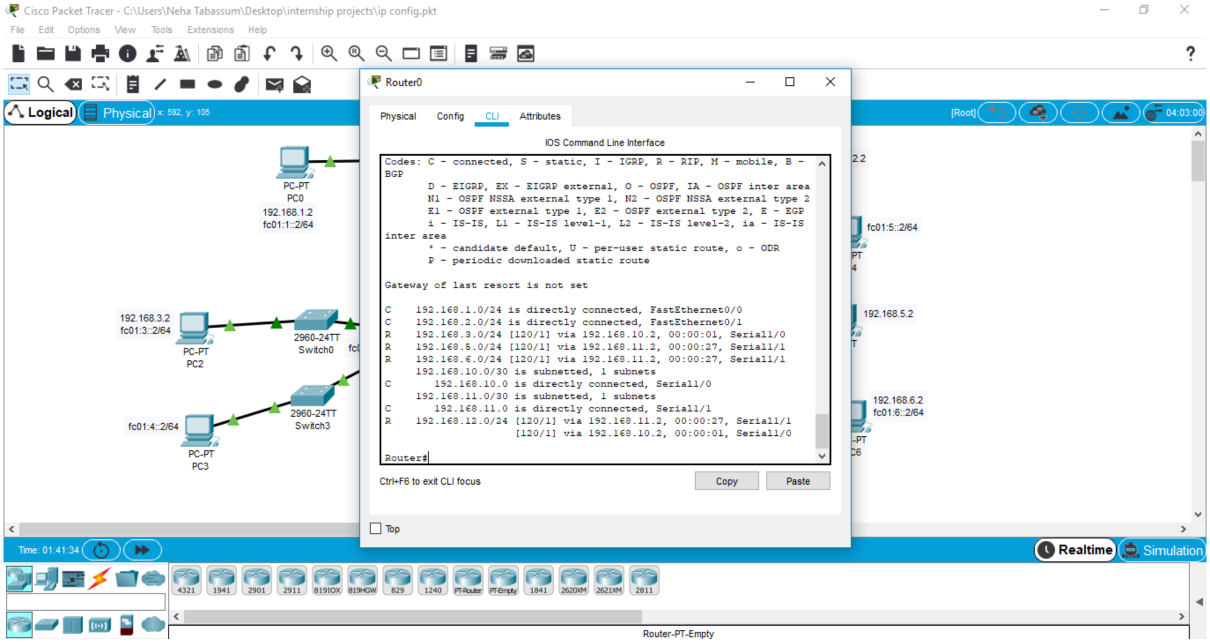
Figure 22: Serial Connection of routers for packet transmission from one server to the other



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	De
●	Successful	PC0	PC6	ICMP	🟩	0.000	N	0	(edit)	(de
●	Successful	PC1	PC5	ICMP	🟩	0.000	N	1	(edit)	(de
●	Successful	PC6	PC2	ICMP	🟦	0.000	N	2	(edit)	(de

