./

Learning Report – Genesis SDLC

Course Code: <CODE>



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

**Document History**

# 

Contents

Activity and Tasks 3

Activity 1– System/Software DevelopmenT 5

1.1– WHAT IS MEMORY FOAM? 5

1.2 –Ageing 6

1.3 – Costing 7

1.4– Problem Statement 8

1.5–Swot Analysis 8

1.6– Requirements 9

1.7–Test Plan 9

1.8– UML 10

Activity 2 –CI Workflow for C Programming 12

2.1–GIT Summary 12

Activity 3 – Agile Aspects 14

3.1– V model 14

3.2– Agile model 16

Activity 4 – 17

1.2 –Electronic Calculator 17

1.2 –Ageing 18

1.3 – Costing 19

1.4– Problem Statement 20

1.5–Swot Analysis 20

1.6– Requirements 20

1.7–Test Plan 21

1.8–UML 22

.5–GIT 25

LINK FOR GIT–

1.2 –ACTIVITY 2 14

1.2 –ACTIVITY 4 27

Referance 29

list of tables 14

Table 1:Memory Foam Requirements 14

Table 2:Memory FOam Requirements with testing 15

Table 1: Electronic Calculator Requirements 18

Table 2:Electronic Calculator Requirements with testing 119

List of figures 6

Fig 1: durability 6

Fig 2: ageing and costing 7

fig 3: activity uml 10

fig 4: State UML 10

fig 5: Class UML 11

Fig 6: Object UML 11

Fig 7: Evolution 18

Fig 8: ageing and costing 19

fig 9: activity uml 22

fig 10: State UML 23

fig 11: Class uml 24

#### ACTIVITY:1

What is memory foam?

It is a polyurethane material that is sensitive to pressure and temperature, used especially in mattresses, where it moulds to the shape of an individual's body. It is also known as Visco elastic foams and was invented by Charles Yost in 1976. It was first used by NASA in the seats for astronomers in the spaceship. But since 1990’s it has made its way in furnitures as well as shoes. But more popularly in mattresses.[1]

There are popularly three types of memory foam:

1. **Traditional:** This type first started to take hold of the market in the mid 1990’s and for good reason. Unlike a spring mattress, traditional memory foam has been designed so your body is cradled while you sleep. What this does is helps to increase blood circulation and reduces the amount of stress you get on various joints across the body while you sleep.
2. **Air Cool Memory Foam:** One of the biggest complaints about memory foam mattresses is the amount of heat they can produce. This is because lots of models have been designed so they react to your body temperature. When this happens, the material softens which is why you get such great support from them. In fact, you could say it’s a little like sleeping on air.
3. **Gel memory foam:** Just like traditional memory foam, gel technology will help stop motion transfer when people sleep in the same bed and will cradle the pressure points over your body. Gel foam is also anti-microbial so you can keep those allergies at bay.[2]

Ageing

The Gel memory foam is more durable than the traditional memory foam. Some companies claim to provide a durability of 25 years. But with time it loses its ability to retain back its shape, and might even start to sag. The memory foam have a tendancy to have an increase in its density and tend to become heavier. Hence the actual durability of the memory foam ranges from 4 years to 12 years .

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Type of memory foam** | **Durability** |
| 1 | Traditional | 3-4 years |
| 2 | Air Cool Memory Foam | 3-5 years |
| 3 | Gel memory Foam | 8-12 years |

**Fig 1:Durability**

Costing

The cost varies from 6000/- to 20000/- depending upon the feature of the memory foam, Gel memory foam being the highest in price and traditional being the lowest.

**Fig 2: Ageing vs Costing**

Problem Statement

The memory foam has a disadvantage that it is not waterproof, even too much humidity in the surrounding can make it dense as it would absorb moisture so on adding a humidity sensor can give alerts in order to have a longer durability of the memory foam.

SWOT analysis

1. Strength: The intimidation to protect the health of the memory foam is possible.
2. Weakness: Using a sensor will require of provision of supply. The number of sensors will also increase with the increase in the surface area of the foam for better results.
3. Opportunity: This will make the memory foam smarter.
4. Threat: Glitch in the power supply or any peaks

Requirements

**High Level Requirements:**

1. Customer shall be intimidated when the surrounding’s humidity can affect the memory foam.
2. The alert should come as a notification on the phone.
3. Recommendations to protect the memory

**Low Level requirement**:

1. Should be cost effective.
2. The system shall have not many sensors.

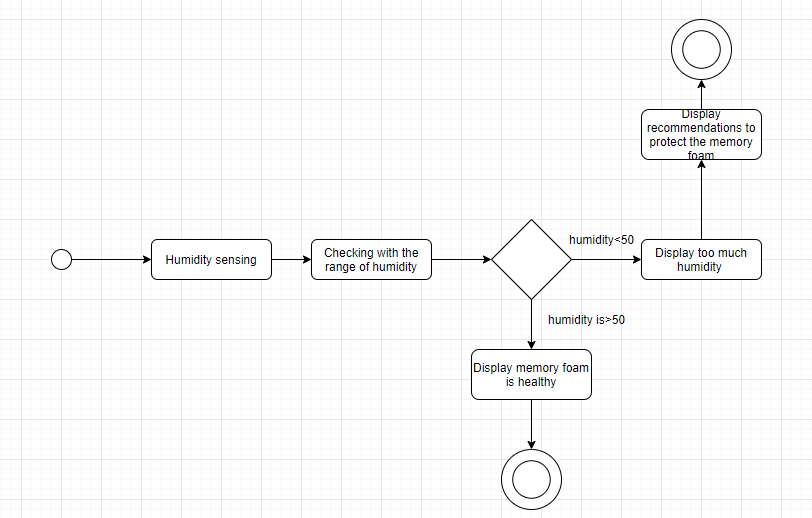
Test Plan

1.**Requirement based Test Plan**: If the relative humidity is more than 50% then it should intimidate by giving a notification on a smart phone which has access to it.[3][4]

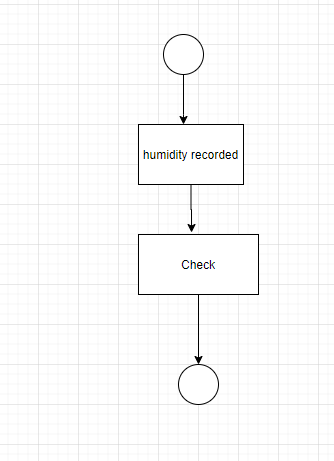
2.**Scenario based**: Verification of the reading of the sensor with the relative humidity is done and if it is not verified, then a notification is generated that “malfunction detected”.

3.**Boundary based Test Plan**: If the humidity is 50% then notification is given that “maximum humidity is reached”.

