

Awkash 1

Advanced Calculator

-  Quick Submit
-  Quick Submit
-  National University of Computer and Emerging Sciences, Islamabad

Document Details

Submission ID

trn:oid:::1:3419692088

11 Pages

Submission Date

Nov 21, 2025, 2:56 PM GMT+5

1,042 Words

Download Date

Nov 21, 2025, 2:59 PM GMT+5

5,512 Characters

File Name

PF_Project_Report_by_25k-0627,_25k-0590_1.pdf

File Size

357.3 KB

0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

0 AI-generated only 0%

Likely AI-generated text from a large-language model.

0 AI-generated text that was AI-paraphrased 0%

Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

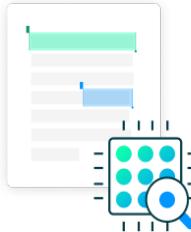
AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.



National University of Computer & Emerging Sciences

Karachi Campus



Advanced Calculator

Project Report

Programming Fundamentals

Section: BCS1-F

Group Members

**25k-0627 Awkash
25k-0590 Kasak Devi**

Abstract:

This project is a **Advanced Calculator** made in C language. It helps us to make our concept good in programming , it covers from basic arithmetic calculations to Matrix, roots, and history track with time etc. The system has Menu for each program like: Trigonometric, Unit conversion and the whole the control will remain in that menu until and unless user do not ask for exit. Since this was our first big console based C project, we practiced using functions, file handling, error handling, and creating a user friendly menu system. Overall, this project helped us to understand how different C concepts come together in one working concept file.

Our console based calculator will never overflow because it is designed with proper ERROR handling.

1. Introduction:

We have developed a console based **Advanced Calculator** using the **C Programming Language**. The system is to do multiple calculations at one place with user friendly environment, including Menu for different major calculations. This System includes Arithmetic, Scientific, Matrix, Quadratic equation solver, Unit conversion, show HISTORY. We understand from this how bigger programs are structured using functions, file handling, and give proper user desire. In a case if user enters a wrong input then it will warn it that you entered wrong input and will never overflow. It has a separate text file that will automatically update and save all calculations done by user & will also show the date and time when user performed it, and this is the main thing that keeps our project different from others.

2. Objectives :

In today's time, people have to calculate many things at a single platform. Keeping track of past calculations, handling , menus control, making sure to stay in a specific mode, taking inputs and making sure of error handling messages. So we tried to do a mini-version of a advanced calculator to make calculation and store that at one place for future access.

- To build a small but User friendly calculator with advance features.
- To add or subtract multiple numbers many times in very easy way.
- To learn and apply file handling, arrays and nested loops
- To store data permanently using text files.
- To practice for big work using functions to make every part as function our work is becoming easy.

The idea was not to create fancy(GUI) But to see how these students' tasks can be handled through programming.

Project Specification:

Our Advanced Calculator includes these types of operations:

- **Basic Operations**
- **Scientific Functions**
- **Matrix Operations**
- **Quadratic Equation Solver**
- **Unit Conversion**
- **Calculation History (with backend text files)**
- **Showing History any time during code**
- **Error Handling**
- **User-Friendly Interface**

Main features in our System:

- Menu(along with Exit each mode option)
- System (cls) to provide a amazing display control
- Saving of all calculations in files with that TIME of execution performed.
- If file is closed & opened again then previous data will NOT loss.
- Keep track of infinite cases in calculation
- If user want to perform only matrix operation then control will remain their till the user wants to exit that mode.

Flow of Advanced Calculator:

Start Choose Mode(:Mentioned above) Display inner menu Perform tasks

Data Saved display message for next operation Exit.

Problem Analysis:

Calculator usually handles many of the tasks at once. Some issues that we identified:

- Performing determinant without loop because it is of that type So we didn't use loop.
- Hard to handle error on each step (that increased size of our code)
- Display and manipulate all Menus with error handling.
- Handling infinite cases in trigonometric operations.

- To save data in text file with that corresponding calculation (eg. Addition, matrix, conversion, product etc)

Solution Design:

1. File Handling

- Calc_history.txt

2. Sub Menu(s)

- Each option has different options using nested switch statements.

3. Choice and execution system

When a user choose any mode:

- It goes according to choice (using switch)
- If it is a wrong choice it will display error
- Can use a menu infinite times.

4. History save and display method

- Every calculation saved with that time info and calculation mode.

