

A Minor Project Synopsis on

# **Currency Converter**

Submitted to Manipal University, Jaipur

Towards the partial fulfillment for the Award of the Degree of

**BACHELOR OF TECHNOLOGY**

**In Computers Science and Engineering**

**2022-2023**

By

1. Molik Jain (Regn No.:- 229301105 )
2. Tathagata Ghosh(Regn No.:- 229301664)



**MANIPAL UNIVERSITY  
JAIPUR**

Under the guidance of

Dr. Gireesh Kumar

Mr. Mayank Kumar Jain

**Department of Computer Science and Engineering**

**School of Computer Science and Engineering**

**Manipal University Jaipur**

**Dehmi Kalan, Jaipur-Ajmer Expressway,**

**Jaipur-303007, Rajasthan , India**

## **Introduction :-**

A currency converter is a software application that allows users to convert one currency to another, using the latest exchange rates. Currency converters are useful for a variety of purposes, such as:

***International travel:*** Currency converters can help travelers to determine how much money they need to exchange before they travel, and to budget their money while they are traveling.

***International business:*** Currency converters can help businesses to determine the cost of goods and services from other countries, and to set prices for their own goods and services in multiple currencies.

***Personal finance:*** Currency converters can help individuals to track the value of their investments in foreign currencies, and to compare the cost of products and services from different countries.

The Currency Converter project is a console-based Java application that allows users to convert between a variety of currencies using the latest exchange rates. The project is designed to be easy to use and navigate, and it is well-documented and easy to maintain.

## **Project Requirements :-**

The Currency Converter project has the following requirements:

The project must be a console-based application.

The project must allow users to convert between a variety of currencies using the latest exchange rates.

The project must be easy to use and navigate.

The project must be well-documented and easy to maintain.

## Project Design

The Currency Converter project is designed using the following components:

*Main class:* The main class controls the overall flow of the application. It is responsible for displaying the application's main menu, handling user input, and calling the other components of the application to perform the currency conversion.

*Currency class:* The currency class represents a currency. It has fields for the currency's name, symbol, and exchange rate. It also has methods for converting the currency to another currency.

*Currency converter class:* The currency converter class performs the currency conversion. It takes the currency to be converted and the currency to be converted to as input, and it returns the converted amount as output.

*User interface class:* The user interface class displays the application's user interface. It has methods for displaying the application's main menu, getting user input, and displaying the results of the currency conversion.

*Integration:* The components of the application will be integrated together using the following steps:

- The main class will create instances of the currency class, currency converter class, and user interface class.
- The main class will then call the user interface class to display the application's main menu.
- The main class will then handle user input and call the appropriate methods of the currency converter class to perform the currency conversion.
- The main class will then call the user interface class to display the results of the currency conversion.

## Project Objective :

The objective of this Java console-based currency converter project is to develop a simple and easy-to-use tool that allows users to convert between a set of supported currencies. The project will be designed to be modular and extensible, making it easy to add new features and support for additional currencies in the future.

## Benefits to Users :

The currency converter will be a valuable tool for a variety of users, including:

- Travelers who need to convert money between different currencies when visiting foreign countries.
- Businesses that conduct international trade or have customers in multiple countries.
- Individuals who need to track the value of their investments in foreign currencies.
- Anyone who needs to convert money between different currencies for any reason.

The currency converter will provide users with the following benefits:

- *Convenience:* The converter can be used from anywhere, without the need for an internet connection.
- *Accuracy:* The converter uses real-time exchange rates to provide users with the most accurate conversion results possible.
- *Flexibility:* The converter allows users to convert between a variety of currencies, including major and minor currencies.
- *Simplicity:* The converter is easy to use and understand, even for beginners.

## Project Scope :

The project will focus on developing a simple and easy-to-use currency converter that supports a set of common currencies. The converter will be implemented using the Java programming language and will be console-based.

