

A Project Report
on

Daily Step Tracker With Goal Monitoring



by

Yashfa Arif BCY243034
Rahimullah BCY243017

A Project Report submitted to the
DEPARTMENT OF ELECTRICAL AND COMPUTER
ENGINEERING
in partial fulfillment of the requirements for the degree of
BACHELORS OF SCIENCE IN CYBER SECURITY

Faculty of Engineering
Capital University of Science & Technology,
Islamabad

January, 2025

Daily Step Tracker With Goal Monitoring



by

Yashfa Arif BCY243034

Rahimullah BCY243017

A Project Report submitted to the
DEPARTMENT OF ELECTRICAL AND COMPUTER
ENGINEERING
in partial fulfillment of the requirements for the degree of
BACHELORS OF SCIENCE IN CYBER SECURITY

Faculty of Engineering
Capital University of Science & Technology,
Islamabad

January, 2025

Copyright © 2022 by CUST Student

All rights reserved. Reproduction in whole or in part in any form requires the prior written permission of Yashfa Arif BCY243034, Rahimullah BCY243017 or designated representative.

DECLARATION

It is declared that this is an original piece of our own work, except where otherwise acknowledged in text and references. This work has not been submitted in any form for another degree or diploma at any university or other institution for tertiary education and shall not be submitted by us in future for obtaining any degree from this or any other University or Institution.

Yashfa Arif
Reg. No. BCY243034

Rahimullah
Reg. No. BCY243017

January, 2025

CERTIFICATE OF APPROVAL

It is certified that the project titled “Daily Step Tracker With Goal Monitoring” carried out by Yashfa Arif Reg. No. BCY243034, Rahimullah Reg. No. BCY243017, under the supervision of Mr. Syed Waqas Ayub Shah, Capital University of Science & Technology, Islamabad, is fully adequate, in scope and in quality, as a first semester project for the degree of BS Cyber Security.

Supervisor:

Mr. Syed Muhammad Waqas Ayub Shah
Lecturer
Department of Electrical and Computer Engineering
Faculty of Engineering
Capital University of Science & Technology, Islamabad

HOD:

Dr. Noor Mohammad Khan
Professor
Department of Electrical and Computer Engineering
Faculty of Engineering
Capital University of Science & Technology, Islamabad

ACKNOWLEDGMENT

We would like to thank “Mr. Syed Muhammad Waqas Ayub Shah” whose guidance helped to complete our project in the given time. This project cannot be completed without the effort and co-operation of group member.

ABSTRACT

This project report presents a Daily Step Tracker with Goal Monitoring system designed to promote healthy living by tracking daily steps and providing feedback on goal achievement. The system, developed in C++, features user-friendly input validation, looping for valid input, and clear motivational output. The program structure consists of a main function, stepTracker function, and input validation. The report outlines the program's objectives, features, advantages, and limitations. The system's flowchart and program structure are also presented. Overall, the Daily Step Tracker with Goal Monitoring system provides an efficient and effective way to monitor daily step progress and achieve fitness goals.

Table of Contents

DECLARATION	iii
CERTIFICATE OF APPROVAL	iv
ACKNOWLEDGMENT	5
ABSTRACT	6
Chapter 1	8
INTRODUCTION	8
1.1 Overview	8
1.2 Project Idea	8
1.3 Purpose of Project	8
1.4 Project Significance	8
1.5 Application of the Project	9
1.6 Project Plan	9
Chapter 2	11
Program Objectives	11
Program Objectives	11
2.1 Program Features	11
2.2 Program Structure	11
2.3 Program Advantages	12
2.4 Program Limitations	12
Program Structure	13
3.1 Main Function	13
3.2 stepTracker Function	14
3.3 Input Validation	14
3.4 Program Flow:	14
Chapter 4	16
Flowchart	16
Chapter 5	18
Conclusion	18
References	19

Chapter 1

INTRODUCTION

The provided C++ program is designed to help users track their daily steps and compare them to a set goal. It includes basic input validation to ensure user input is valid. Additionally, it provides feedback on whether the user met, exceeded or fell short of their step goal.

