

PROJECT NAME : Temprature converter

Student Name: Harsh

UID: 24bca10485

Branch:BCA

Section/Group:7-A

Semester: 1

Date of Performance:

Subject Name: Computer programming

Subject Code: 24CAH-101

1. Aim/Overview of the practical:

To developed a temperature converter in the C programming language.

1. Task to be done:

- . Set Up the Development Environment
- . Write the Program
- . Add Input Validation (Optional)
- .Test the Program
- .Document the Code
- .Reflect and Improve

1. Algorithm/Flowchart :

1. Start.

2. Display Options:

- Show the user the available conversion options (Celsius to Fahrenheit or Fahrenheit to Celsius).

3.Get User Input:

- Prompt the user to select an option (1 for Celsius to Fahrenheit, 2 for Fahrenheit to Celsius).

4. Read User Choice:

- Store the user's choice in a variable.

5. If User Chooses Option 1 (Celsius to Fahrenheit):

- a. Prompt the user to enter the temperature in Celsius.
- b. Read the temperature value.
- c. Calculate the temperature in Fahrenheit using the formula:

$$\text{Fahrenheit} = (\text{Celsius} \times \frac{9}{5}) + 32$$

$$\text{Fahrenheit} = (\text{Celsius} \times 9) / 5 + 32$$
- d. Display the result.

6. Else If User Chooses Option 2 (Fahrenheit to Celsius):

- a. Prompt the user to enter the temperature in Fahrenheit.
- b. Read the temperature value.
- c. Calculate the temperature in Celsius using the formula: $\text{Celsius} = (\text{Fahrenheit} - 32) \times \frac{5}{9}$

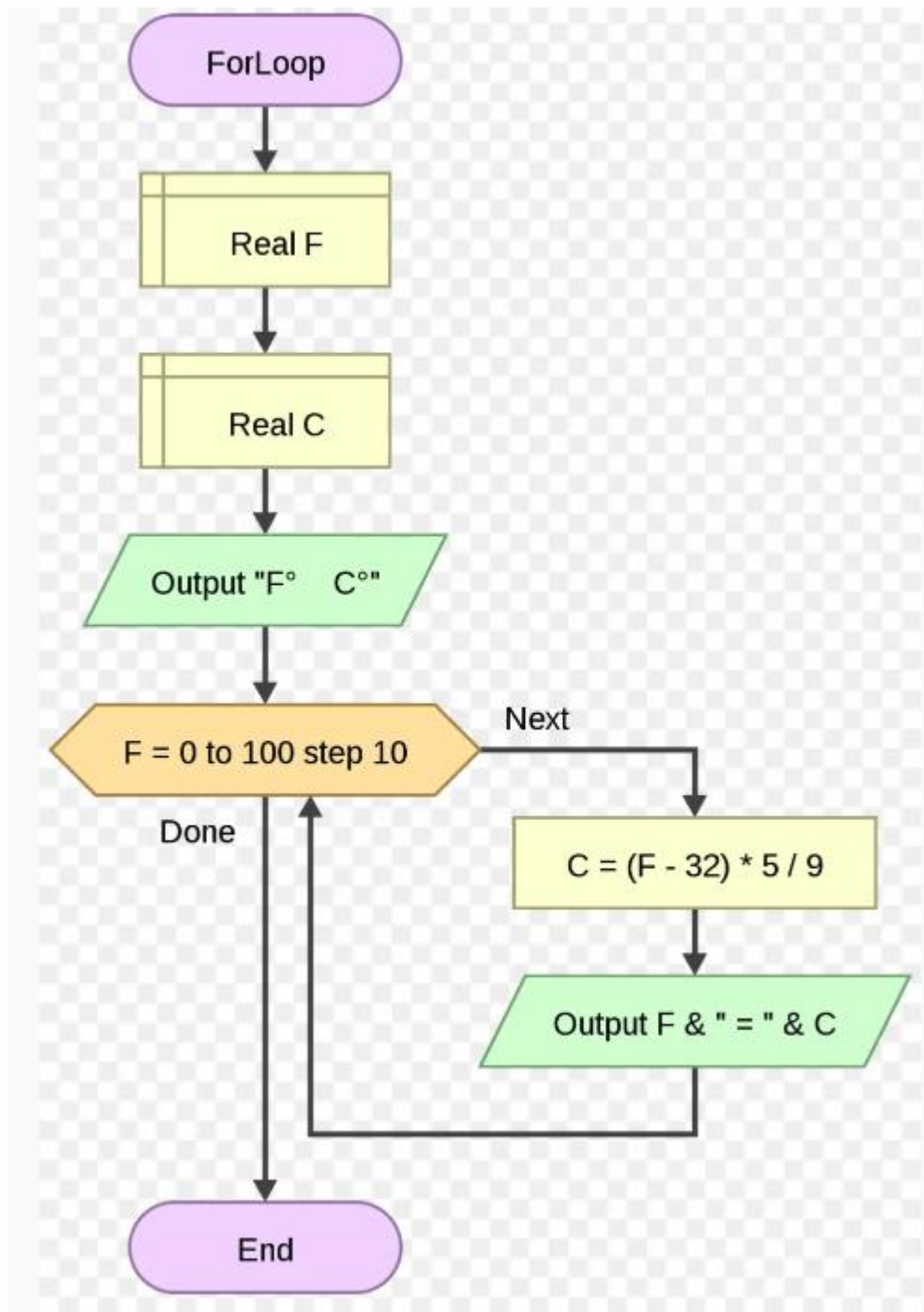
$$\text{Celsius} = (\text{Fahrenheit} - 32) \times 5 / 9$$
- d. Display the result.