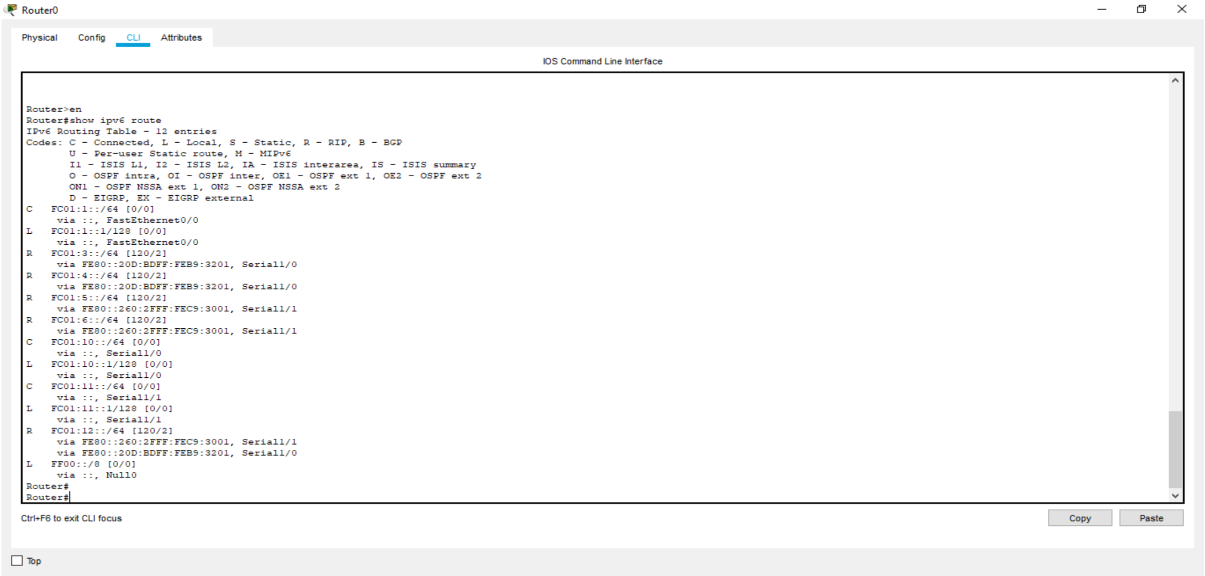
Figure 23:Transfer of Packets from different sources to different destination

Test Plans:



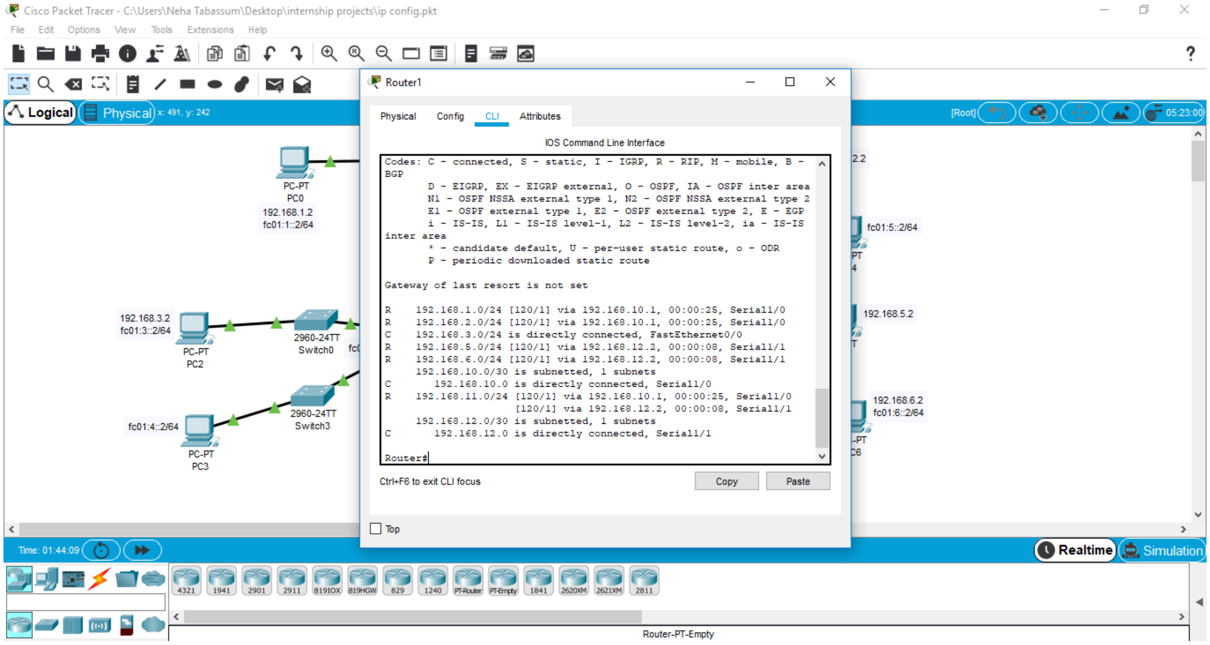
```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -
BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.1.0/24 is directly connected, FastEthernet0/0
C    192.168.2.0/24 is directly connected, FastEthernet0/1
R    192.168.3.0/24 [120/1] via 192.168.10.2, 00:00:01, Serial1/0
R    192.168.5.0/24 [120/1] via 192.168.11.2, 00:00:27, Serial1/1
R    192.168.6.0/24 [120/1] via 192.168.11.2, 00:00:27, Serial1/1
     192.168.10.0/30 is subnetted, 1 subnets
C       192.168.10.0 is directly connected, Serial1/0
     192.168.11.0/30 is subnetted, 1 subnets
C       192.168.11.0 is directly connected, Serial1/1
R    192.168.12.0/24 [120/1] via 192.168.11.2, 00:00:27, Serial1/1
                     [120/1] via 192.168.10.2, 00:00:01, Serial1/0

Router#
```

