

C PROGRAMMING PROJECT: CURRENCY CONVERTER

GROUP NAME:
ZENCODERS

GROUP MEMBERS:

- MUHAMMAD TAYYAB KHAN (CT-25088)
- MUHAMMAD AYAAN (CT-24288)
- VISHAL KUMAR (CT-25081)

A software tool for seamless currency conversion



PROJECT OVERVIEW





TITLE SLIDE

Project Overview

The project aims to develop a robust and scalable currency converter using the C programming language.

Team Composition

The development team, ZenCoders, consists of three members collaborating on this application.

User-Friendly Design

The converter is designed to be user-friendly and reads currency rates from an external data file.



TABLE OF CONTENTS

Structured Presentation Sections

The presentation is divided into clear sections covering introduction, technical design, and future improvements.

Comprehensive Project Overview

Covers project inception, technical details, code analysis, program flow, and conclusion for full understanding.

Final Reflections and Thanks

Ends with a conclusion and thank you slide to summarize and appreciate the audience.

PROJECT INTRODUCTION



INTRODUCTION & OBJECTIVES



Project Challenges

Existing currency converters are often hard-coded, lack scalability, and can crash frequently, limiting usability.

Primary Objective

Develop a robust, scalable, and user-friendly console application using the C programming language.

Data-Driven Approach

Use an external file `rates.txt` to source all currency data, enabling easy updates without changing core code.

TECHNICAL DESIGN



CORE FEATURES & TECHNICAL DESIGN

External File Handling

Reads currency data from an external file to support over 180 currencies dynamically without code changes.

Intuitive User Interface

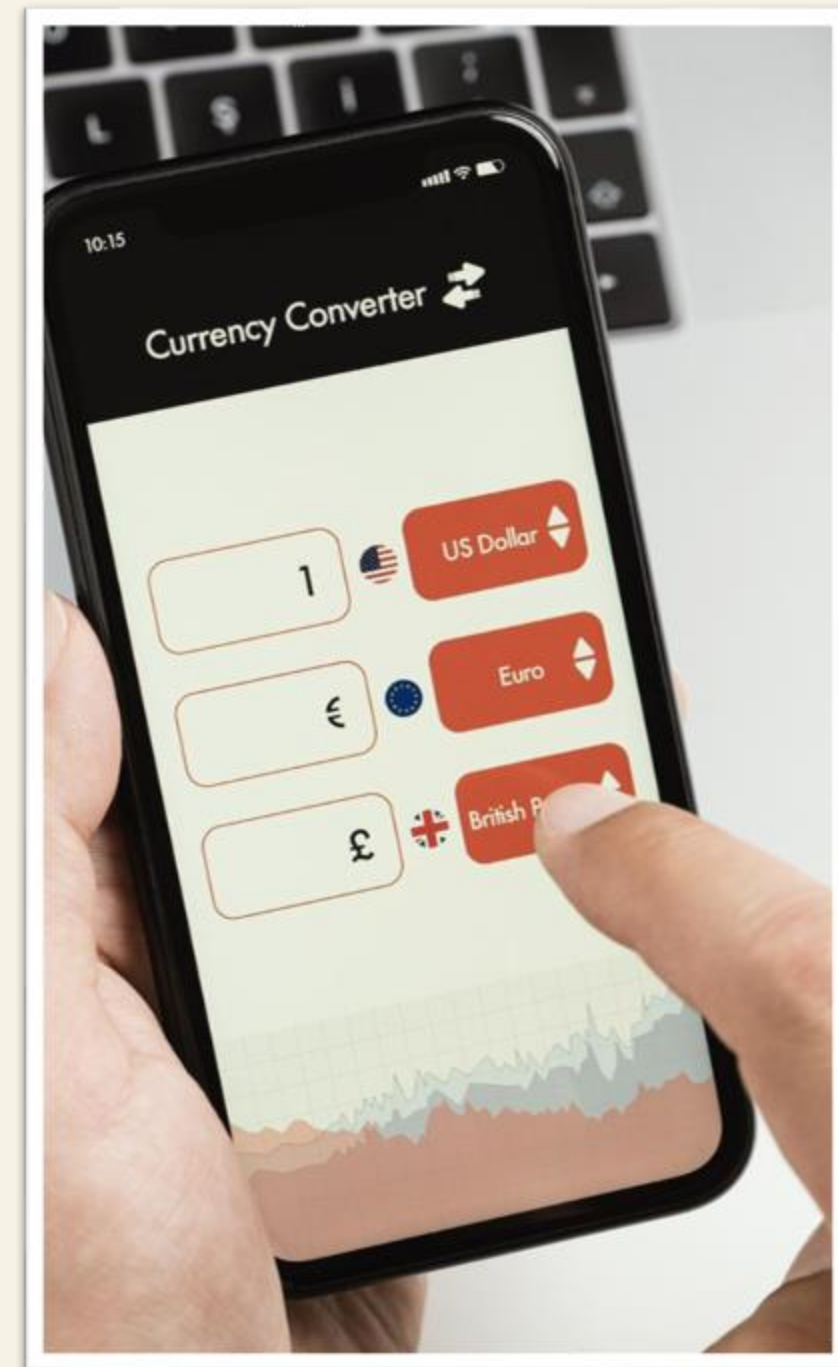
Offers three main user options: Convert currencies, list all currencies, and exit the application easily.

