**Topic :Unit converter**

**Team member – 1) Harsha Borkhade 0019**

**2) Swapnali Tanpure 0006**

**Step1 -Decomposition**

Decomposition, or breaking down a problem into smaller, more manageable parts, helps in understanding and solving complex problems. Here's a decomposition of the Unit Converter program:

1.User Interface:

Display the menu of conversion options.

Read and validate the user's choice.

Handle user input for length values.

2.Conversion Functions:

Implement the feetToMeters() function:

Accept a length value in feet as a parameter.

Multiply the length value by 0.3048 to convert it to meters. Return the converted length in meters.

Implement the metersToFeet() function:

Accept a length value in meters as a parameter.

Divide the length value by 0.3048 to convert it to feet.

Return the converted length in feet.

**Step 2 - pattern recognition**

Pattern recognition in the decomposition of the Unit Converter program can be as follows:

1.User Interface:

Displaying a menu of options to the user and reading their input.

Recognizing and validating the user's choice to ensure it falls within the acceptable range of options.

Handling user input for length values, which may involve error checking and conversion from string input to numeric values.

2.Conversion Functions:

Identifying the common pattern of converting units from one form to another (feet to meters or meters to feet).

Implementing separate functions for each conversion type to encapsulate the logic and ensure modularity and reusability.

3.Error Handling:Recognizing and handling potential errors in user input, such as non-numeric values, out-of-range choices, or division by zero (for example, if the user enters 0 as the length value for meters to feet conversion).

Providing appropriate error messages to guide the user in correcting their input or understanding the limitations of the program.

4.Main Program Flow:

Employing control structures, such as conditionals (if-else or switch statements), to direct the flow of the program based on the user's choice.

The interaction between different components, such as the user interface, conversion functions, and error handling, to ensure a smooth and cohesive execution of the program.

**Step 3 - Abstraction**

Abstraction in the context of the Unit Converter program refers to the process of simplifying and representing complex details by focusing on essential features. By applying procedural abstraction, each step is separated and encapsulated into its own procedure, allowing for modular and reusable code.Abstraction for each step:

1.User Interface:

Abstract the user interface as a module responsible for displaying the menu, reading user input, and providing an interface for obtaining the conversion choice and length value.

1. Conversion Functions:

Abstract the conversion functions as modules that encapsulate the logic for converting units. Each function abstracts the specific conversion formula and provides a clean interface to convert between feet and meters.

3.Error Handling:

Abstract the error handling module as a component that recognizes and handles errors related to user input. It provides abstraction for validating input and generating appropriate error messages.

4.Main Program Flow:

Abstract the main program flow as a module responsible for the interactions between the user interface, conversion functions, and error handling. It provides the high-level control flow and coordinates the execution of the program.

By abstracting these components, the complexities of the program are hidden, allowing developers to focus on their individual responsibilities.

**Step 4 - Algorithm**

Algorithm for converting length into feet:

1.Start.

2.Read the length value to be converted.

3.Read the unit of the input length .

4.Determine the conversion based on the unit of the input length.

5.Customize the conversion factor according to the specific unit being used.

6.Multiply the length value by the conversion factor to obtain the equivalent length in feet.

7.Display the converted length in feet.

8.End.