Figure 24: IPv4 routes on router 0 (show ip route)



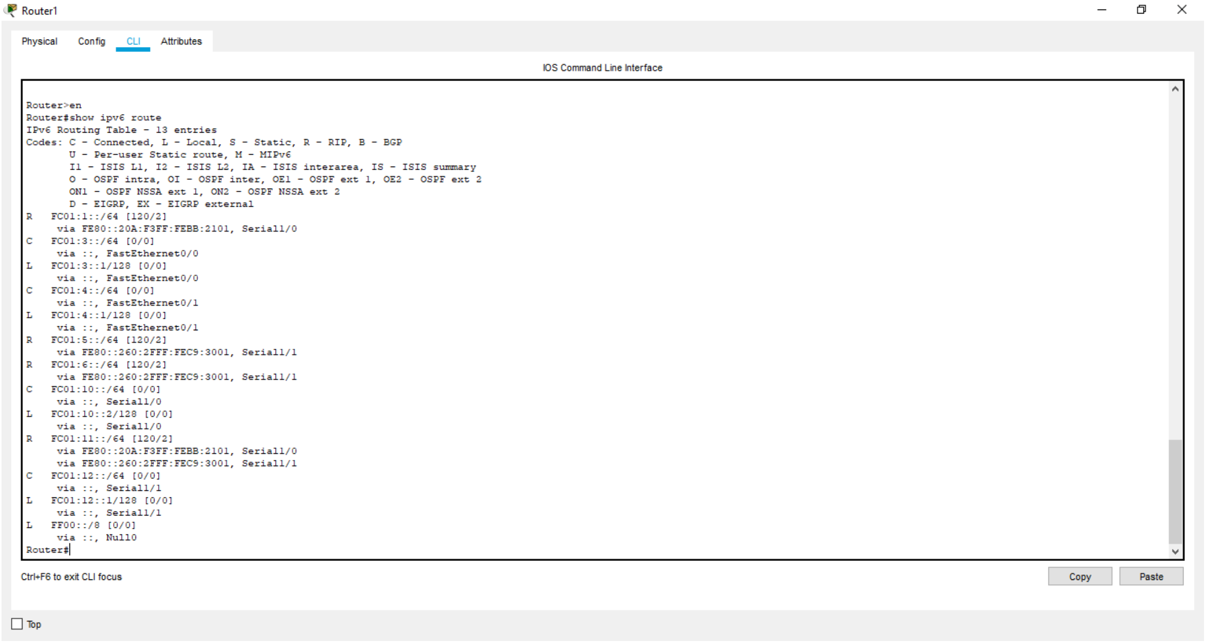
```
Router>en
Router#show ipv6 route
IPv6 Routing Table - 12 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route, M - MIPv6
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
C   FC01:1::/64 [0/0]
     via ::, FastEthernet0/0
L   FC01:1::1/128 [0/0]
     via ::, FastEthernet0/0
R   FC01:3::/64 [120/2]
     via FE80::20D:BDFF:FEB9:3201, Serial1/0
R   FC01:4::/64 [120/2]
     via FE80::20D:BDFF:FEB9:3201, Serial1/0
R   FC01:5::/64 [120/2]
     via FE80::260:2FFF:FEC9:3001, Serial1/1
R   FC01:6::/64 [120/2]
     via FE80::260:2FFF:FEC9:3001, Serial1/1
C   FC01:10::/64 [0/0]
     via ::, Serial1/0
L   FC01:10::1/128 [0/0]
     via ::, Serial1/0
C   FC01:11::/64 [0/0]
     via ::, Serial1/1
L   FC01:11::1/128 [0/0]
     via ::, Serial1/1
R   FC01:12::/64 [120/2]
     via FE80::260:2FFF:FEC9:3001, Serial1/1
     via FE80::20D:BDFF:FEB9:3201, Serial1/0
L   FF00::/8 [0/0]
     via ::, Null0
Router#
Router#
```