Robust Input Validation

Prevents crashes by validating user input and rejecting invalid entries like letters instead of numbers.

Efficient Conversion Logic

Uses a single base currency (USD) for all conversions to simplify calculations and improve accuracy.



CODE IMPLEMENTATION



CODE ANALYSIS: HOW IT WORKS

Structured Currency Data

A C struct named Currency organizes currency information such as code, name, and rate for effective data handling.

File Reading Mechanism

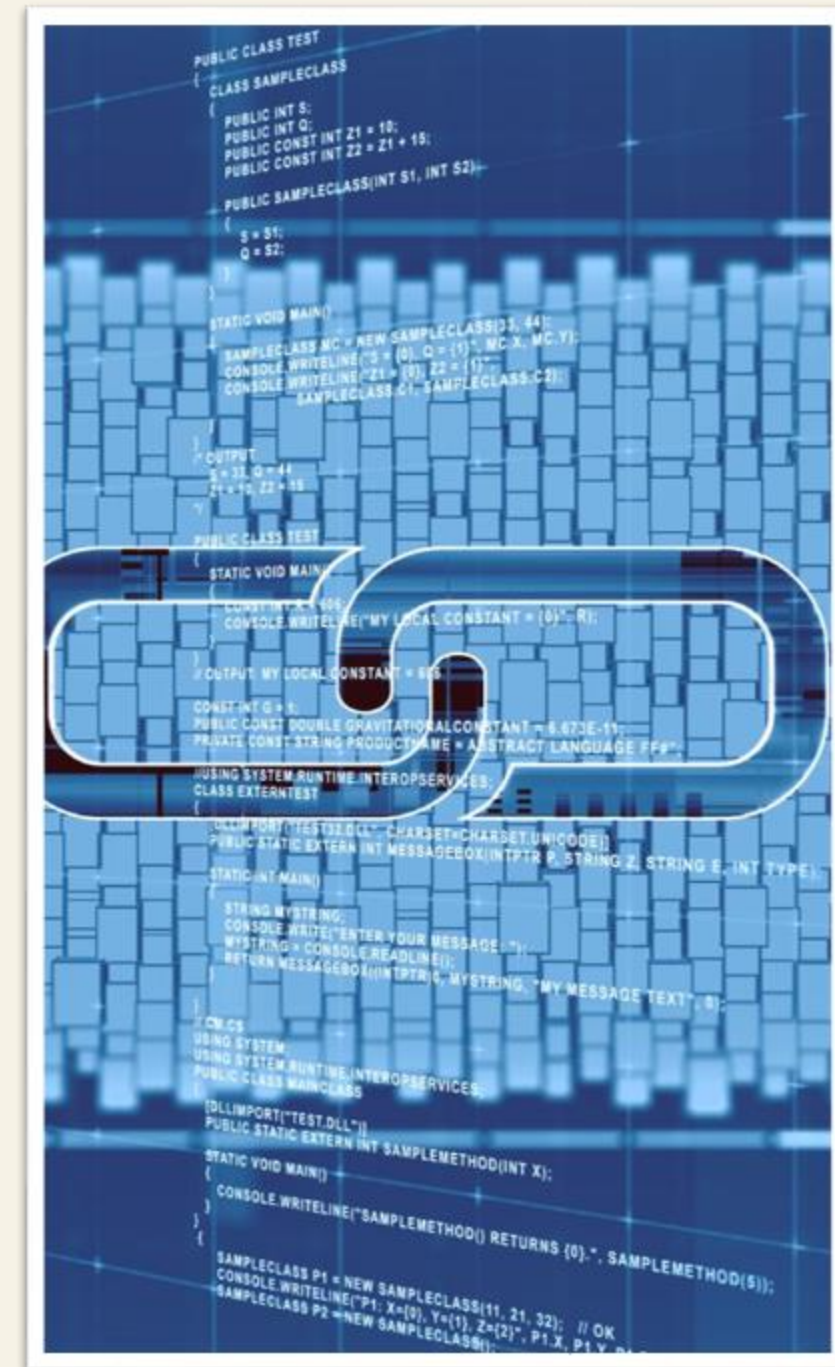
The program reads exchange rates from rates.txt file line-by-line using fopen() and fscanf() within a while loop.