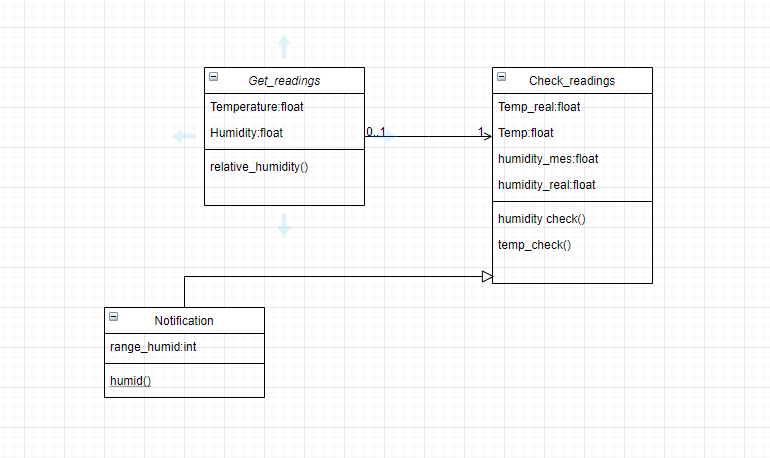
UML



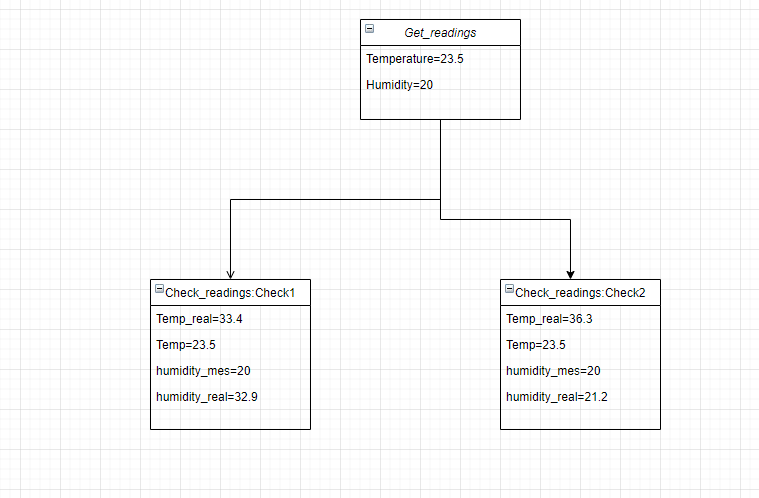
**Fig 3: Activity UML**



**Fig 4:State UML**



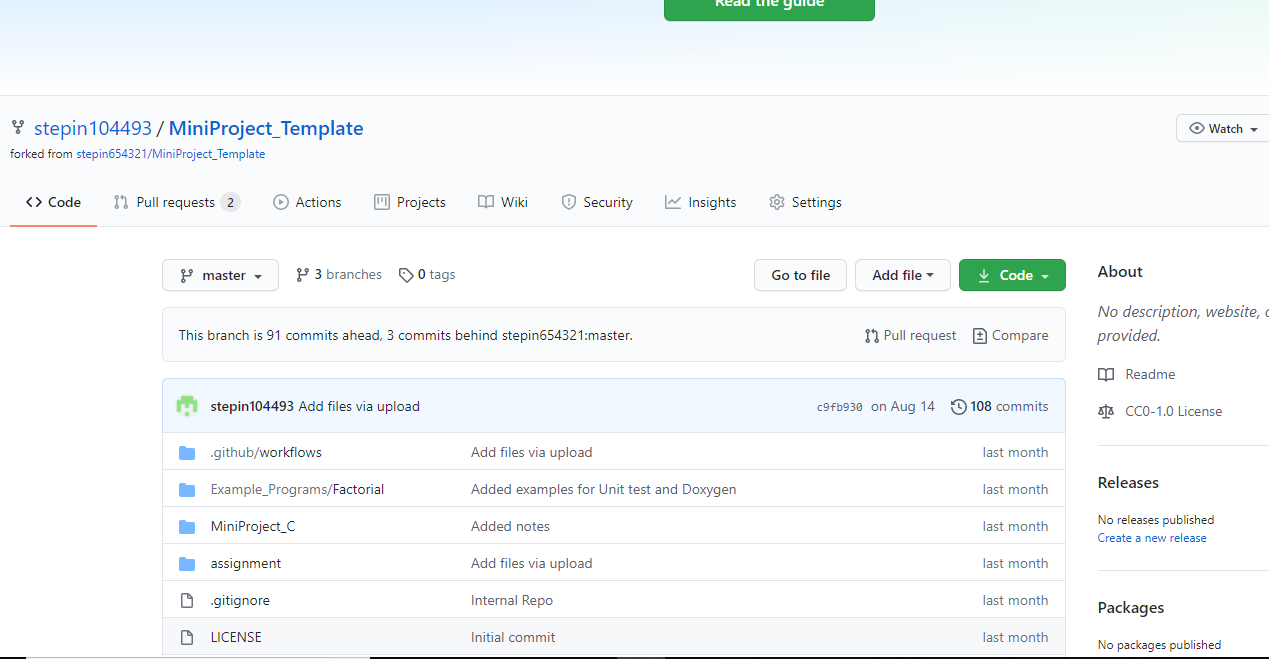
**Fig 5:Class UML**

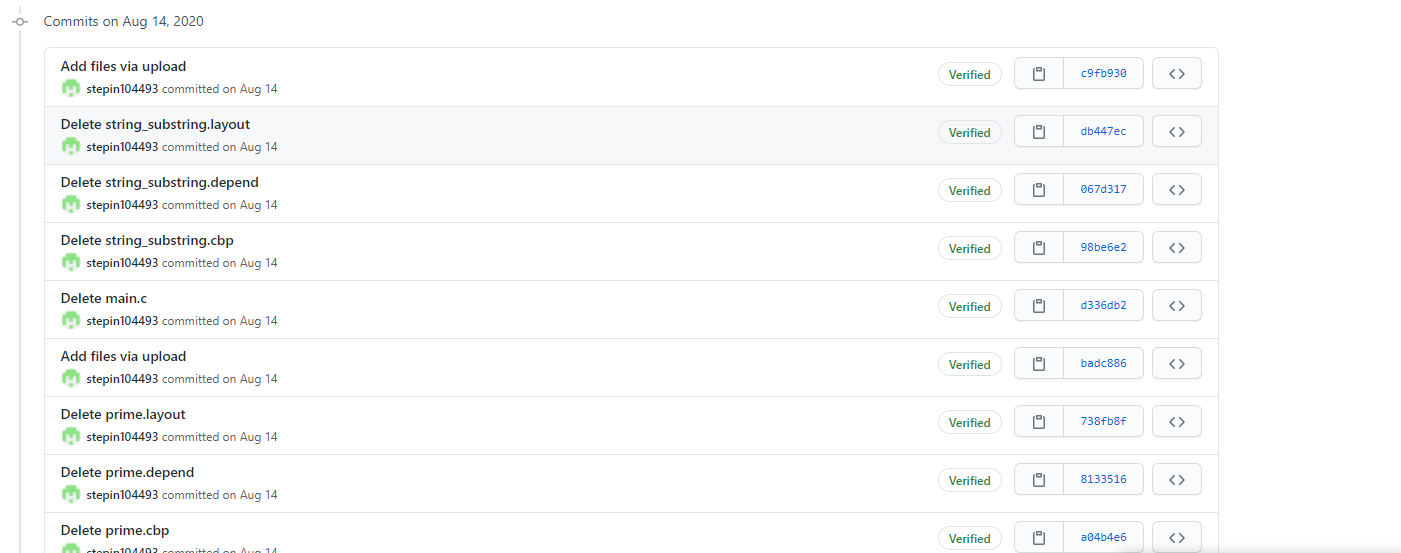


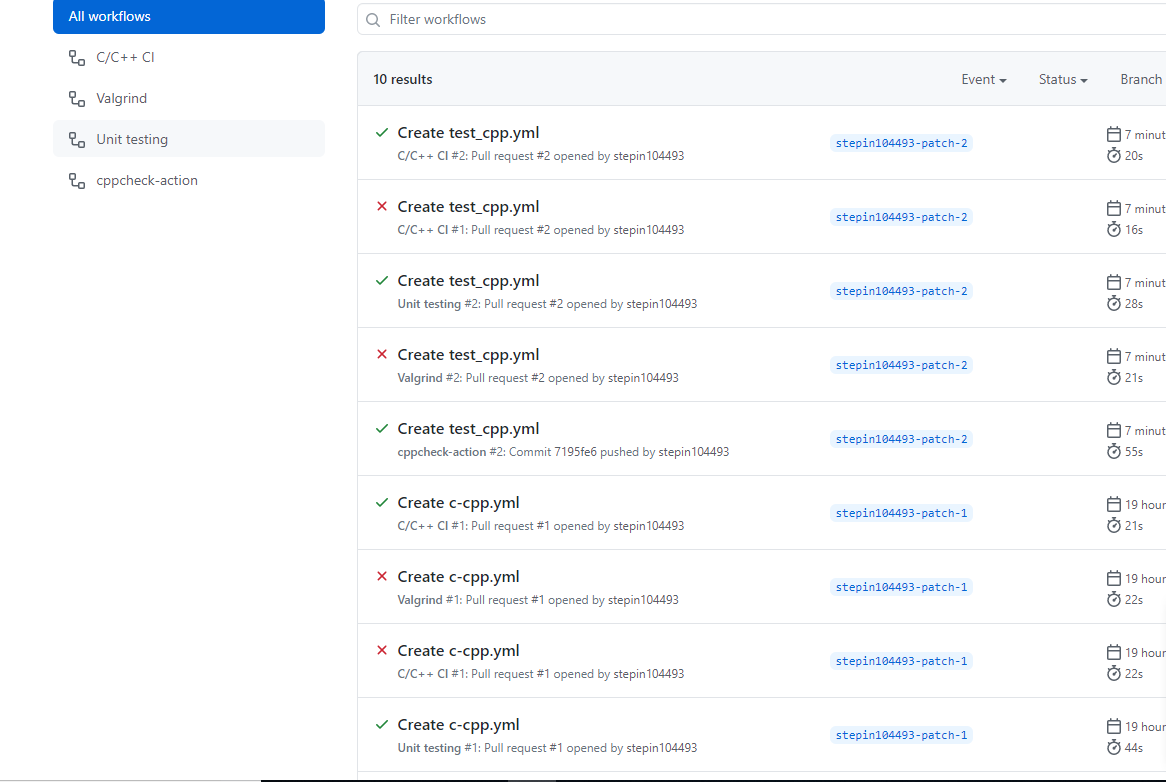
**Fig 6: Object UML**

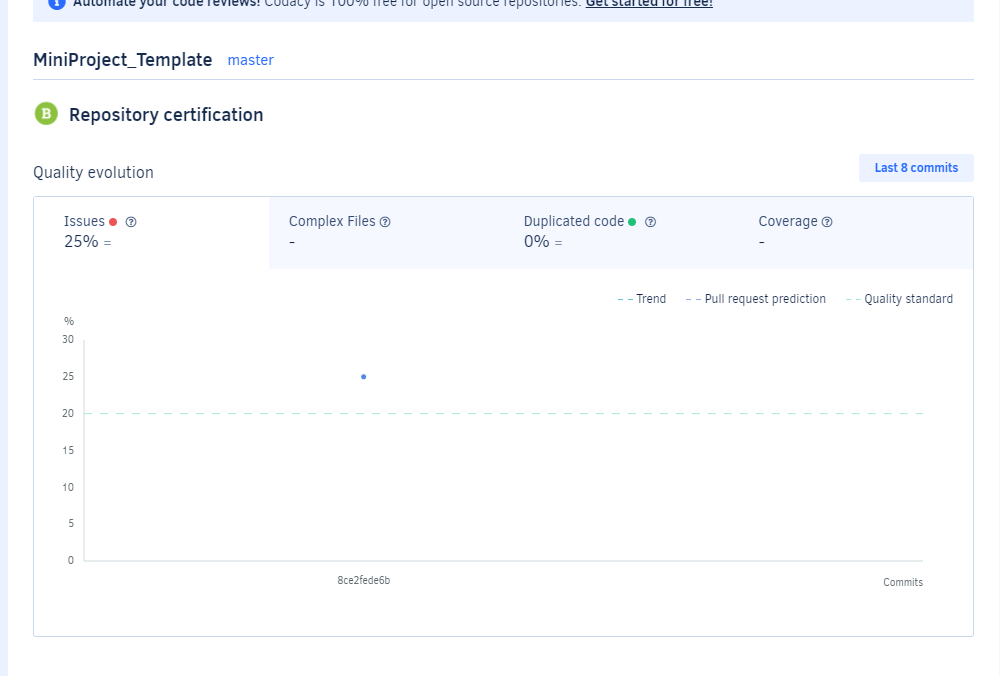
#### ACTIVITY 2:

GIT Summary









Link for GIT:

<https://github.com/stepin104493/MiniProject_Template>

#### ACTIVITY:3

V MODEL

**High level testing:**

1. If the humidity is greater than 50% then give an alert to the smart phone .
2. If humidity is equal to 50% then give a notification as “ maximum limit reached” in the smart phone.

**Low level testing:**

1. The sensor should be able to detect the humidity as precisely as possible.
2. The ardiuno should be connected properly with the sensor

|  |  |
| --- | --- |
| ID | DESCRIPTION |
| 1 | Customer shall be intimidated when the surrounding’s humidity can affect the memory foam. |
| 2 | The alert should come as a notification on the phone. |

**Table 1: Requirement**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Description | Preconditions | Expected  Input | Expected Output | Actual Output |
| 1 | Customer shall be intimidated when the surrounding’s humidity can affect the memory foam. | Continuous power supply, setting of the sensor,WIFI | If humidity<50% | \_\_\_\_\_\_\_\_\_\_ |  |
|  |  |  | If humidity>50% | What is the humidity |  |
|  |  |  | If humidity is =50% | What is the humidity |  |
| 2 | The alert should come as a notification on the phone. | Continuous power supply, setting of the sensor,WIFI | If humidity>50% | ALERT! |  |
|  |  |  | If humidity = 50% | Maximum limit is reached |  |

# Table 2: Requirement with testing

AGILE MODEL

**Theme:** Memory foam

**Epic:** The memory foam has a tendancy to increase its density, to avoid this situation by avoiding the contact of excess humidity with it can be done by adding a humidity sensor, along with that arduino to make it IOT based so that notification immediately reaches the smart phone. To enable this continuous power supply and wifi connection is needed. The power supply has to be DC supply.

**User story 1:**

Memory foam is supposed to be made more durable.