1.1 Overview

The program is written in C++ that starts by asking the user to enter their daily step goal. It prompts the user for their step goal and number of steps taken. The program then compares the steps taken to the goal and provides feedback on whether the user met, exceeded or fell short of their target.

1.2 Project Idea

The project idea is to develop a simple daily step tracker with goal monitoring that can be used by individuals who want to monitor their daily physical activity and maintain a healthy lifestyle.

1.3 Purpose of Project

The purpose of this project is to promote healthy living and encourage user to stay physically active by tracking their daily steps and providing feedback. With a straight forward interface for manual step tracking, it ensures accessibility to a diverse range of users. Additionally, the project serves a demonstration of core programming principles such as input validation, loops and conditional logic, highlighting the integration of technology with health promotion and educational value.

1.4 Project Significance

This project holds significance in both health promotion and technology education. It encourages users to adopt an active lifestyle by tracking daily step goals, promoting physical well-being, and reducing sedentary behavior. Its user-friendly design makes step tracking accessible to people of all ages, even without advanced fitness devices. The project

demonstrates essential programming concepts, making it a valuable tool for students and developers learning C++.

1.5 Application of the Project

Individuals can use the program to monitor their daily activity levels and achieve their fitness goals. It can be adapted for use in wellness programs, helping patients or elderly individuals track their steps for prescribed activity targets. This program can also be used to organize step-tracking competitions, encouraging group fitness challenges in workplaces or schools.

1.6 Project Plan

The project plan includes the following steps:

1. Define the project requirements.
2. Design the program structure.
3. Implement the program.
4. Test the program.
5. Debug the program.
6. Document the program.

1.6.1 Define the Project Requirements

- Identify the project's objectives and scope.
- Determine the functional and non-functional requirements of the system.
- Define the user interface and user experience requirements.
- Identify the hardware and software requirements.

1.6.2 Design the Program Structure

- Develop a detailed design of the program's structure.
- Define the program's modules and their interactions.
- Determine the data structures and algorithms to be used.
- Create a detailed design document.

1.6.3 Implement the Program

- Write the program code based on the design document.
- Implement the program's modules and their interactions.
- Test each module individually.
- Integrate the modules into a complete program.

1.6.4 Test the Program

- Develop a testing plan to ensure the program meets the requirements.
- Test the program's functionality and performance.
- Identify and fix any bugs or errors.
- Conduct user acceptance testing.

1.6.5 Debug the Program

- Identify and fix any bugs or errors found during testing.
- Use debugging tools to identify and fix issues.
- Conduct thorough testing to ensure the program is error-free.

1.6.6 Document the Program

- Create a user manual to guide users in using the program.
- Develop a technical document to describe the program's design and implementation.
- Create a maintenance document to guide future maintenance and updates.
- Document any known issues or limitations

Chapter 2

Program Objectives

Program Objectives

The objectives of the program are:

1. To provide simple and efficient way of input validation for goal and step taken.
2. To track progress against goal.
3. To provide user-friendly feedback.
4. To provide a user-friendly interface for users to interact with the program.

2.1 Program Features

The program has the following features:

- 1. User Input For Goal And Steps Taken:** The program prompts the user to enter their daily step goal and the actual number of steps they've taken.
- 2. Input Validation:** Ensures that the goal is a positive integer and the number of steps is non-negative. Handles invalid inputs (like letters or negative numbers) by prompting the user again until valid data is entered.
- 3. Feedback On Goal Achievement:** It compares the user's steps taken to the goal and provides feedback.
- 4. Looping For Valid Input:** The program uses loops to repeatedly prompt the user for valid input if valid data is entered, ensuring a smooth user experience.
- 5. Clear And Motivational Output:** The program provides clear user-friendly messages that motivate the user to stay on track with their fitness goals.
- 6. Modular Design:** The logic for tracking steps and giving feedback is separated into a function (stepTracker), promoting cleaner, more maintainable code.