Figure 25: IPv6 routes on router 0 (show ipv6 route)



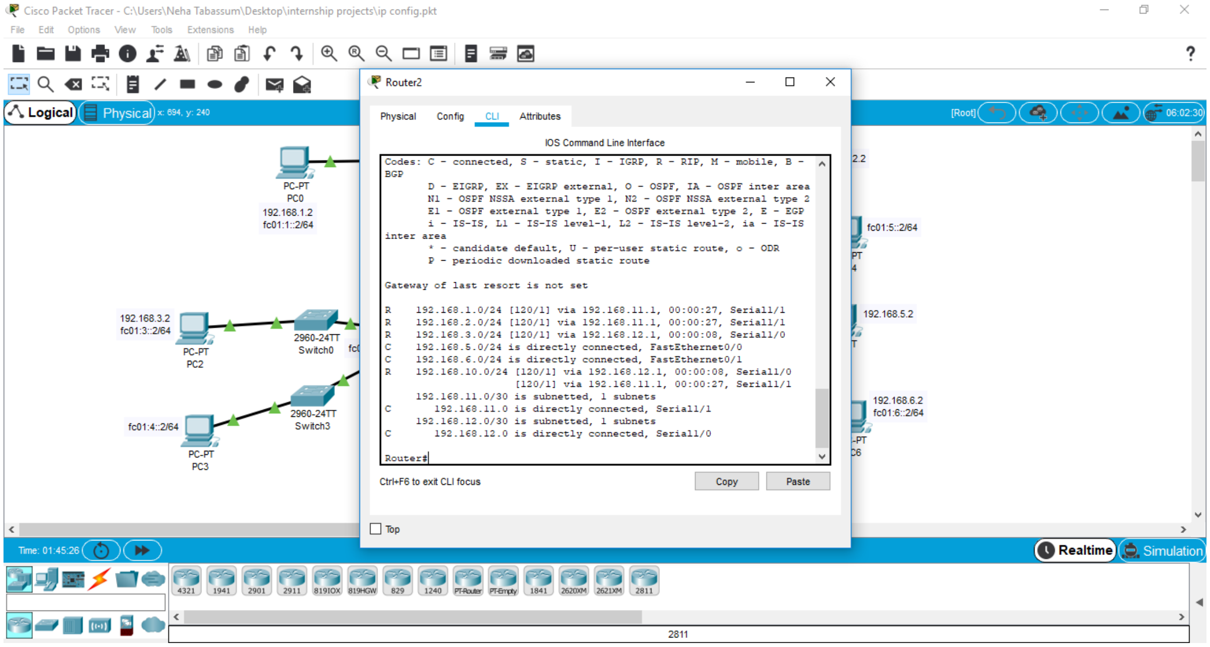
```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -
BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

R    192.168.1.0/24 [120/1] via 192.168.10.1, 00:00:25, Serial1/0
R    192.168.2.0/24 [120/1] via 192.168.10.1, 00:00:25, Serial1/0
C    192.168.3.0/24 is directly connected, FastEthernet0/0
R    192.168.5.0/24 [120/1] via 192.168.12.2, 00:00:08, Serial1/1
R    192.168.6.0/24 [120/1] via 192.168.12.2, 00:00:08, Serial1/1
     192.168.10.0/30 is subnetted, 1 subnets
C       192.168.10.0 is directly connected, Serial1/0
R    192.168.11.0/24 [120/1] via 192.168.10.1, 00:00:25, Serial1/0
                     [120/1] via 192.168.12.2, 00:00:08, Serial1/1
     192.168.12.0/30 is subnetted, 1 subnets
C       192.168.12.0 is directly connected, Serial1/1

Router#
```