## Implementation Plan :

The project will be implemented using the following steps:

1. *Requirements gathering*: The first step will be to gather requirements from potential users to understand their needs and expectations.
2. *Design*: Once the requirements have been gathered, the project will be designed using a simple object-oriented approach.
3. *Implementation*: The converter will be implemented using the Java programming language.
4. *Testing*: The converter will be thoroughly tested to ensure that it works correctly and provides accurate results.
5. *Deployment*: The converter will be deployed as a console application that can be used on any platform that supports Java.

## Development Code :-

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

abstract class CurrencyConverter {
    private double conversionRate;

    public CurrencyConverter(double rate) {
        this.conversionRate = rate;
    }

    public abstract double convertTo(double amount);

    public abstract double convertFrom(double amount);
}
```

```

        public double getConversionRate() {
            return conversionRate;
        }
    }

class INR_USD_CurrencyConverter extends CurrencyConverter {
    public INR_USD_CurrencyConverter(double rate) {
        super(rate);
    }

    @Override
    public double convertTo(double amount) {
        return amount * getConversionRate();
    }

    @Override
    public double convertFrom(double amount) {
        return amount / getConversionRate();
    }
}

class USD_INR_CurrencyConverter extends CurrencyConverter {
    public USD_INR_CurrencyConverter(double rate) {
        super(rate);
    }

    @Override
    public double convertTo(double amount) {
        return amount * getConversionRate();
    }

    @Override
    public double convertFrom(double amount) {
        return amount / getConversionRate();
    }
}

class INR_GBP_CurrencyConverter extends CurrencyConverter {
    public INR_GBP_CurrencyConverter(double rate) {
        super(rate);
    }

    @Override
    public double convertTo(double amount) {
        return amount * getConversionRate();
    }

    @Override

```

```

        public double convertFrom(double amount) {
            return amount / getConversionRate();
        }
    }

class GBP_INR_CurrencyConverter extends CurrencyConverter {
    public GBP_INR_CurrencyConverter(double rate) {
        super(rate);
    }

    @Override
    public double convertTo(double amount) {
        return amount * getConversionRate();
    }

    @Override
    public double convertFrom(double amount) {
        return amount / getConversionRate();
    }
}

class INR_EURO_CurrencyConverter extends CurrencyConverter {
    public INR_EURO_CurrencyConverter(double rate) {
        super(rate);
    }

    @Override
    public double convertTo(double amount) {
        return amount * getConversionRate();
    }

    @Override
    public double convertFrom(double amount) {
        return amount / getConversionRate();
    }
}

class EURO_INR_CurrencyConverter extends CurrencyConverter {
    public EURO_INR_CurrencyConverter(double rate) {
        super(rate);
    }

    @Override
    public double convertTo(double amount) {
        return amount * getConversionRate();
    }

    @Override

```

```

        public double convertFrom(double amount) {
            return amount / getConversionRate();
        }
    }

class INR_JPY_CurrencyConverter extends CurrencyConverter {
    public INR_JPY_CurrencyConverter(double rate) {
        super(rate);
    }

    @Override
    public double convertTo(double amount) {
        return amount * getConversionRate();
    }

    @Override
    public double convertFrom(double amount) {
        return amount / getConversionRate();
    }
}

class JPY_INR_CurrencyConverter extends CurrencyConverter {
    public JPY_INR_CurrencyConverter(double rate) {
        super(rate);
    }

    @Override
    public double convertTo(double amount) {
        return amount * getConversionRate();
    }

    @Override
    public double convertFrom(double amount) {
        return amount / getConversionRate();
    }
}

class USD_GBP_CurrencyConverter extends CurrencyConverter {
    public USD_GBP_CurrencyConverter(double rate) {
        super(rate);
    }

    @Override
    public double convertTo(double amount) {
        return amount * getConversionRate();
    }

    @Override

```



```

        public double convertFrom(double amount) {
            return amount / getConversionRate();
        }
    }

    class GBP_USD_CurrencyConverter extends CurrencyConverter {
        public GBP_USD_CurrencyConverter(double rate) {
            super(rate);
        }