**User story 2**:

The level of humidity which can affect the health of the memory foam should be intimidated on a smart phone.

**User story 3:**

Recommendations of how to protect the memory foam should also be provided.

#### ACTIVITY 4:

Electronic Calculator

An **electronic calculator** is typically a portable [electronic](https://en.wikipedia.org/wiki/Electronics) device used to perform [calculations](https://en.wikipedia.org/wiki/Calculation), ranging from basic [arithmetic](https://en.wikipedia.org/wiki/Arithmetic) to complex [mathematics](https://en.wikipedia.org/wiki/Mathematics).

The first [solid-state electronic](https://en.wikipedia.org/wiki/Solid-state_electronics) calculator was created in the early 1960s. Pocket-sized devices became available in the 1970s, especially after the [Intel 4004](https://en.wikipedia.org/wiki/Intel_4004), the first [microprocessor](https://en.wikipedia.org/wiki/Microprocessor), was developed by [Intel](https://en.wikipedia.org/wiki/Intel) for the Japanese calculator company [Busicom](https://en.wikipedia.org/wiki/Busicom" \o "Busicom). They later became used commonly within the [petroleum industry](https://en.wikipedia.org/wiki/Petroleum_industry) (oil and gas).

Modern electronic calculators vary from cheap, give-away, [credit-card-sized](https://en.wikipedia.org/wiki/ISO/IEC_7810) models to sturdy desktop models with built-in printers. They became popular in the mid-1970s as the incorporation of [integrated circuits](https://en.wikipedia.org/wiki/Integrated_circuit) 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](https://en.wikipedia.org/wiki/Operating_system) as far back as [early Unix](https://en.wikipedia.org/wiki/Ancient_UNIX) have included interactive calculator [programs](https://en.wikipedia.org/wiki/Computer_program) such as [dc](https://en.wikipedia.org/wiki/Dc_(computer_program)) and [hoc](https://en.wikipedia.org/wiki/Hoc_(programming_language)), and calculator functions are included in almost all [personal digital assistant](https://en.wikipedia.org/wiki/Personal_digital_assistant) (PDA) type devices, the exceptions being a few dedicated address book and dictionary devices.[5]

Ageing

They work fine mostly for around 5 to 6 years. Some companies even guarantee that their products work fine for 10 years, and mostly depending on the battery type. But it has evolved a long way to be what it is today.[6]

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Type of electronic calculator** | **Year** |
| 1 | Abacus | 2000 BC |
| 2 | Mechanical Calculator | 1642 |
| 3 | Counting machine was manufactured | 1851 |
| 4 | Grant Mechanical Calculating Machine | 1877 |
| 5 | Burroughs Adding Machine | 1886 |
| 6 | Curta calculator | 1948 |
| 7 | Vulva Tube calculator | 1961 |
|  | Transistor age calculator | 1967 |

**Fig 7:Evolution**

Costing

The cost varies from 100/- to 1000/- depending upon how advanced the calculator is. Some examples are:[7]

|  |  |  |
| --- | --- | --- |
| S.No. | Calculator | Price |
| 1. | Villy Dual Power Electronic Basic Blck Basic Calculator  (12 Digit) | 220/- |
| 2. | Casio MJ-120DPLUS-BU Desktop Basic Calculator  (12 Digit) | 447/- |
| 3. | CLTIZEN Dual Power Electronic Basic Calculator  (12 Digit) | 199/- |
| 4. | Meena Collection CT 512 Basic Electronic Basic Calculator  (12 Digit) | 140/- |
| 5. | Casio fx-991ES PLUS Scientific Calculator | 992/- |
| 6. | Casio FX-991ES Plus-2nd Edition Scientific Scientific Calculator  (12 Digit) | 1087/- |

**Fig 8: Costing**

Problem Statement

An electronic calculator which is capable of doing most of the mathematical operations including operations such as factorial and finding solution to quadratic equation.

SWOT analysis

1. Strength: The possibility of calculation of almost all kinds mathematical operations.
2. Weakness: It might not be able to solve all kind of polynomial equations.
3. Opportunity: scope of more development as it opens the doors for further advancement
4. Threat: the output can take little longer to come, because of the lot of background code.

Requirements

**High Level Requirements:**

1. The choice of operations to be given.
2. User friendly.
3. Result should be accurate, with the possible highest precision.

**Low Level requirement**:

1. Output should come very fast.

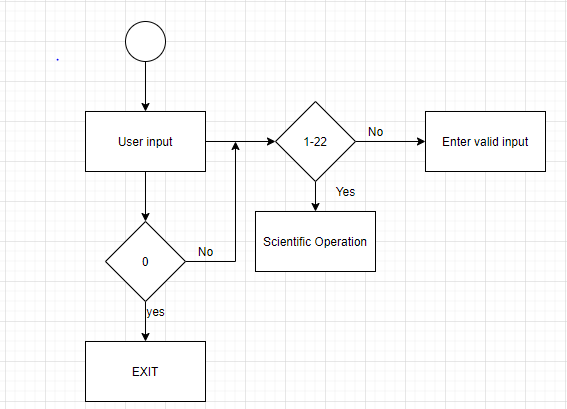
Test Plan

1.**Requirement based Test Plan**: the program will be menu driven, according to the menu if the user chooses addition, addition should happen, if the user chooses multiplication , multiplication of the numbers should happen.

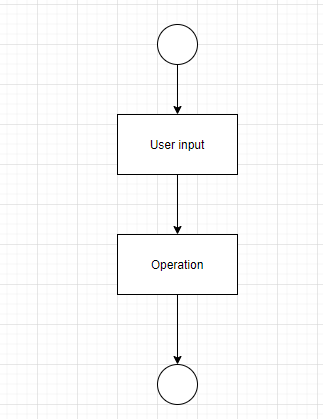
2.**Scenario based**: If there is a division by zero then there will be no output, as it is not defined.

3.**Boundary based Test Plan**: any operation which has an output as infinity is not defined.