Figure 26: IPv4 routes on router 1 (show ip route)



```
Router>en
Router#show ipv6 route
IPv6 Routing Table - 13 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route, M - MIPv6
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
R   FC01:1::/64 [120/2]
     via FE80::20A:F3FF:FEBB:2101, Serial1/0
C   FC01:3::/64 [0/0]
     via ::, FastEthernet0/0
L   FC01:3::1/128 [0/0]
     via ::, FastEthernet0/0
C   FC01:4::/64 [0/0]
     via ::, FastEthernet0/1
L   FC01:4::1/128 [0/0]
     via ::, FastEthernet0/1
R   FC01:5::/64 [120/2]
     via FE80::260:2FFF:FEC9:3001, Serial1/1
R   FC01:6::/64 [120/2]
     via FE80::260:2FFF:FEC9:3001, Serial1/1
C   FC01:10::/64 [0/0]
     via ::, Serial1/0
L   FC01:10::2/128 [0/0]
     via ::, Serial1/0
R   FC01:11::/64 [120/2]
     via FE80::20A:F3FF:FEBB:2101, Serial1/0
     via FE80::260:2FFF:FEC9:3001, Serial1/1
C   FC01:12::/64 [0/0]
     via ::, Serial1/1
L   FC01:12::1/128 [0/0]
     via ::, Serial1/1
L   FF00::/8 [0/0]
     via ::, Null0
Router#
```

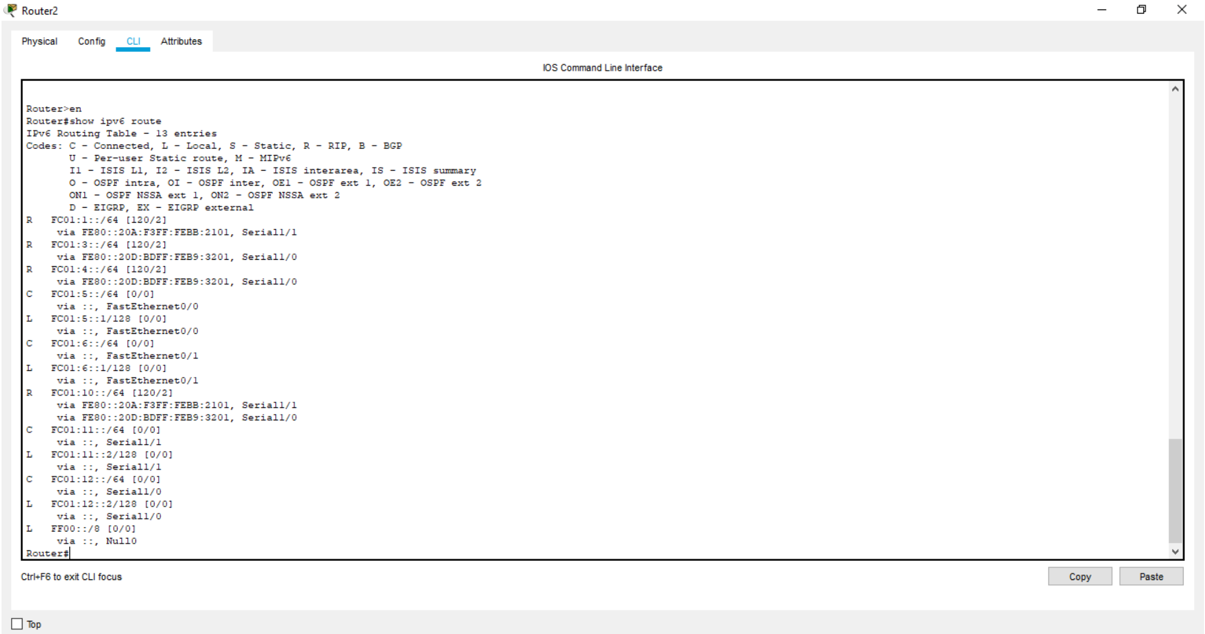
Figure 27: IPv6 routes on router 1 (show ipv6 route)



```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B -
BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

R    192.168.1.0/24 [120/1] via 192.168.11.1, 00:00:27, Serial1/1
R    192.168.2.0/24 [120/1] via 192.168.11.1, 00:00:27, Serial1/1
R    192.168.3.0/24 [120/1] via 192.168.12.1, 00:00:08, Serial1/0
C    192.168.5.0/24 is directly connected, FastEthernet0/0
C    192.168.6.0/24 is directly connected, FastEthernet0/1
R    192.168.10.0/24 [120/1] via 192.168.12.1, 00:00:08, Serial1/0
                     [120/1] via 192.168.11.1, 00:00:27, Serial1/1
     192.168.11.0/30 is subnetted, 1 subnets
C       192.168.11.0 is directly connected, Serial1/1
     192.168.12.0/30 is subnetted, 1 subnets
C       192.168.12.0 is directly connected, Serial1/0

Router#
```