        @Override
        public double convertTo(double amount) {
            return amount * getConversionRate();
        }

        @Override
        public double convertFrom(double amount) {
            return amount / getConversionRate();
        }
    }

    class USD_EURO_CurrencyConverter extends CurrencyConverter {
        public USD_EURO_CurrencyConverter(double rate) {
            super(rate);
        }

        @Override
        public double convertTo(double amount) {
            return amount * getConversionRate();
        }

        @Override
        public double convertFrom(double amount) {
            return amount / getConversionRate();
        }
    }

    class EURO_USD_CurrencyConverter extends CurrencyConverter {
        public EURO_USD_CurrencyConverter(double rate) {
            super(rate);
        }

        @Override
        public double convertTo(double amount) {
            return amount * getConversionRate();
        }

        @Override

```

```

        public double convertFrom(double amount) {
            return amount / getConversionRate();
        }
    }

class USD_JPY_CurrencyConverter extends CurrencyConverter {
    public USD_JPY_CurrencyConverter(double rate) {
        super(rate);
    }

    @Override
    public double convertTo(double amount) {
        return amount * getConversionRate();
    }

    @Override
    public double convertFrom(double amount) {
        return amount / getConversionRate();
    }
}

class JPY_USD_CurrencyConverter extends CurrencyConverter {
    public JPY_USD_CurrencyConverter(double rate) {
        super(rate);
    }

    @Override
    public double convertTo(double amount) {
        return amount * getConversionRate();
    }

    @Override
    public double convertFrom(double amount) {
        return amount / getConversionRate();
    }
}

class EURO_GBP_CurrencyConverter extends CurrencyConverter {
    public EURO_GBP_CurrencyConverter(double rate) {
        super(rate);
    }

    @Override
    public double convertTo(double amount) {
        return amount * getConversionRate();
    }

    @Override

```

```

        public double convertFrom(double amount) {
            return amount / getConversionRate();
        }
    }

    class GBP_EURO_CurrencyConverter extends CurrencyConverter {
        public GBP_EURO_CurrencyConverter(double rate) {
            super(rate);
        }

        @Override
        public double convertTo(double amount) {
            return amount * getConversionRate();
        }

        @Override
        public double convertFrom(double amount) {
            return amount / getConversionRate();
        }
    }

    class EURO_JPY_CurrencyConverter extends CurrencyConverter {
        public EURO_JPY_CurrencyConverter(double rate) {
            super(rate);
        }

        @Override
        public double convertTo(double amount) {
            return amount * getConversionRate();
        }

        @Override
        public double convertFrom(double amount) {
            return amount / getConversionRate();
        }
    }

    class JPY_EURO_CurrencyConverter extends CurrencyConverter {
        public JPY_EURO_CurrencyConverter(double rate) {
            super(rate);
        }

        @Override
        public double convertTo(double amount) {
            return amount * getConversionRate();
        }

        @Override

```

```

        public double convertFrom(double amount) {
            return amount / getConversionRate();
        }
    }

class GBP_JPY_CurrencyConverter extends CurrencyConverter {
    public GBP_JPY_CurrencyConverter(double rate) {
        super(rate);
    }

    @Override
    public double convertTo(double amount) {
        return amount * getConversionRate();
    }

    @Override
    public double convertFrom(double amount) {
        return amount / getConversionRate();
    }
}

class JPY_GBP_CurrencyConverter extends CurrencyConverter {
    public JPY_GBP_CurrencyConverter(double rate) {
        super(rate);
    }

    @Override
    public double convertTo(double amount) {
        return amount * getConversionRate();
    }

    @Override
    public double convertFrom(double amount) {
        return amount / getConversionRate();
    }
}

public class Main {
    public static void main(String[] args) {
        try {
            // Read the conversion rates from files