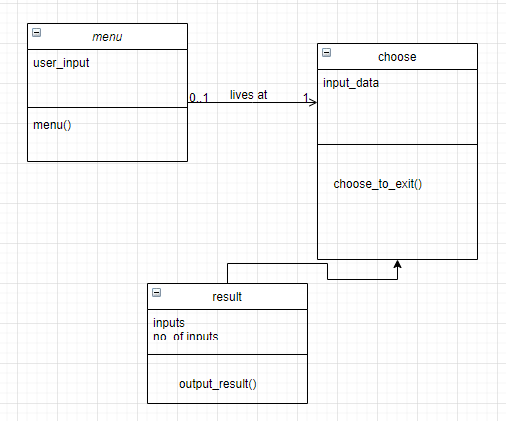
UML



**Fig 9: Activity UML**



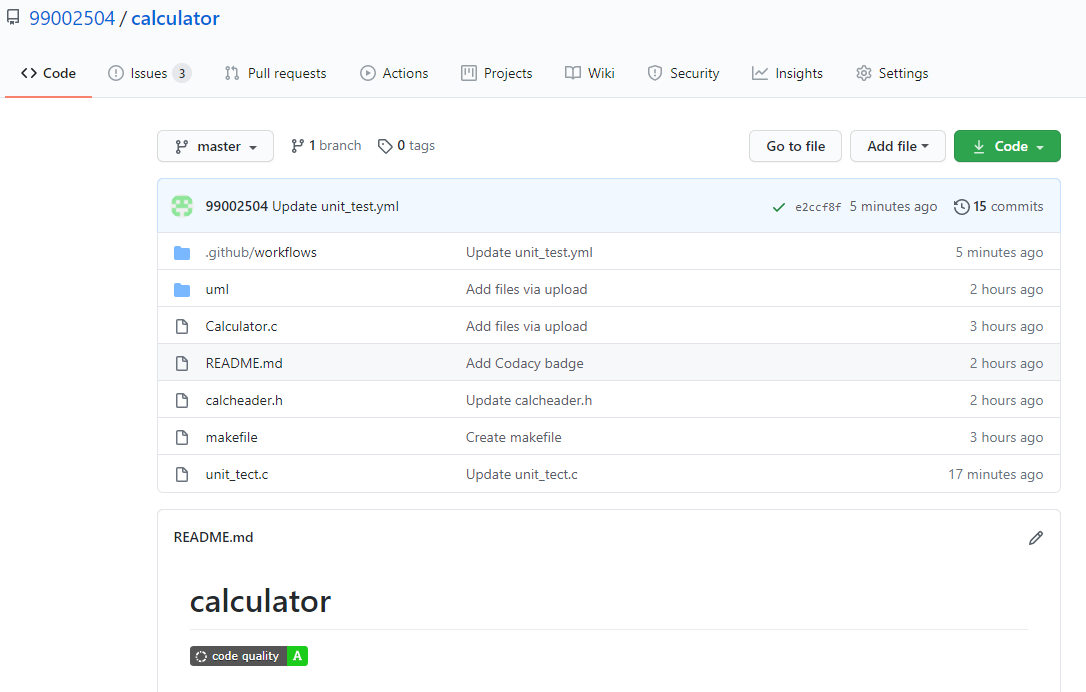
**Fig 10:State UML**



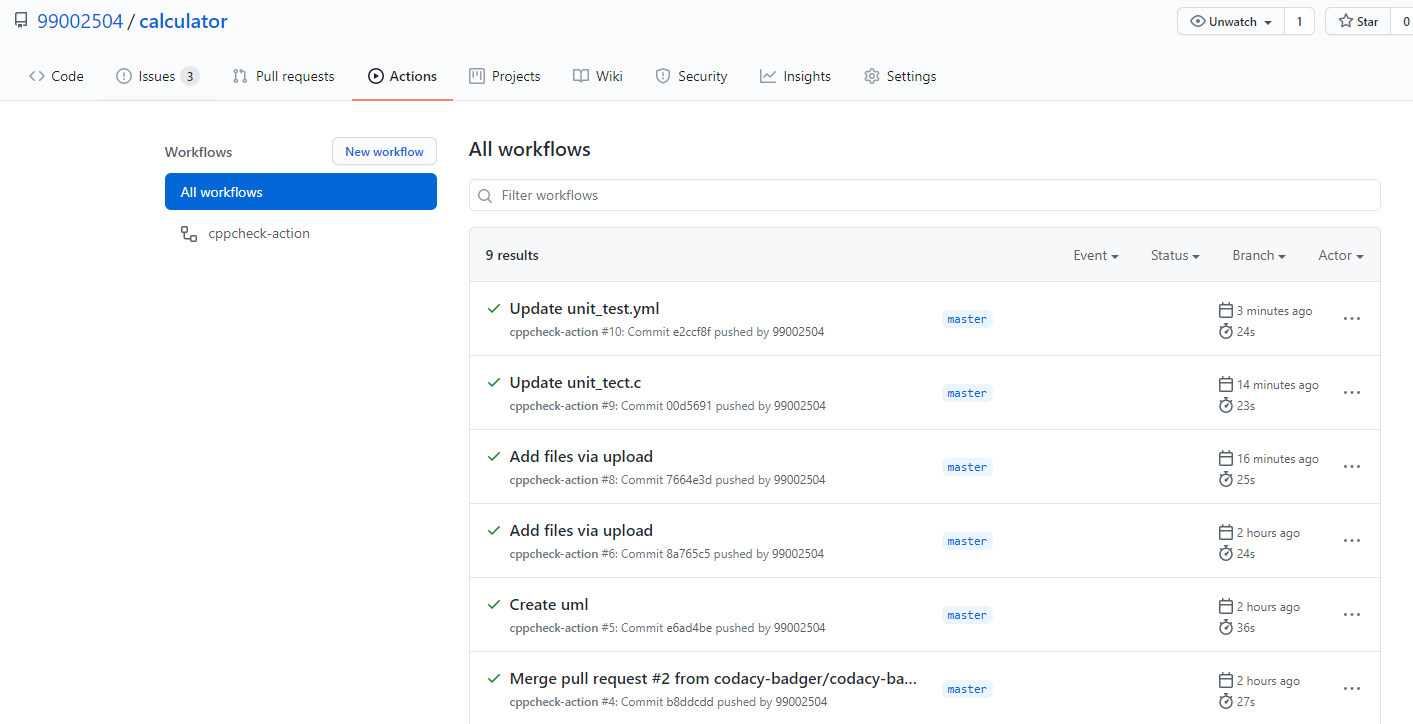
**Fig 11:Class UML**

GIT Summary

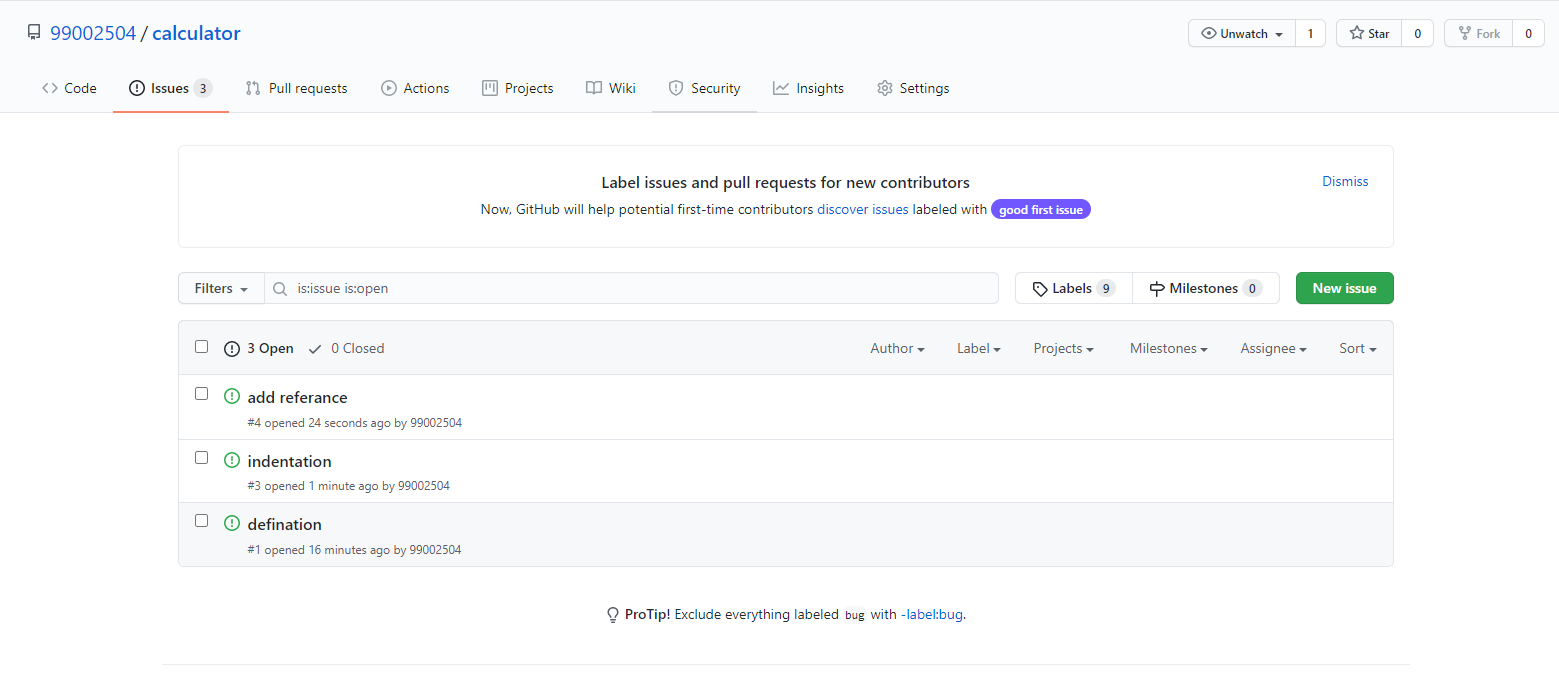
**Commits**



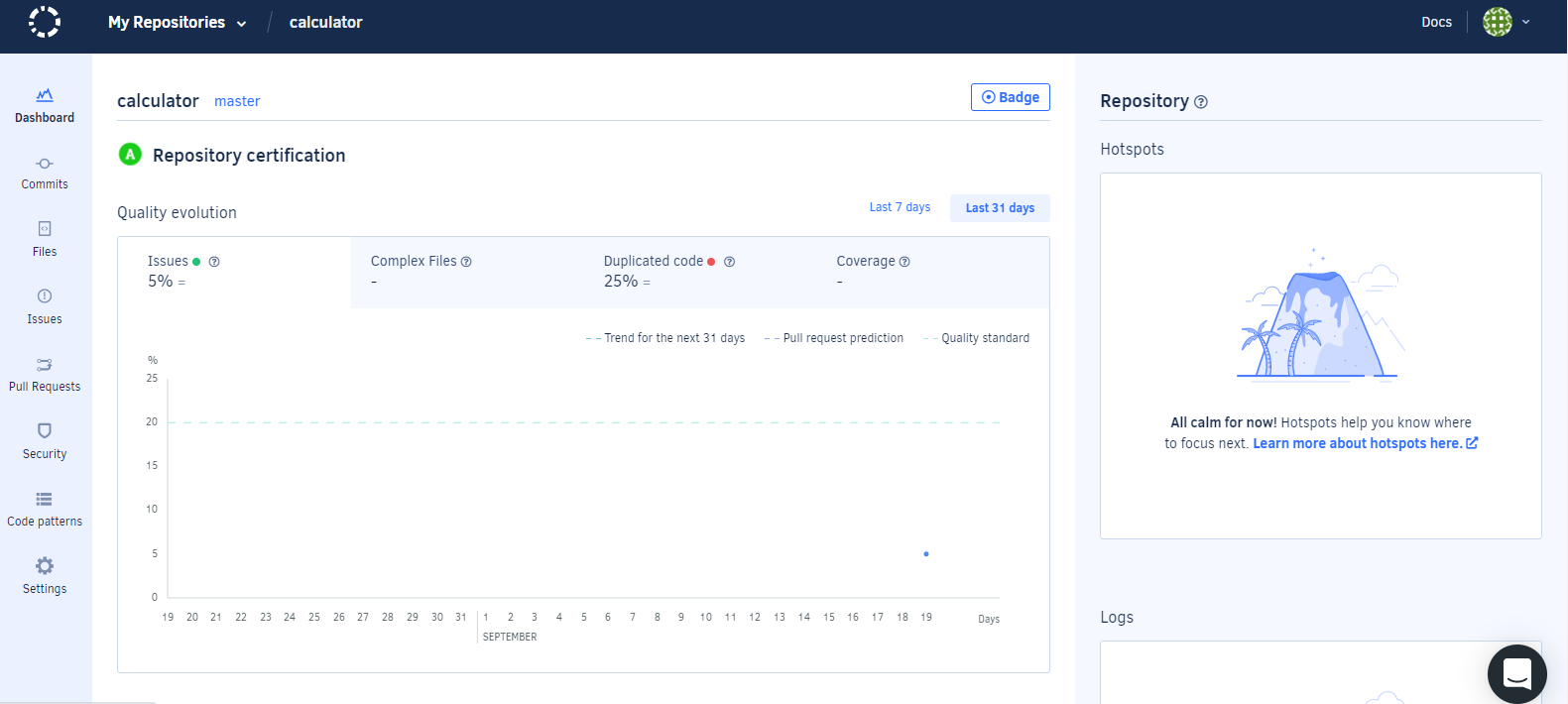
**Cpp check and Build**



**Issues**



**Codacy**



**Link for GIT:**

<https://github.com/99002504/calculator>

V MODEL

**High level testing:**

1. The entire menu driven program should run according to the user input.

**Low level testing:**

1. Each operation should give correct result when tested individually.

|  |  |
| --- | --- |
| ID | DESCRIPTION |
| 1 | User friendly. |
| 2 | Menu driven. |

**Table 3: Requirement**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Description | Preconditions | Expected  Input | Expected Output | Actual Output |
| 1 | User friendly | User should give an input | 1. To 22 | Numeric output | Numeric output |
| 2 | Menu driven | User should give an input | 0 | EXIT | EXIT |
|  |  |  | 1 | Addition | Addition |
|  |  |  | 2 | Subtraction | Subtraction |
|  |  |  | 13 | factorial | factorial |
|  |  |  | 14 | Modulus | Modulus |
|  |  |  | 21 | Cot(x) | Cot(x) |
|  |  |  | Division with denominator as 0 | \_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_ |

# Table 4: Requirement with testing

Reference

[1]wiki.naturalphilosophy.org/index.php?title=Charles\_A\_Yost

[2] <https://www.foamnights.com/types-of-foam>

[3] en.wikipedia.org/wiki/Relative\_humidity

[4] [https://www.electronicsforu.com/resources/electronics- components/humidity-sensor-basic-usage-parameter](https://www.electronicsforu.com/resources/electronics-%20%20components/humidity-sensor-basic-usage-parameter)

[5] https://en.wikipedia.org/wiki/Calculator

[6] https://www.wisegeek.com/what-are-some-types-of-calculators.htm

[7] [flipkart.com](https://www.electronicsforu.com/resources/electronics-%20%20components/humidity-sensor-basic-usage-parameter)

[8] https://codeforwin.org/2016/04/c-program-to-find-all-roots-of-quadratic-equation-using-switch.html