2.2 Program Structure

The program consists of the following components:

- 1. Main Function:** The main function is the entry point of the program.
- 2. Conditions:** If, else-if and else statements are used to check the conditions.
- 3. Loop:** While loop is used for specific iterations.
- 4. Function:** The program consists of one function which performs a specific task.

2.3 Program Advantages

The program has the following advantages:

- 1. Easy to Use:** The program is easy to use, even for users who are not familiar with computers.
- 2. Efficient:** The program is efficient and can handle a large number of employee and department records.
- 3. Accurate:** The program is accurate and ensures that data is accurate and consistent.

2.4 Program Limitations

The program has the following limitations:

- 1. No Support For Multiple Users:** The program is designed to track steps for only one user per session. It doesn't allow for tracking multiple user's goals and steps within the same run of program.
- 2. Lack Of Time Based Tracking:** The program does not track time or data associated with the step taken.

Chapter 3

Program Structure

```
#include <iostream>
using namespace std;
void stepTracker(int stepsTaken, int goal) {
    if (stepsTaken < goal) {
        cout << "You did not meet your goal. You need to take " << (goal - stepsTaken) << " more steps." << endl;
    } else if (stepsTaken == goal) {
        cout << "Congratulations, you met your goal!" << endl;
    } else {
        cout << "Congratulations, you exceeded your goal by " << (stepsTaken - goal) << " steps! Great job!" << endl;
    }
}

int main() {
    cout << "Welcome to the StepTracker Device!" << endl << endl;
    int goal, stepsTaken;
    while (true) {
        cout << "Enter your daily step goal: " << endl;
        cin >> goal;
        if (cin.fail() || (goal <= 0)) {
            cin.clear(); // remove error
            cin.ignore(); // clear all the mess
            cout << "Invalid input. Please enter an integer value only." << endl;
        } else {
            break;
        }
    }
    while (true) {
        cout << "Enter the number of steps you have taken today: " << endl;
        cin >> stepsTaken;
        if (cin.fail() || (stepsTaken <= 0)) {
            cin.clear();
            cin.ignore();
            cout << "Invalid input. Please enter a non-negative number for steps taken." << endl;
        } else {
            break;
        }
    }
    stepTracker(stepsTaken, goal);

    return 0;
}
```

3.1 Main Function:

- The main function is the entry point of the program.
- It contains two while loops to validate user input for the daily step goal and the number of steps taken.
- Once valid input is obtained, it calls the stepTracker function to evaluate the user's progress.

3.2stepTracker Function:

- This function takes two integer parameters: stepsTaken and goal.
- It evaluates the user's progress by comparing stepsTaken to goal.
- Based on the comparison, it provides feedback to the user through cout statements.

```
Welcome to the StepTracker Device!

Enter your daily step goal:
2000
Enter the number of steps you have taken today:      •
1500
You did not meet your goal. You need to take 500 more steps.
-----
```

3.3Input Validation:

- The program uses two while loops to validate user input.
- The first loop checks if the user's input for the daily step goal is a positive integer.
- The second loop checks if the user's input for the number of steps taken is a non-negative integer.

```
Welcome to the StepTracker Device!

Enter your daily step goal:
2000
Enter the number of steps you have taken today:
y
Invalid input. Please enter a non-negative number for steps taken.
```