            //INR conversions both ways
            double inr_usd_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/inr_usd_conversion_rate.txt");
            double usd_inr_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/usd_inr_conversion_rate.txt");
            double inr_gbp_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/inr_gbp_conversion_rate.txt");

```

```

        double gbp_inr_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/gbp_inr_conversion_rate.txt");
        double inr_euro_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/inr_euro_conversion_rate.txt");
        double euro_inr_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/euro_inr_conversion_rate.txt");
        double inr_jpy_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/inr_jpy_conversion_rate.txt");
        double jpy_inr_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/jpy_inr_conversion_rate.txt");

        //USD conversions both ways
        double usd_gbp_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/usd_gbp_conversion_rate.txt");
        double gbp_usd_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/gbp_usd_conversion_rate.txt");
        double usd_euro_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/usd_euro_conversion_rate.txt");
        double euro_usd_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/euro_usd_conversion_rate.txt");
        double usd_jpy_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/usd_jpy_conversion_rate.txt");
        double jpy_usd_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/jpy_usd_conversion_rate.txt");

        //EURO conversions both ways
        double euro_gbp_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/euro_gbp_conversion_rate.txt");
        double gbp_euro_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/gbp_euro_conversion_rate.txt");
        double euro_jpy_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/euro_jpy_conversion_rate.txt");
        double jpy_euro_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/jpy_euro_conversion_rate.txt");

        //GBP conversions both ways
        double gbp_jpy_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/gbp_jpy_conversion_rate.txt");
        double jpy_gbp_rate = readConversionRate("C:/Users/TATHAGATA
GHOSH/Desktop/Currency_converter_2/src/jpy_gbp_conversion_rate.txt");

        CurrencyConverter inr_usd_Converter = new
INR_USD_CurrencyConverter(inr_usd_rate);
        CurrencyConverter usd_inr_Converter = new
USD_INR_CurrencyConverter(usd_inr_rate);
        CurrencyConverter inr_gbp_Converter = new
INR_GBP_CurrencyConverter(inr_gbp_rate);

```

```

        CurrencyConverter gbp_inr_Converter = new
GBP_INR_CurrencyConverter(gbp_inr_rate);
        CurrencyConverter inr_euro_Converter = new
INR_EURO_CurrencyConverter(inr_euro_rate);
        CurrencyConverter euro_inr_Converter = new
EURO_INR_CurrencyConverter(euro_inr_rate);
        CurrencyConverter inr_jpy_Converter = new
INR_JPY_CurrencyConverter(inr_jpy_rate);
        CurrencyConverter jpy_inr_Converter = new
JPY_INR_CurrencyConverter(jpy_inr_rate);
        CurrencyConverter usd_gbp_Converter = new
USD_GBP_CurrencyConverter(usd_gbp_rate);
        CurrencyConverter gbp_usd_Converter = new
GBP_USD_CurrencyConverter(gbp_usd_rate);
        CurrencyConverter usd_euro_Converter = new
USD_EURO_CurrencyConverter(usd_euro_rate);
        CurrencyConverter euro_usd_Converter = new
EURO_USD_CurrencyConverter(euro_usd_rate);
        CurrencyConverter usd_jpy_Converter = new
USD_JPY_CurrencyConverter(usd_jpy_rate);
        CurrencyConverter jpy_usd_Converter = new
JPY_USD_CurrencyConverter(jpy_usd_rate);
        CurrencyConverter euro_gbp_Converter = new
EURO_GBP_CurrencyConverter(euro_gbp_rate);
        CurrencyConverter gbp_euro_Converter = new
GBP_EURO_CurrencyConverter(gbp_euro_rate);
        CurrencyConverter euro_jpy_Converter = new
EURO_JPY_CurrencyConverter(euro_jpy_rate);
        CurrencyConverter jpy_euro_Converter = new
JPY_EURO_CurrencyConverter(jpy_euro_rate);
        CurrencyConverter gbp_jpy_Converter = new
GBP_JPY_CurrencyConverter(gbp_jpy_rate);
        CurrencyConverter jpy_gbp_Converter = new
JPY_GBP_CurrencyConverter(jpy_gbp_rate);