Extra Features:

- User can watch history any time he wants.
- User can see time of that operation performance.
- After a calculation it will display “Press any key...” then the above content will be hidden for user and the next content will be shown.
- User friendly Menu

Implementation:

Language: C

Compiler/IDE: Code: Dev C++ / Visual studio (vs code)

All the information such as menu operations, sub menu info, and all other functions operations will save itself in text files so nothing gets lost when the program closes. We used loops and conditions to make a simple step-by-step menu that is easy for the user to follow. Each user type his/her choice, and it has separate functions so the program stays organized and easier to understand. Overall, the implementation is focused on keeping the code clean, dividing the work into small parts, and giving the user a smooth and simple experience.

Testing system (Output Screenshots):

- The welcome screen displays main menu and the user to interact and proceed accordingly. following are some outputs of my system:

Main Interface:

```
=====
Advanced Calculator
=====
1) Basic Arithmetic Operations.
2) Scientific Operations
3) Matrix Operations
4) Quadratic Equation Solver.
5) Units Conversion.
6) Watch History
7) Exit.

Enter choice: |
```

1. Sub menu after a choice (eg. Unit conversion)

Unit Conversion

1. Length Conversion
2. Temperature Conversion
3. Exit Unit Conversion Mode

Choice : 1

Length Convertor

2. Kilometers to Meters
3. Centimeters to Meters
4. Meters to Centimeters
5. Inches to Centimeters
6. Centimeters to Inches
7. Feet to Meters
8. Meters to Feet
9. Miles to Kilometers
10. Kilometers to Miles
11. Exit form Length mode

Choice = █

Calculation Example:

```
=====
          Trigonometric Operations
=====

1: sin
2: cos
3: tan
4: sec
5: cosec
6: cot
7: Exit trigonometric mode..

Enter choice: 3

1... for degrees
2... for radians
choose 1 or 2: 1

Enter angle: 90
tan(90.00) is INFINITY
Press Enter to continue...■
```



```
=====
          Trigonometric Operations
=====

1: sin
2: cos
3: tan
4: sec
5: cosec
6: cot
7: Exit trigonometric mode..

Enter choice: 1

1... for degrees
2... for radians
choose 1 or 2: 1

Enter angle: 23
sin(23.00) is 0.39
Press Enter to continue...■
```

History Display:

```
=====
Advanced Calculator
=====
1) Basic Arithmetic Operations.
2) Scientific Operations
3) Matrix Operations
4) Quadratic Equation Solver.
5) Units Conversion.
6) Watch History
7) Exit.

-----
Enter choice: 6

===== Saved History =====
[2025-11-20 23:17:50] Addition: sum = 18.0000
[2025-11-21 13:12:52] Trigonometric: tan(90.00 deg) = INFINITY
[2025-11-21 13:14:00] Trigonometric: sin(23.00 deg) = 0.390731
===== End of History =====
```

Press Enter to continue... █

Some Matrix Manipulations:

```
=====
```

Matrices

- ```
=====
```
- 1. Addition
  - 2. Subtraction
  - 3. Multiplication
  - 4. Determinant
  - 5. Exit Matrix Mode
- ```
-----
```

Enter Choice : 3

Enter rows: 1

Enter columns: 2

Enter Matrix1:

Enter Row 1 elements:

Enter [1][1] element: 4

Enter [1][2] element: 5

Enter rows of other Matrix : 2

Enter columns of other Matrix: 2

Enter Matrix2:

Enter Row 1 elements:

Enter [1][1] element: 2

Enter [1][2] element: 3

Enter Row 2 elements:

Enter [2][1] element: 4

Enter [2][2] element: 5

Matrices Multiplication =

28 37

Press Enter to continue...■

Conclusion:

The Advanced Calculator makes possible basic operations and advanced calculations at one application. work easier by organizing menu(s) ,sub menu, choices, error handling like infinite cases and wrong inputs and storing all calculation in text file at any instance. Building this project helped us to get real practice with file handling, use of functions, loops, multiple conditions, and some extra libraries.

- File handling
- math.h library
- time.h library
- Loops and conditional logic
- Stdlib.h, windows.h (for clear screen looks good in console)

Limitations

- No GUI based (Console-only)
- No advanced features like integration, derivation.

Future Improvements:

- Add a graphical user interface.
- Will try to reduce the length of code.
- Will add more functions (eg. Cubic eq. solver, derivative)

References:

- Let us c.
- geeksforgeeks.org
- class notes and online lectures (apna college, college wallah) and prior knowledge.