

# Project Report: Advanced Currency Converter

## 1. Title Page

**Project Title:** Advanced Currency Converter in C Programming

**Submitted By:** Atulya Singh

**Course:** BTech CSE (Batch 20)

**Instructor:** Rahul Prasad

**Date:** 24-11-2025

---

## 2. Abstract

This project presents an Advanced Currency Converter developed using the C programming language. The program allows users to convert monetary values between six widely used international currencies, including USD, EUR, GBP, INR, JPY, and AUD. The system provides a user-friendly interface, handles invalid inputs, and displays accurate results using predefined exchange rates. This report includes the problem definition, system design, implementation details, testing, results, and future improvements.

---

## 3. Problem Definition

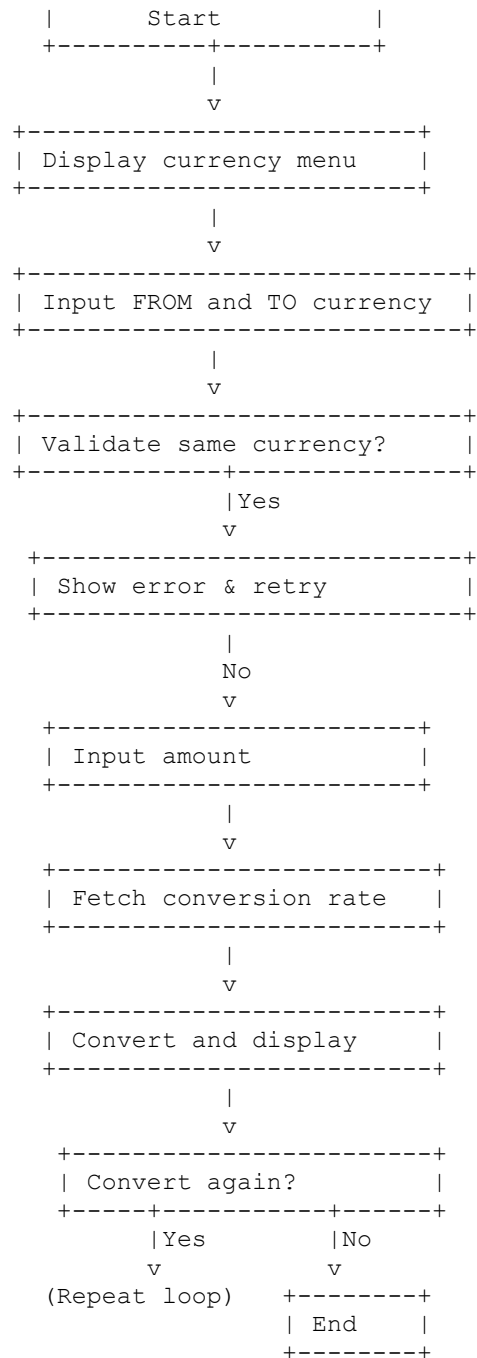
With globalization, currency conversion has become an essential need for travelers, businesses, and online shoppers. Manual conversion using calculators or searching online may be time-consuming. The goal of this project is to design an efficient currency converter in C that:

- Supports multiple currencies.
  - Performs accurate and instantaneous conversions.
  - Allows repeated conversions.
  - Checks invalid or repeated currency choices.
  - Provides a clean and interactive user experience.
- 

## 4. System Design

### 4.1 Flowchart

+-----+



## 4.2 Algorithm

1. Start the program.
2. Show the currency list.
3. Ask the user to enter source currency.
4. Ask the user to enter target currency.
5. If both currencies are the same, show an error and restart.
6. Ask for the amount.
7. Retrieve the conversion rate from a 2D rate matrix.
8. Multiply amount  $\times$  rate.

9. Display the result.
  10. Ask if the user wants to convert again.
  11. If yes, repeat; otherwise end.
- 

## 5. Implementation Details

Below is the implementation written in C:

```
#include <stdio.h>
#include <stdlib.h>

void showCurrencies();
double convert(double amount, double rate);
void clearScreen();
double getRate(int from, int to);

int main() {
    int fromCurrency, toCurrency;
    double amount, result;
    char cont;

    do {
        clearScreen();
        printf("Currency conveter\n\n");

        showCurrencies();

        printf("\nEnter FROM currency (1-6): ");
        scanf("%d", &fromCurrency);

        printf("Enter TO currency (1-6): ");
        scanf("%d", &toCurrency);

        if (fromCurrency == toCurrency) {
            printf("\nError: Both currencies cannot be the same!\n");
            printf("Press Enter to try again...");
            getchar(); getchar();
            continue;
        }

        printf("\nEnter amount: ");
        scanf("%lf", &amount);

        double rate = getRate(fromCurrency, toCurrency);

        if (rate == 0.0) {
            printf("\nConversion rate not found!\n");
        } else {
            result = convert(amount, rate);
            printf("\n %.2f (Currency %d) = %.2f (Currency %d)\n",
amount, fromCurrency, result, toCurrency);
        }

        printf("\nDo you want to convert again? (y/n): ");
        scanf(" %c", &cont);
    } while (cont == 'y');
```

```

    } while (cont == 'y' || cont == 'Y');

    printf("\nThank you for using the Advanced Converter!\n");
    return 0;
}

```

## Additional Functions

```

void showCurrencies() {
    printf("Available Currencies:\n");
    printf("1. USD - United States Dollar\n");
    printf("2. EUR - Euro\n");
    printf("3. GBP - British Pound\n");
    printf("4. INR - Indian Rupee\n");
    printf("5. JPY - Japanese Yen\n");
    printf("6. AUD - Australian Dollar\n");
}

void clearScreen() {
#ifdef _WIN32
    system("cls");
#else
    system("clear");
#endif
}

double convert(double amount, double rate) {
    return amount * rate;
}

double getRate(int from, int to) {
    double rates[6][6] = {
        {1.0, 0.91, 0.78, 83.20, 143.50, 1.48},
        {1.10, 1.0, 0.86, 91.20, 158.40, 1.62},
        {1.28, 1.16, 1.0, 105.80, 180.22, 1.88},
        {0.012, 0.011, 0.0094, 1.0, 1.71, 0.018},
        {0.007, 0.0063, 0.0055, 0.58, 1.0, 0.010},
        {0.67, 0.62, 0.53, 54.40, 98.00, 1.0}
    };

    if (from < 1 || from > 6 || to < 1 || to > 6)
        return 0.0;

    return rates[from - 1][to - 1];
}

```

---

## 6. Testing & Results

### Test Case Example:

- Input: FROM = USD (1), TO = INR (4), Amount = 10
- Rate = 83.20
- Output: 10.00 (Currency 1) = 832.00 (Currency 4)

### Other test cases:

From	To	Amount	Output
USD	EUR	50	45.50
INR	USD	1000	12.00
AUD	GBP	10	5.30

Program successfully converts all supported currencies and handles invalid inputs like selecting same currency.

---

## 7. Conclusion & Future Work

### Conclusion

The Advanced Currency Converter is efficient, user-friendly, and demonstrates modular C programming practices. It is easy to maintain, extend, and suitable for beginners learning function-based design.

### Future Enhancements

- Real-time online currency rate fetching.
- GUI-based interface.
- Support for more currencies.
- Error-tolerant input handling.
- Adding file logging for conversion history.

---

## 8. References

- Programming in ANSI C – E. Balagurusamy
- Online C Documentation – [cplusplus.com](http://cplusplus.com)
- Currency rate data examples (static values used for project)
- Let us C

---

**End of Report**