        // Get user input
        System.out.println();
        System.out.println("#####__JAVA CURRENCY
CONVERTER__#####");
        System.out.println();
        double amountToConvert = getUserInput("Enter the amount: ");
        int fromCurrencyChoice = getUserCurrencyChoice("Select the source
currency: ");
        int toCurrencyChoice = getUserCurrencyChoice("Select the
destination currency: ");

        double convertedAmount = 0.0;
        String fromCurrency = "";

```

```

String toCurrency = "";

switch (fromCurrencyChoice) {
    case 1:
        fromCurrency = "INR";
        break;
    case 2:
        fromCurrency = "USD";
        break;
    case 3:
        fromCurrency = "Euro";
        break;
    case 4:
        fromCurrency = "GBP";
        break;
    case 5:
        fromCurrency = "JPY";
        break;
    default:
        System.err.println("Invalid source currency choice.");
        return;
}

switch (toCurrencyChoice) {
    case 1:
        toCurrency = "INR";
        break;
    case 2:
        toCurrency = "USD";
        break;
    case 3:
        toCurrency = "Euro";
        break;
    case 4:
        toCurrency = "GBP";
        break;
    case 5:
        toCurrency = "JPY";
        break;
    default:
        System.err.println("Invalid destination currency
choice.");
        return;
}

if (fromCurrencyChoice == 1 && toCurrencyChoice == 2) {
    convertedAmount =
inr_usd_Converter.convertTo(amountToConvert);

```

```

        } else if (fromCurrencyChoice == 2 && toCurrencyChoice == 1) {
            convertedAmount =
            usd_inr_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 1 && toCurrencyChoice == 3) {
            convertedAmount =
            inr_euro_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 3 && toCurrencyChoice == 1) {
            convertedAmount =
            euro_inr_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 1 && toCurrencyChoice == 4) {
            convertedAmount =
            inr_gbp_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 4 && toCurrencyChoice == 1) {
            convertedAmount =
            gbp_inr_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 1 && toCurrencyChoice == 5) {
            convertedAmount =
            inr_jpy_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 5 && toCurrencyChoice == 1) {
            convertedAmount =
            jpy_inr_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 2 && toCurrencyChoice == 3) {
            convertedAmount =
            usd_euro_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 3 && toCurrencyChoice == 2) {
            convertedAmount =
            euro_usd_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 2 && toCurrencyChoice == 4) {
            convertedAmount =
            usd_gbp_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 4 && toCurrencyChoice == 2) {
            convertedAmount =
            gbp_usd_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 2 && toCurrencyChoice == 5) {
            convertedAmount =
            usd_jpy_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 5 && toCurrencyChoice == 2) {
            convertedAmount =
            jpy_usd_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 3 && toCurrencyChoice == 4) {
            convertedAmount =
            euro_gbp_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 4 && toCurrencyChoice == 3) {
            convertedAmount =
            gbp_euro_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 3 && toCurrencyChoice == 5) {
            convertedAmount =
            euro_jpy_Converter.convertTo(amountToConvert);

```



```

        } else if (fromCurrencyChoice == 5 && toCurrencyChoice == 3) {
            convertedAmount =
jpy_euro_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 4 && toCurrencyChoice == 5) {
            convertedAmount =
gbp_jpy_Converter.convertTo(amountToConvert);
        } else if (fromCurrencyChoice == 5 && toCurrencyChoice == 4) {
            convertedAmount =
jpy_gbp_Converter.convertTo(amountToConvert);
        }

```

```

        System.out.printf("%.2f %s is equivalent to %.2f %s.\n",
amountToConvert, fromCurrency, convertedAmount, toCurrency);
    } catch (IOException | NumberFormatException e) {
        System.err.println("Error: Unable to read conversion rates from
files or invalid rates.");
    }
}

```