[9] https://www.sanfoundry.com/c-program-perform-matrix-multiplication/

[10] https://www.programiz.com/c-programming/examples/add-matrix

[11] https://www.sanfoundry.com/c-program-value-sin-x/

APPENDIX

#### **Header file**

|  |
| --- |
| #include<math.h> |
|  | #include<stdio.h> |
|  |  |
|  | float add(float a, float b) |
|  | { |
|  | return (a+b); |
|  | } |
|  | float sub(float a, float b) |
|  | { |
|  | return (a-b); |
|  | } |
|  | float mult(float a, float b) |
|  | { |
|  | return (a\*b); |
|  | } |
|  | float division(float a, float b) |
|  | { |
|  | return (a/b); |
|  | } |
|  | float squareroot(float a) |
|  | { |
|  | return sqrt(a); |
|  | } |
|  | float power(float a, float b) |
|  | { |
|  | return pow(a, b); |
|  | } |
|  | float square(float a) |
|  | { |
|  | return pow(a, 2); |
|  | } |
|  | float cube(float a) |
|  | { |
|  | return pow(a, 3); |
|  | } |
|  | float inverse(float a) |
|  | { |
|  | return pow(a, -1); |
|  | } |
|  | float yroot(float a, float b) |
|  | { |
|  | return pow(a, (1/b)); |
|  | } |
|  | float cuberoot(float a) |
|  | { |
|  | return pow(a, (1/3)); |
|  | } |
|  | float tenpow(float a) |
|  | { |
|  | return pow(10, a); |
|  | } |
|  | float factorial(float a) |
|  | { |
|  | float result = 1; |
|  | for(int i = 1; i <= a; i++) |
|  | { |
|  | result = result \* i; |
|  | } |
|  | return result; |
|  | } |
|  | float percent(float a, float b) |
|  | { |
|  | return (a \* b) / 100; |
|  | } |
|  | float logten(float a) |
|  | { |
|  | return log10(a); |
|  | } |
|  | int modulus(int a, int b) |
|  | { |
|  | return (a % b); |
|  | } |
|  | float trigosin(float a) |
|  | { |
|  | return sin(a \* 3.14159 / 180); |
|  | } |
|  | float trigocos(float a) |
|  | { |
|  | return cos(a \* 3.14159 / 180); |
|  | } |
|  | float trigotan(float a) |
|  | { |
|  | return tan(a \* 3.14159 / 180); |
|  | } |
|  | float trigocosec(float a) |
|  | { |
|  | return 1 / (sin(a \* 3.14159 / 180)); |
|  | } |
|  | float trigocot(float a) |
|  | { |
|  | return 1 / (tan(a \* 3.14159 / 180)); |
|  | } |
|  | float trigosec(float a) |
|  | { |
|  | return 1 / (cos(a \* 3.14159 / 180)); |
|  | } |
|  | void quadroots(float a, float b, float c) |
|  | { |
|  | float root1, root2, imaginary; |
|  | float discriminant; |
|  |  |
|  | /\* Calculate discriminant \*/ |
|  | discriminant = (b \* b) - (4 \* a \* c); |
|  |  |
|  | /\* Compute roots of quadratic equation based on the nature of discriminant \*/ |
|  | switch(discriminant > 0) |
|  | { |
|  | case 1: |
|  | /\* If discriminant is positive \*/ |
|  | root1 = (-b + sqrt(discriminant)) / (2 \* a); |
|  | root2 = (-b - sqrt(discriminant)) / (2 \* a); |
|  | printf("Two distinct and real roots exists: %.2f and %.2f", |
|  | root1, root2); |
|  | break; |
|  |  |
|  | case 0: |
|  | /\* If discriminant is not positive \*/ |
|  | switch(discriminant < 0) |
|  | { |
|  | case 1: |
|  | /\* If discriminant is negative \*/ |
|  | root1 = root2 = -b / (2 \* a); |
|  | imaginary = sqrt(-discriminant) / (2 \* a); |
|  | printf("Two distinct complex roots exists: %.2f + i%.2f and %.2f - i%.2f", |
|  | root1, imaginary, root2, imaginary); |
|  | break; |
|  |  |
|  | case 0: |
|  | /\* If discriminant is zero \*/ |
|  | root1 = root2 = -b / (2 \* a); |
|  | printf("Two equal and real roots exists: %.2f and %.2f", root1, root2); |
|  | break; |
|  | } |
|  | } |
|  |  |
|  | } |
|  | void matrixMultiplication(int m, int n, int p, int q, int first[10][10], int second[10][10]) |
|  | { |
|  | int multiply[10][10]; |
|  | int sum[10][10]; |
|  |  |
|  | for ( int c = 0 ; c < m ; c++ ) |
|  | { |
|  | for ( int d = 0 ; d < q ; d++ ) |
|  | { |
|  | for ( int k = 0 ; k < p ; k++ ) |
|  | { |
|  | sum = sum + ((first[c][k])\*(second[k][d])); |
|  | } |
|  | multiply[c][d] = sum; |
|  | sum = 0; |
|  | } |
|  | } |
|  |  |
|  | printf("Product of entered matrices:-\n"); |
|  |  |
|  | for ( int c = 0 ; c < m ; c++ ) |
|  | { |
|  | for ( int d = 0 ; d < q ; d++ ) |
|  | printf("%d\t", multiply[c][d]); |
|  | printf("\n"); |
|  | } |
|  | } |
|  | void matrixAddition(int m, int n, int first[10][10], int second[10][10]) |
|  | { |
|  | int sum[10][10]; |
|  |  |
|  | for (int i = 0; i < m; i++) |
|  | { |
|  | for (int j = 0; j < n; j++) |
|  | { |
|  | sum[i][j] = first[i][j] + second[i][j]; |
|  | } |
|  |  |
|  | printf("\nSum of two matrices: \n"); |
|  | for (int i = 0; i < m; i++) |
|  | { |
|  | for (int j = 0; j < n; j++) |
|  | { |
|  | printf("%d ", sum[i][j]); |
|  | if (j == m - 1) |
|  | { |
|  | printf("\n\n"); |
|  | } |
|  | } |
|  | } |
|  | } |
|  | } |
|  | int gcd(int x, int y) { |
|  | int r = 0, a, b; |
|  | a = (x > y) ? x : y; // a is greater number |
|  | b = (x < y) ? x : y; // b is smaller number |
|  |  |
|  | r = b; |
|  | while (a % b != 0) { |
|  | r = a % b; |
|  | a = b; |
|  | b = r; |
|  | } |
|  | return r; |
|  | } |
|  | int lcm(int x, int y) { |
|  | int a; |
|  | a = (x > y) ? x : y; // a is greater number |
|  | while (1) { |
|  | if (a % x == 0 && a % y == 0) |
|  | return a; |
|  | ++a; |
|  | } |
|  | }[8][9][10][11] |

