Development and Implementation of a Capital Gains Tax Calculation Program in Java

A Report on SENG1110 Assignment 1

The University of Newcastle

SCHOOL OF INFORMATION AND PHYSICAL SCIENCES

SENG1110 - Object-Oriented Programming

Authors: Min Thu Khaing, Thet Paing Hmu

Instructor: Dr. Poh Kok Loo

Submission Date : 24/02/2024

Table of Contents

Abstract

This report details the development and implementation and testing of a Java program designed to calculate Capital Gains Tax (CGT) and profits from cryptocurrency transactions, created for the SENG1110 Programming Assignment 1 at the University of Newcastle, Trimester 1, 2024. The program is structured into three classes: CgtInterface, Investment, and User. CgtInterface acts as the main entry point, facilitating user interaction, data validation, and result display. Investment manages investment details over three years and predicts profits, while User represents the investor and handles CGT calculation. Key functionalities include user input gathering, accurate CGT and profit calculation, and error handling mechanisms which are done by comparing the input value against the specified regular expression. The report summarizes extensive testing covering valid inputs, boundary cases, and error scenarios, affirming the program's accuracy and reliability.

The program underwent rigorous testing, covering various scenarios to ensure accuracy and reliability. Initial testing focused on valid inputs, including user details and transaction information, which the program successfully processed to calculate CGT percentages, profits after taxation, and future profit predictions. Extensive testing also addressed boundary cases and error handling, verifying the program's ability to identify and handle invalid inputs effectively, with appropriate error messages. Additionally, the handling of investment prompts and input validation for investment amounts over three years was thoroughly tested to ensure accurate profit calculation and prediction.

Overall, the testing results demonstrate the program's robustness in handling various input scenarios while accurately calculating CGT and profit for cryptocurrency transactions. For further details on specific test cases and outcomes, the full report is recommended.

Introduction

We present CGT calculation program, a Java program designed to address two primary functions: calculating Capital Gains Tax (CGT) on cryptocurrency investments and predicting future profits based on selected coins. It begins by gathering user information including name, salary, residency status, cryptocurrency buying and selling prices, and the duration of investment. Leveraging this information, the program proceeds to compute CGT percentages and the actual profit after taxation.

Furthermore, the program undertakes a secondary task of predicting profits for the subsequent three years, drawing upon the user's initial investment and yearly contributions. The program aims to offer users detailed insights into their cryptocurrency investments.

Design and Implementation

The program is designed using object-oriented principles, with a focus on modularity, and scalability, and is structured with three key classes:

1. **CgtInterface** - The CgtInterface class serves as the main entry point for the Capital Gains Tax (CGT) calculation program. It interacts with the user to collect necessary information, performs calculations, and displays results.

   The run method is the core of this class. It orchestrates the flow of the program, from asking user details and income, calculating CGT, asking the user if they want to invest, continuing with investment details if the user wishes to invest, calculating investment, and finally printing the user data.

2. **Investment** - The Investment class represents an investment that the user plans to make over three years. It includes information about the deposits made each year and methods for calculating predicted profits based on the selected coin and deposits.

   This class allows setting the selected coin for investment, retrieving, and setting deposits for each year, calculating predicted profits, and getting predicted yearly and total profits for each year.

3. **User** - The User class represents a user who is making an investment and provides methods to manage user data and calculate capital gains tax (CGT) based on user inputs.

   This class includes information such as the user's name, annual salary, residency status, investment details, tax rate, CGT, and actual profit after deducting CGT.

Functionality

The program prompts users to input relevant data such as name, annual salary, cryptocurrency transaction details, and investment preferences.

Input validation ensures that only appropriate values are accepted, enhancing the reliability of calculations.

Calculation methods accurately determine CGT liabilities and projected profits based on user inputs.

The program displays clear and structured outputs, providing users with actionable insights into their cryptocurrency transactions and potential investments.