7. Else (Invalid Choice):

- a. Display an error message indicating the choice is invalid.

8 End. .

FLOWCHART :



Code for experiment/practical:

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

```
int main() {
```

```
    float celsius, fahrenheit;
```

```
    int choice;
```

```

printf("Temperature Converter\n");

printf("1. Celsius to Fahrenheit\n");
printf("2. Fahrenheit to Celsius\n");
printf("Choose an option (1 or 2): ");

scanf("%d", &choice);


if (choice == 1) {

    printf("Enter temperature in Celsius: ");

    scanf("%f", &celsius);

    fahrenheit = (celsius * 9 / 5) + 32;

    printf("%.2f Celsius = %.2f Fahrenheit\n", celsius, fahrenheit);
} else if (choice == 2) {

    printf("Enter temperature in Fahrenheit: ");

    scanf("%f", &fahrenheit);

    celsius = (fahrenheit - 32) * 5 / 9;

    printf("%.2f Fahrenheit = %.2f Celsius\n", fahrenheit, celsius);
} else {

    printf("Invalid choice. Please select 1 or 2.\n");

}


return 0;

}

```

1. Result/Output/Writing Summary:

Output

[Clear](#)

```
/tmp/AhB6BE3VQH.o
```

```
Temperature Converter
```

```
1. Celsius to Fahrenheit
```

```
2. Fahrenheit to Celsius
```

```
Choose an option (1 or 2): 1
```

```
Enter temperature in Celsius: 78
```

```
78.00 Celsius = 172.40 Fahrenheit
```

```
=== Code Execution Successful ===
```

Writing Summary:

The temperature converter program in C is a straightforward application that enables users to convert temperatures between Celsius and Fahrenheit. Upon execution, the program prompts the user to select a conversion type: either converting Celsius to Fahrenheit or vice versa. After receiving the user's choice, it asks for the corresponding temperature input. Using the appropriate conversion formula—either $\text{Fahrenheit} = (\text{Celsius} \times \frac{9}{5}) + 32$ or $\text{Celsius} = (\text{Fahrenheit} - 32) \times \frac{5}{9}$ —the program calculates the converted temperature and displays the result. If the user inputs an invalid option, an error message is provided. This simple yet effective program showcases fundamental concepts such as user input, conditionals, and arithmetic operations in C, making it an excellent learning project for beginners.

Learning outcomes (What I have learnt):

1. **Understanding Basic Syntax:** Learners will become familiar with the basic syntax of C programming, including variable declaration, input/output functions, and control structures.
2. **Data Types and Operations:** Students will gain experience with different data types (e.g., float, int) and arithmetic operations, enhancing their ability to manipulate numerical data.
3. **User Input Handling:** The program will teach how to capture and process user input, which is crucial for interactive applications.
4. **Conditional Statements:** Learners will understand how to implement conditional logic using if-else statements, allowing the program to respond differently based on user choices.

5. **Mathematical Formulas:** Participants will apply mathematical formulas in programming, reinforcing the connection between math concepts and programming logic.
6. **Debugging Skills:** Through testing and refining their code, learners will develop problem-solving skills and gain experience in debugging common programming errors.
7. **Code Structure and Organization:** Students will learn the importance of structuring code clearly, making it easier to read, maintain, and expand in the future.
8. **Basic Output Formatting:** Participants will understand how to format output for better readability, including controlling decimal places.

Overall, this project serves as a practical introduction to programming in C, reinforcing foundational skills that are applicable to more complex programming tasks.

Evaluation Grid:

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Demonstration and Performance (Pre Lab Quiz)		5
2.	Worksheet		10
3.	Post Lab Quiz		5