#### **Source File**

|  |
| --- |
| #include<stdio.h> |
|  | #include<stdlib.h> |
|  | #include<math.h> |
|  | #include"calcheader.h" |
|  |  |
|  | int main() |
|  | { |
|  | int choice, i, a, b, res; |
|  | float x, y, z, result; |
|  | int m, n, p, q, c, d, k, sum = 0; |
|  | int first[10][10], second[10][10]; |
|  | do |
|  | { |
|  | printf("\nSelect your operation (0 to exit):\n"); |
|  | printf("1. Addition\n2. Subtraction\n3. Multiplication\n4. Division\n"); |
|  | printf("5. Square root\n6. X ^ Y\n7. X ^ 2\n8. X ^ 3\n"); |
|  | printf("9. 1 / X\n10. X ^ (1 / Y)\n11. X ^ (1 / 3)\n"); |
|  | printf("12. 10 ^ X\n13. X!\n14. %\n15. log10(x)\n16. Modulus\n"); |
|  | printf("17. Sin(X)\n18. Cos(X)\n19. Tan(X)\n20. Cosec(X)\n"); |
|  | printf("21. Cot(X)\n22. Sec(X)\n"); |
|  | printf("23. GCD \n24. LCM\n"); |
|  | printf("25. Roots of Quadratic Equation\n26. Matrix Addition\n27. Matrix Multiplication\n"); |
|  | printf("Choice: "); |
|  | scanf("%d", &choice); |
|  | if(choice == 0) exit(0); |
|  | switch(choice) |
|  | { |
|  | case 1: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x) ; |
|  | printf("\nEnter Y: "); |
|  | scanf("%f", &y); |
|  | result = add(x, y); |
|  | printf("\nResult: %f", result); |
|  | break; |
|  | case 2: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | printf("\nEnter Y: "); |
|  | scanf("%f", &y); |
|  | result = sub(x, y); |
|  | printf("\nResult: %f", result); |
|  | break; |
|  | case 3: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | printf("\nEnter Y: "); |
|  | scanf("%f", &y); |
|  | result = mult(x, y); |
|  | printf("\nResult: %f", result); |
|  | break; |
|  | case 4: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | printf("\nEnter Y: "); |
|  | scanf("%f", &y); |
|  | result = division(x, y); |
|  | printf("\nResult: %f", result); |
|  | break; |
|  | case 5: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | result = squareroot(x); |
|  | printf("\nResult: %f", result); |
|  | break; |
|  | case 6: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | printf("\nEnter Y: "); |
|  | scanf("%f", &y); |
|  | result = power(x, y); |
|  | printf("\nResult: %f", result); |
|  | break; |
|  | case 7: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | result = square(x); |
|  | printf("\nResult: %f", result); |
|  | break; |
|  | case 8: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | result = cube(x); |
|  | printf("\nResult: %f", result); |
|  | break; |
|  | case 9: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | result = inverse(x); |
|  | printf("\nResult: %f", result); |
|  | break; |
|  | case 10: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | printf("\nEnter Y: "); |
|  | scanf("%f", &y); |
|  | result = yroot(x, y); |
|  | printf("\nResult: %f", result); |
|  | break; |
|  | case 11: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | y = 3; |
|  | result = cuberoot(x); |
|  | printf("\nResult: %f", result); |
|  | break; |
|  | case 12: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | result = tenpow(x); |
|  | printf("\nResult: %f", result); |
|  | break; |
|  | case 13: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | result = factorial(x); |
|  | printf("\nResult: %.f", result); |
|  | break; |
|  | case 14: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | printf("\nEnter Y: "); |
|  | scanf("%f", &y); |
|  | result = percent(x, y); |
|  | printf("\nResult: %.2f", result); |
|  | break; |
|  | case 15: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | result = logten(x); |
|  | printf("\nResult: %.2f", result); |
|  | break; |
|  | case 16: |
|  | printf("Enter X: "); |
|  | scanf("%d", &a); |
|  | printf("\nEnter Y: "); |
|  | scanf("%d", &b); |
|  | res = modulus(a, b); |
|  | printf("\nResult: %d", res); |
|  | break; |
|  | case 17: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | result = trigosin(x); |
|  | printf("\nResult: %.2f", result); |
|  | break; |
|  | case 18: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | result = trigocos(x); |
|  | printf("\nResult: %.2f", result); |
|  | break; |
|  | case 19: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | result = trigotan(x); |
|  | printf("\nResult: %.2f", result); |
|  | break; |
|  | case 20: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | result = trigocosec(x); |
|  | printf("\nResult: %.2f", result); |
|  | break; |
|  | case 21: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | result = trigocot(x); |
|  | printf("\nResult: %.2f", result); |
|  | break; |
|  | case 22: |
|  | printf("Enter X: "); |
|  | scanf("%f", &x); |
|  | result = trigosec(x); |
|  | printf("\nResult: %.2f", result); |
|  | break; |
|  | case 23:printf("Enter X: "); |
|  | scanf("%d", &a); |
|  | printf("\nEnter Y: "); |
|  | scanf("%d", &b); |
|  | res = gcd(a, b); |
|  | printf("\nResult: %d", res); |
|  | break; |
|  | case 24: |
|  | printf("Enter X: "); |
|  | scanf("%d", &a); |
|  | printf("\nEnter Y: "); |
|  | scanf("%d", &b); |
|  | res = lcm(a, b); |
|  | printf("\nResult: %d", res); |
|  | break; |
|  | case 25: |
|  | printf("Enter values of a, b, c of quadratic equation (aX^2 + bX + c): "); |
|  | scanf("%f %f %f", &x, &y, &z); |
|  | quadroots(x, y, z); |
|  | break; |
|  | case 26: |
|  | printf("Enter the number of rows and columns of matrices\n"); |
|  | scanf("%d%d", &m, &n); |
|  | printf("Enter the elements of first matrix\n"); |
|  | for ( c = 0 ; c < m ; c++ ) |
|  | for ( d = 0 ; d < n ; d++ ) |
|  | scanf("%d", &first[c][d]); |
|  | printf("Enter the elements of second matrix\n"); |
|  | for ( c = 0 ; c < m ; c++ ) |
|  | for ( d = 0 ; d < n ; d++ ) |
|  | scanf("%d", &second[c][d]); |
|  |  |
|  | matrixAddition(m, n, first, second); |
|  | break; |
|  | case 27: |
|  | printf("Enter the number of rows and columns of first matrix\n"); |
|  | scanf("%d%d", &m, &n); |
|  | printf("Enter the elements of first matrix\n"); |
|  | for ( c = 0 ; c < m ; c++ ) |
|  | for ( d = 0 ; d < n ; d++ ) |
|  | scanf("%d", &first[c][d]); |
|  | printf("Enter the number of rows and columns of second matrix\n"); |
|  | scanf("%d%d", &p, &q); |
|  | if ( n != p ) |
|  | printf("Matrices with entered orders can't be multiplied with each other.\n"); |
|  | else |
|  | { |
|  | printf("Enter the elements of second matrix\n"); |
|  | for ( c = 0 ; c < p ; c++ ) |
|  | for ( d = 0 ; d < q ; d++ ) |
|  | scanf("%d", &second[c][d]); |
|  | } |
|  |  |
|  | matrixMultiplication(m, n, p, q, first, second); |
|  | break; |
|  | default: |
|  | printf("\nInvalid Choice!"); |
|  | } |
|  | printf("\nPress 1 to Continue or 0 to Exit\n"); |
|  | scanf("%d", &choice); |
|  | } while(choice); |
|  | return 0; |
|  | }//[8][9][10][11] |