Error Handling

Error checking and input validation are performed in the class CgtInterface, within the methods getValidatedNumInput() for double data type, and getValidatedInput() for Strings. This is achieved by comparing the input value against the specified regular expression.

getValidatedNumInput() - Retrieves and validates numerical input from the user and returns the validated numerical input from the user. And it takes the following parameters.

1. minimum - The minimum allowed value for the input.
2. maximum - The maximum allowed value for the input which can be canceled out by passing the value "-1".
3. acceptEqualToMinimum - Boolean value whether going to accept equal amount to minimum amount.
4. prompt - The message prompting the user for input.
5. console - The Scanner object used for input.
6. regex - The regular expression used to validate the input.
7. invalidMessage - The message displayed when the input is invalid.

getValidatedInput() - Retrieves the String input from console and validates it using given regular expression and return the value if successful. And it takes the following parameters.

1. prompt - The message prompting the user for input.
2. console - The Scanner object used for input.
3. isName - Boolean value whether the function should expect a name or not.
4. regex - The regular expression used to validate the input.
5. invalidMessage - The message displayed when the input does not match the regular expression.

(Refer to this section to view the code implementation.)

Calculation

Calculation of Capital Gains Tax (CGT) on Cryptocurrency Transactions

The program undertakes calculations to determine the Capital Gains Tax (CGT) liability based on cryptocurrency transactions using user inputs and income information. The process involves several steps and formulas detailed below.

Definitions:

Let:

- $P_{selling}$ be the Selling Price of the cryptocurrency.
- P_{buying} be the Buying Price of the cryptocurrency.
- P_{crypto} be the Profit derived from the cryptocurrency transaction.
- T_{year} be the number of years the cryptocurrency is held.
- P_{cgt} be the Profit for CGT.
- I_{annual} be the Total Annual Income.
- S_{annual} be the Annual Salary.

- τ be the tax rate, which can be determined by the function $f(I_{annual})$. f is a function determining the tax rate based on total annual income.
- CGT be the Capital Gains Tax.
- P_{actual} be the Actual Profit.

Formula:

1. Calculate Profit(P_{crypto}):
$$P_{crypto} = P_{selling} - P_{buying}$$
2. Calculate Profit for CGT(P_{cgt}):
$$P_{cgt} = \frac{P}{T_{years}}$$
3. Calculate Total Annual Income(I_{annual}):
$$I_{annual} = S_{annual} + P_{cgt}$$
4. Calculate Capital Gains Tax(CGT):
$$CGT = \tau \times P_{cgt}$$
5. Calculate Actual Profit(P_{actual}):
$$P_{actual} = P_{cgt} - CGT$$

(Refer to this section for code implementation.)

Tax rates for residents and non-residents are applied based on taxable income brackets, as detailed in the provided tax rate tables.

Calculation:

For example, let's consider the following scenario:

P_{buying} = 5,000

$P_{selling}$ = 9,000

T_{year} = 4

S_{annual} = 340,000 (Annual Salary)

Using these values, we can calculate:

1. $P_{crypto} = P_{selling} - P_{buying} = 9{,}000 - 5{,}000 = 4{,}000$
2. $P_{cgt} = \frac{P}{T_{years}} = \frac{4000}{4} = 1{,}000$
3. $I_{annual} = S_{annual} + P_{cgt} = 340{,}000 + 1{,}000 = 341{,}000$

4. $\tau = f(I_{annual}) = 45\%$
5. $CGT = \tau \; x \; P_{cgt} = 0.14 \times 1{,}000 = 450$
6. $P_{actual} = P_{cgt} - CGT = 1000 - 450 = 550$

These calculations are performed to determine the Capital Gains Tax owed on cryptocurrency transactions based on user inputs and income information.

The program's calculated values for CGT and actual profit should match these manually derived values. By comparing the program's outputs with these manual calculations, we can ensure the accuracy of its computations.

Calculation of Predicted Profit of Future Cryptocurrency Investments

The second task of the program involves predicting the profit of future cryptocurrency investments. The user provides inputs regarding their investment strategy, including initial investment amounts and yearly investment increments, along with the choice of cryptocurrency. Predicted profit rates for different cryptocurrencies are predefined.