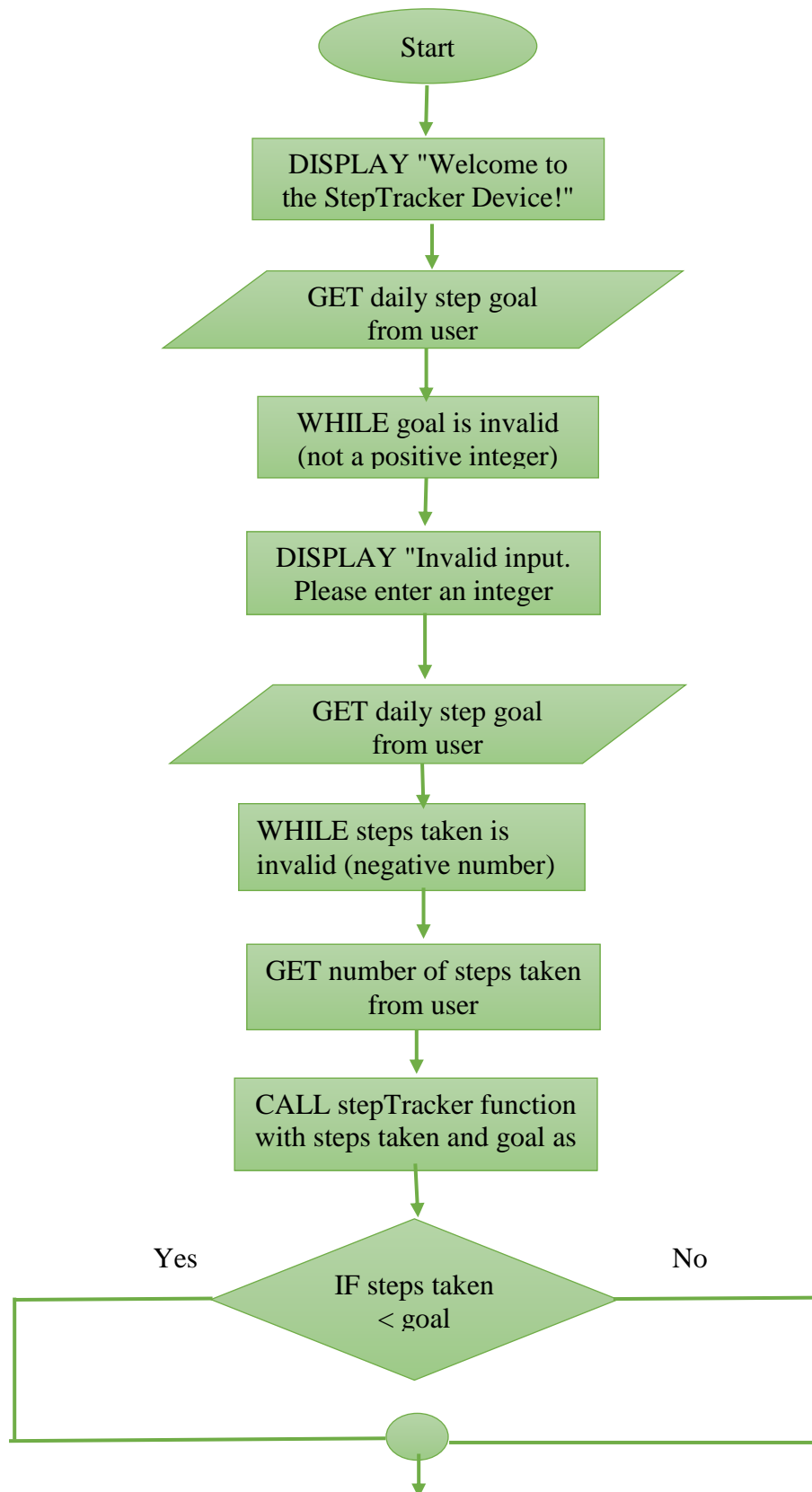
3.4Program Flow:

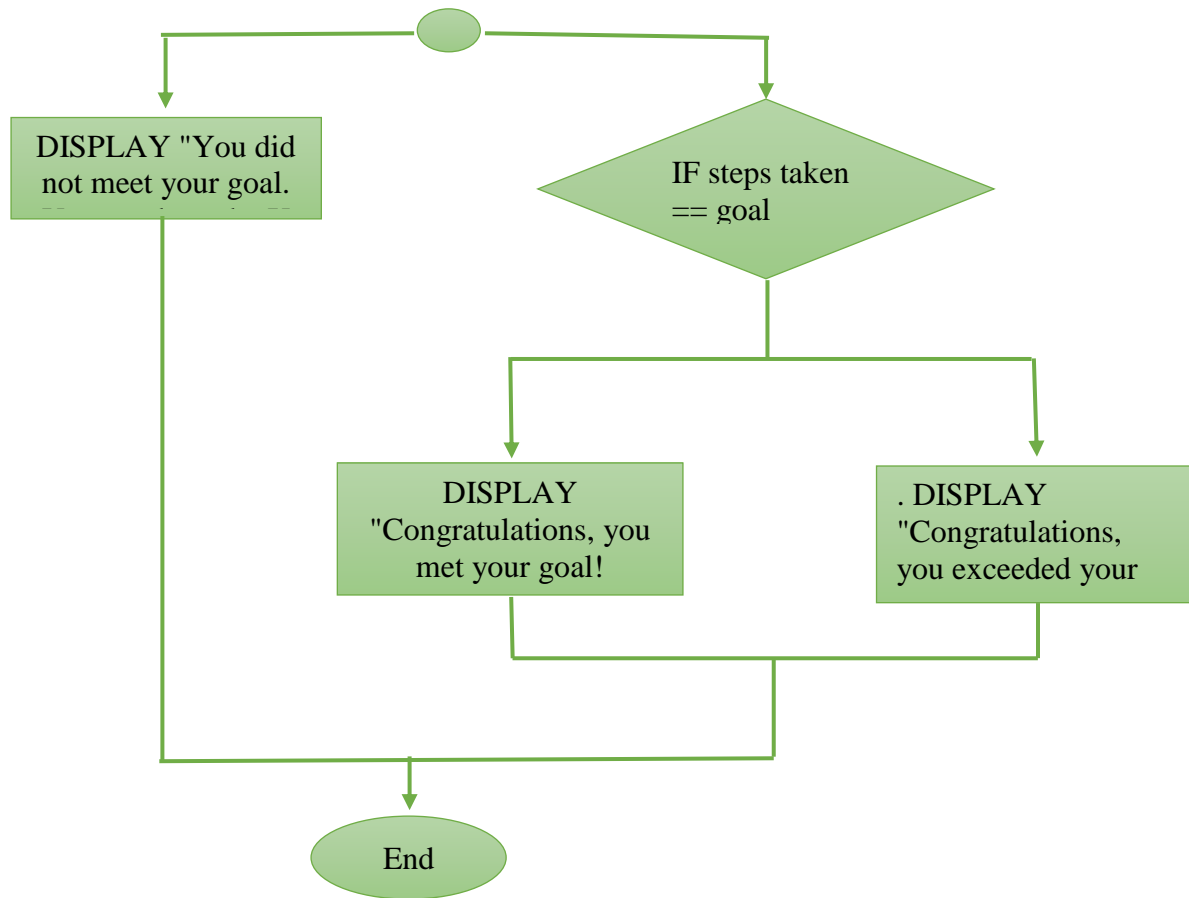
- The program starts by prompting the user to input their daily step goal.
- The program validates the user's input using a while loop.
- steps taken.
- The Once valid input is obtained, the program prompts the user to input the number of
- program validates the user's input using another while loop.

- Once valid input is obtained, the program calls the stepTracker function to evaluate the user's progress.
- The stepTracker function provides feedback to the user based on their progress.

Chapter 4

Flowchart





Chapter 5

Conclusion

The program is structured to provide a simple and intuitive interface for tracking daily step progress. The use of input validation ensures that the program handles invalid user input robustly. The stepTracker function provides clear and concise feedback to the user, making it easy to understand their progress.

References

-C++ Standard Library

-<https://en.cppreference.com/w/>

Standard Library

-<http://www.cplusplus.com/reference/>

- If-Else Statements

<https://en.cppreference.com/w/cpp/language/if-documentation> for the if statement

-<http://www.cplusplus.com/doc/tutorial/control/documentation> for the if statement

- Cin and Input Validation

-https://en.cppreference.com/w/cpp/io/basic_istream

- Error Handling

-<https://en.cppreference.com/w/cpp/error/exception> - documentation for the `std::exception` class

-<http://www.cplusplus.com/reference/exception/exception/> documentation for the `std::exception` class