Loops in java **18 April 2024**

1. ***For Loop***

```
For ( int i=0; i<n; i++) {
// code to be executed repeatedly
}
```

Scenario: Prime Number Generator with Sieve of Eratosthenes

Description: You are given a task to implement a program that generates prime numbers up to a specified limit using the *Sieve of Eratosthenes algorithm*. The program should prompt the user to enter the upper limit for prime number generation and then display all prime numbers up to that limit.

Requirements:

- Prompt the user to enter an upper limit N for prime number generation.
- Implement the Sieve of Eratosthenes algorithm to generate all prime numbers up to N.
- Use a Boolean array to mark all numbers as prime or composite.
- Iterate through the array using a for loop to mark multiples of each prime number as composite.
- Display all prime numbers found up to the specified limit N.

Challenges:

- Efficiently implement the Sieve of Eratosthenes algorithm to handle large values of N.
- Ensure that the program handles user input validation to prevent invalid inputs.
- Optimize the program for performance and memory usage, especially for larger values of N.

1. **While Loop**

This loop is used when the number of iterations is not known before hand, and it continues as long as the specified condition is true.

```
int i=0; while
(i<n) { i++; }
```

Scenario: Temperature Converter

Description: You are task to develop a program to convert temperatures between

Celsius and Fahrenheit scales. The program should allow the user to input a temperature in either Celsius or Fahrenheit and then convert it to the other scale. The conversion formulas are as follows.

- To convert Celsius to Fahrenheit: $F = 9/5 \times C + 32$
- To convert Fahrenheit to Celsius: $C = 5/9 \times (F - 32)$

Requirements:

- Prompt the user to choose the temperature scale they want to convert from (Celsius or Fahrenheit).
- Prompt the user to enter the temperature value in the chosen scale.
- Use a while loop to repeatedly prompt the user for input until they choose to exit the program.
- Convert the input temperature to the other scale using the appropriate conversion formula.
- Display the converted temperature to the user.
- Allow the user to choose whether to convert another temperature or exit the program.

Challenges:

- Handle invalid input from the user, such as non-numeric values or incorrect choices.
- Implement a user-friendly interface with clear prompts and feedback messages.
- Ensure that the program runs efficiently and handles conversions accurately for a wide range of input values.