Two-Step Conversion Process

Currency conversion happens in two steps: from source currency to USD, then USD to target currency ensuring accuracy.

Conversion Formula

The formula $\text{final_amount} = (\text{base_amount} / \text{from_rate}) * \text{to_rate}$ guarantees consistent and accurate currency conversions.



LIVE CODE WALKTHROUGH



We will now go line-by-line through the C code.

VISUAL REPRESENTATION



PROGRAM FLOWCHART

Visual Logic Representation

The flowchart visually represents the application's logic and sequence of operations for clarity and understanding.

User Interaction Flow

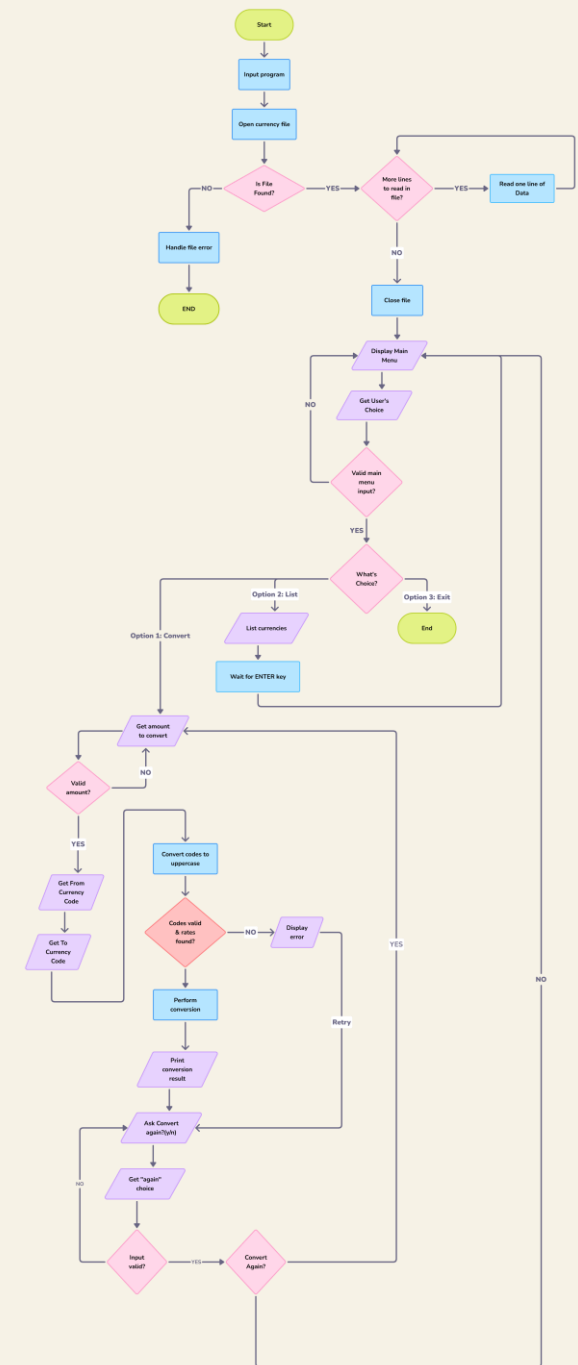
It outlines user input handling and menu navigation to demonstrate program interaction flow.

Currency Conversion Process

The flowchart details the currency conversion steps, showcasing how data is processed and converted.

Program Initialization and Exit

It also includes program start and exit points to complete the operational sequence.



ENHANCEMENT IDEAS



FUTURE IMPROVEMENTS

Graphical User Interface

Integrate a GUI with interactive elements like drop-down menus to replace the console interface for a better user experience.

Live Exchange Rates

Incorporate live exchange rate data using an API and libcurl library to ensure real-time accuracy and relevance.

Search Functionality

Add a search feature to quickly find specific currencies, improving usability and efficiency of the currency list.



PROJECT SUMMARY



CONCLUSION

C Programming Application

The project illustrates effective use of C programming concepts including structs and file input/output operations.

Robust and Scalable Tool

Input validation and program logic separation ensure the application is robust and maintainable for future scaling.

Academic and Practical Value

The project meets academic goals and lays groundwork for future enhancements and real-world utility.

CLOSING REMARKS





THANK YOU

Gratitude and Appreciation

Express sincere thanks to the audience for their time and attention during the presentation.

Open for Questions

Invite questions and feedback from the audience to foster interaction and engagement.

Future Learning and Development

Highlight the project's role as a valuable learning experience and the commitment to future growth.