Ctrl+F6 to exit CLI focus

Figure 28: IPv4 routes on router 2 (show ip route)



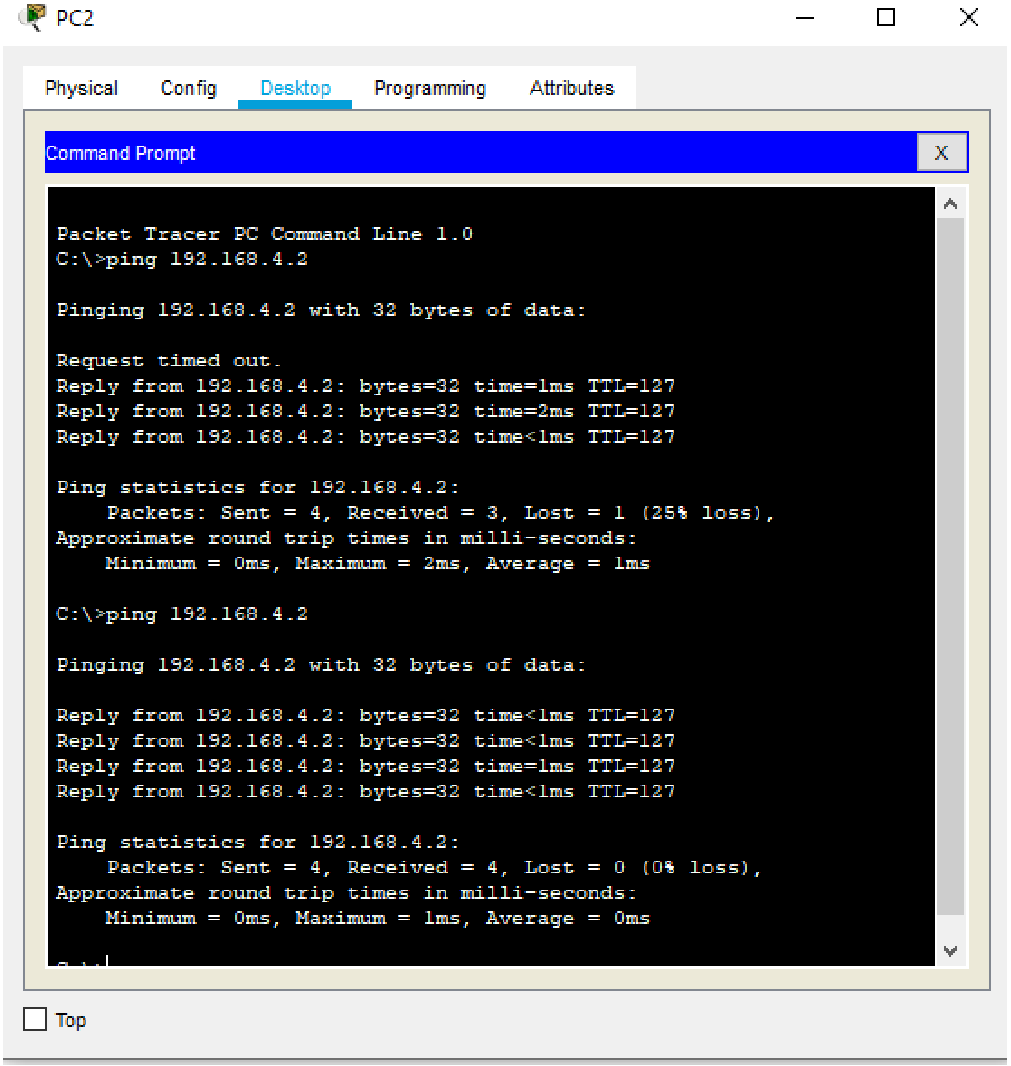
```
Router>en
Router#show ipv6 route
IPv6 Routing Table - 13 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route, M - MIPv6
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
R   FC01:1::/64 [120/2]
     via FE80::20A:F3FF:FEBB:2101, Serial1/1
R   FC01:3::/64 [120/2]
     via FE80::20D:BDFF:FEB9:3201, Serial1/0
R   FC01:4::/64 [120/2]
     via FE80::20D:BDFF:FEB9:3201, Serial1/0
C   FC01:5::/64 [0/0]
     via ::, FastEthernet0/0
L   FC01:5::1/128 [0/0]
     via ::, FastEthernet0/0
C   FC01:6::/64 [0/0]
     via ::, FastEthernet0/1
L   FC01:6::1/128 [0/0]
     via ::, FastEthernet0/1
R   FC01:10::/64 [120/2]
     via FE80::20A:F3FF:FEBB:2101, Serial1/1
     via FE80::20D:BDFF:FEB9:3201, Serial1/0
C   FC01:11::/64 [0/0]
     via ::, Serial1/1
L   FC01:11::2/128 [0/0]
     via ::, Serial1/1
C   FC01:12::/64 [0/0]
     via ::, Serial1/0
L   FC01:12::2/128 [0/0]
     via ::, Serial1/0
L   FF00::/8 [0/0]
     via ::, Null0
Router#
```

Ctrl+F6 to exit CLI focus

Figure 29: IPv6 routes on router 2 (show ipv6 route)