```

private static double getUserInput(String prompt) {
    System.out.print(prompt);
    try {
        BufferedReader reader = new BufferedReader(new
java.io.InputStreamReader(System.in));
        return Double.parseDouble(reader.readLine());
    } catch (IOException | NumberFormatException e) {
        System.err.println("Invalid input. Please enter a valid number.");
        return getUserInput(prompt);
    }
}

```

```

private static int getUserCurrencyChoice(String prompt) {
    System.out.println(prompt);
    System.out.println("1. INR");
    System.out.println("2. USD");
    System.out.println("3. Euro");
    System.out.println("4. GBP");
    System.out.println("5. JPY");
    System.out.print("Enter your choice: ");

    try {
        BufferedReader reader = new BufferedReader(new
java.io.InputStreamReader(System.in));
        return Integer.parseInt(reader.readLine());
    } catch (IOException | NumberFormatException e) {
        System.err.println("Invalid choice. Please enter a valid
number.");
    }
}

```

```

        return getUserCurrencyChoice(prompt);
    }
}

private static double readConversionRate(String fileName) throws
IOException {
    BufferedReader reader = new BufferedReader(new FileReader(fileName));
    double rate = Double.parseDouble(reader.readLine());
    reader.close();
    return rate;
}
}

```

```
#####__JAVA CURRENCY CONVERTER__#####
```

Enter the amount: 1000.00

Select the source currency:

1. INR
2. USD
3. Euro
4. GBP
5. JPY

Enter your choice: 1

Select the destination currency:

1. INR
2. USD
3. Euro
4. GBP
5. JPY

Enter your choice: 2

1000.00 INR is equivalent to 12.00 USD.

```
#####_JAVA CURRENCY CONVERTER_#####
```

Enter the amount: 12.00

Select the source currency:

1. INR
2. USD
3. Euro
4. GBP
5. JPY

Enter your choice: 2

Select the destination currency:

1. INR
2. USD
3. Euro
4. GBP
5. JPY

Enter your choice: 1

12.00 USD is equivalent to 999.36 INR.

```
#####_JAVA CURRENCY CONVERTER_#####
```

Enter the amount: 568.12

Select the source currency:

1. INR
2. USD
3. Euro
4. GBP
5. JPY

Enter your choice: 5

Select the destination currency:

1. INR
2. USD
3. Euro
4. GBP
5. JPY

Enter your choice: 3

568.12 JPY is equivalent to 3.52 Euro.

```
#####_JAVA CURRENCY CONVERTER_#####
```

Enter the amount: 6598.34

Select the source currency:

1. INR
2. USD
3. Euro
4. GBP
5. JPY

Enter your choice: 3

Select the destination currency:

1. INR
2. USD
3. Euro
4. GBP
5. JPY

Enter your choice: 5

6598.34 Euro is equivalent to 1057647.92 JPY.

## **Conclusion :-**

This project developed a simple but useful currency converter application using Java. The project used inheritance, encapsulation, polymorphism, file handling, and interfaces to implement the following features:

Inheritance was used to create a hierarchy of currency classes, with each class representing a different type of currency.

Encapsulation was used to hide the implementation details of the currency classes and to protect the data from being modified directly.

Polymorphism was used to allow the currency converter to work with any type of currency, without having to modify the code.

File handling was used to load the exchange rates from a file.

Interfaces were used to define the contract for the currency converter service.

The currency converter application can be used to convert between a variety of currencies, including major and minor currencies. It is a valuable tool for travelers, businesses, and individuals who need to track the value of their investments in foreign currencies.

In addition to the features listed above, the currency converter application can be extended to add more features, such as:

Support for more currencies

The ability to save user-defined conversions

Overall, this project was a valuable learning experience that demonstrated the power of Java and its object-oriented programming features.

-----X-----