**Project Documentation Report**

**Project Title**

**Unit Converter (Java - CLI Project using OOP)**

**Introduction**

The **Unit Converter** is a beginner-friendly Java project designed to perform conversions between **Weight, Length, and Temperature units**. The project is implemented using **Object-Oriented Programming (OOP)** principles such as **classes, methods, and inheritance**.

This project demonstrates how OOP can be applied in real-world applications while keeping the design **simple, modular, and understandable**.

**Objectives**

* To practice **basic OOP concepts** in Java.
* To implement **separate methods** for each conversion to increase readability.
* To provide a **Command-Line Interface (CLI)** application for unit conversions.
* To help beginners understand how to **structure a Java project** with multiple classes.

**Features**

* Supports multiple conversions in **Weight, Length, and Temperature** categories.
* Each category is placed in a **separate class**.
* Each unit conversion has its **own method** for clarity.
* Easy to extend: new conversions can be added by creating additional methods.
* CLI menu allows interactive user input.

**Project Structure**

UnitConverterProject/

│

├── Main.java # Main entry point (menu + user interaction)

├── WeightConverter.java # Contains all weight conversion methods

├── LengthConverter.java # Contains all length conversion methods

├── TemperatureConverter.java # Contains all temperature conversion methods

└── README.md # Project documentation

**Tools and Technologies**

* **Programming Language**: Java
* **Paradigm**: Object-Oriented Programming (OOP)
* **Environment**: JDK 8+
* **IDE**: Any Java IDE (IntelliJ, Eclipse or VS Code)

**How it Works (Flow)**

1. The program starts with **Main.java**, displaying a **menu**.
2. The user selects a category (Weight, Length, Temperature).
3. The appropriate class (WeightConverter, LengthConverter, or TemperatureConverter) is used.
4. Within the class, the **specific conversion method** is called.
5. The result is displayed on the screen.
6. The program loops back to the menu until the user exits.

**Example Methods**

* convertKilogramToGram(double kg)
* convertGramToMilligram(double g)
* convertMeterToCentimeter(double m)
* convertCelsiusToFahrenheit(double c)

**Limitations**

* Only works in **command-line** (no GUI).
* Input validation is minimal (assumes valid numeric input).
* Does not yet support every possible unit of measure.

**Conclusion**

This project is a **practical example of OOP in action**. It not only strengthens programming logic but also provides a hands-on approach to building **modular and reusable Java applications**. Beginners can extend this system by adding new units, improving the menu, or even creating a **GUI version**.