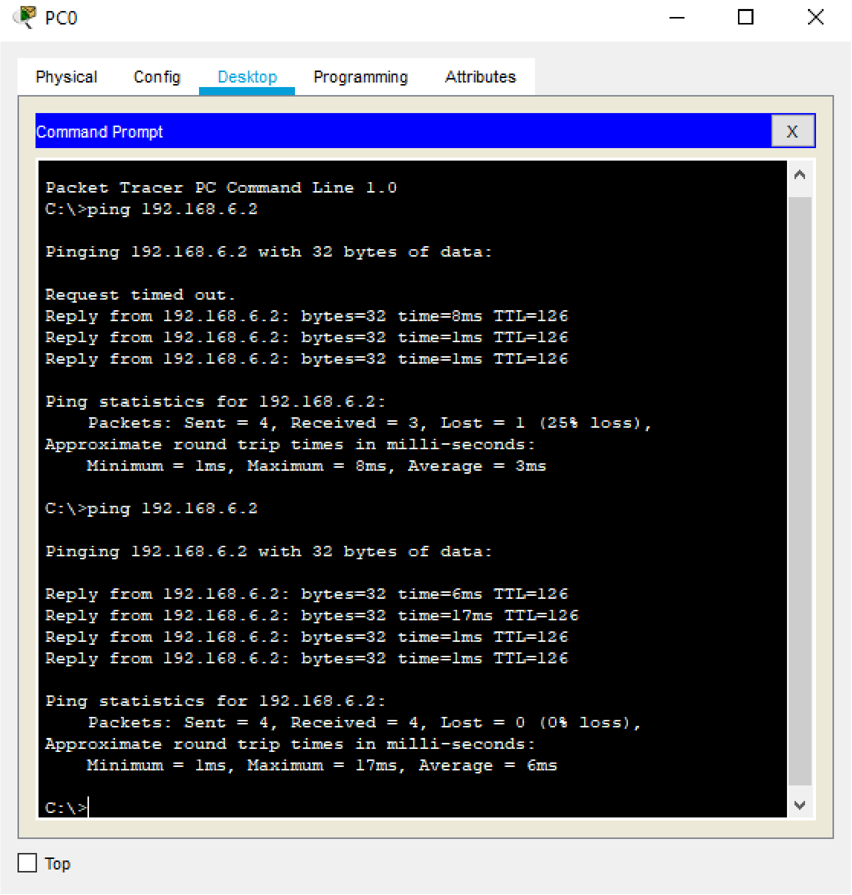
Testing Results:

To test whether a network is working properly or not, a ping command is used. Ping is a tool or a network utility program which allows you to check weather a particular host is reachable or not. Loop back address is another IP address which used to check working of the self server. Example of loop back ip address is 127.0.0.1 and it will always return a reply unless a network security system prevents it. Eg firewall.



```
Packet Tracer PC Command Line 1.0
C:\>ping 192.168.4.2

Pinging 192.168.4.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.4.2: bytes=32 time=1ms TTL=127
Reply from 192.168.4.2: bytes=32 time=2ms TTL=127
Reply from 192.168.4.2: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.4.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 1ms

C:\>ping 192.168.4.2

Pinging 192.168.4.2 with 32 bytes of data:

Reply from 192.168.4.2: bytes=32 time<1ms TTL=127
Reply from 192.168.4.2: bytes=32 time<1ms TTL=127
Reply from 192.168.4.2: bytes=32 time=1ms TTL=127
Reply from 192.168.4.2: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.4.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

Figure 30: Pinging from IPv4 to IPv4 in the same network



```
Packet Tracer PC Command Line 1.0
C:\>ping 192.168.6.2

Pinging 192.168.6.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.6.2: bytes=32 time=8ms TTL=126
Reply from 192.168.6.2: bytes=32 time=1ms TTL=126
Reply from 192.168.6.2: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.6.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 8ms, Average = 3ms

C:\>ping 192.168.6.2

Pinging 192.168.6.2 with 32 bytes of data:

Reply from 192.168.6.2: bytes=32 time=6ms TTL=126
Reply from 192.168.6.2: bytes=32 time=17ms TTL=126
Reply from 192.168.6.2: bytes=32 time=1ms TTL=126
Reply from 192.168.6.2: bytes=32 time=1ms TTL=126

Ping statistics for 192.168.6.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 17ms, Average = 6ms

C:\>
```

Figure 31: Pinging from IPv4 to IPv4 in different network

Implementation Summary

The implemented design consist of three networks interconnected by router which follows RIP routing protocol. Each Data packets from every server is able to send data to every other server.

Individual Contribution & Highlights:

The whole project was the individual project.

Challenges faced and how were they overcome

- Interconnection of different networks using routers but using routing protocols it was interconnected
- Implementation of routing protocols but by learning and reading different routing protocols it was implemented
- Sending of data packets from one network to another but with proper integration data packets were able to send to different networks

Future Scope (If applicable):

NA