Formula:

1. $P_{yearly}(i) = \begin{cases} I_{initial} \times r, & for\ i = 1 \\ (I_{initial} + \sum_{j=1}^{i-1} I_j) \times r, & for\ i > 1 \end{cases}$

2. $P_{total}(i) = \begin{cases} P_{yearly}(i), & for\ i = 1 \\ P_{total}(i-1) + P_{yearly}(i), & for\ i > 1 \end{cases}$

(Refer to this section for full code implementation.)

Where:

- P is the initial profit from crypto currency transaction.
- $I_{initial}$ is the initial Investment amount.
- I_j is the additional investment at the end of the j^{th} year.
- r is the predicted profit rate, which is based on coin selection and can be retrieved by function $f(coin)$, where coin is user selected coin.
- $P_{yearly}(i)$ is the yearly profit in the i^{th} year.
- $P_{total}(i)$ is the total profit up to the i^{th} year.

The predicted profit rate is predefined and it's based on the user-selected coin and can be seen in this table.

Calculation:

For this calculation, we will consider the following scenario:

- $I_{initial} = 500$
- $I_2 = 1{,}000$
- $I_3 = 500$
- $r = 18\%$
- We will be predicting profits for the next subsequent three years.

I. Yearly Profit over Three Years:

1. $P_{yearly}(1) = I_{initial} \times r = 500 \times 0.18 = 90$
2. $P_{yearly}(2) = \left(I_{initial} + \sum_{j=1}^{2-1} I_j\right) \times r = (I_{initial} + I_1) \times \mathrm{r} = (500 + 1000) \times 0.18 = 270$
3. $P_{yearly}(3) = \left(I_{initial} + \sum_{j=1}^{3-1} I_j\right) \times r = (I_{initial} + I_1 + I_2) \times \mathrm{r} = (500 + 1000 + 6000) \times 0.18 = 1350$

II. Predicted Total Profit over Three Years:

1. $P_{total}(1) = P_{yearly}(1) = 90$
2. $P_{total}(2) = P_{total}(2-1) + P_{yearly}(2) = P_{total}(1) + P_{yearly}(2) = 90 + 270 = 360$
3. $P_{total}(3) = P_{total}(3-1) + P_{yearly}(3) = P_{total}(2) + P_{yearly}(3) = 360 + 1350 = 1710$

So, the predicted total profit over the three-year investment period would be $1710.

Results and Analysis

We conducted extensive testing to ensure the accuracy and reliability of the program. Test cases covered various scenarios including valid inputs, invalid inputs, boundary cases, and error handling. The program consistently produced correct results, demonstrating its effectiveness in calculating CGT (Capital Gains Tax) and profit for cryptocurrency transactions.

I. Testing the Program with Valid Input

When the user starts executing the program, it will prompt for the name, salary, residential status, the purchase price, the selling price, and the number of years held. The program will attempt to validate each input and process it. Upon success, the program will print Capital Gains Tax details with the following data:

- Tax rate applied to the user.
- Capital Gains Tax the user is liable for.

- Profit after taxation.

```
What's your name? [put your name hit enter] Min Thu Khaing
Your Annual Salary? [input number only] 340000
Are you resident of Australia? [yes/no] no
Buying price [type in positive number] : $5000
Selling price [type in positive number] : $9000
Number of years held [type in positive number] : 4

Capital Gains Tax:
Tax Rate : 45.0%
Capital Gains Tax : 450.0
Profit : 550.0

Would you like to invest? [yes/no]
```

After that the user will then be prompted to indicate if they want to invest or not. If the user enters “no,” the program will end there and print all available user details.

```
Would you like to invest? [yes/no] no

------------------------------------------------

User Details
Name : Min Thu Khaing
Annual Salary : 340000.0
Residential Status : No

Crypto Currency
Buying Price : 5000.0
Selling Price : 9000.0
Number of years held : 4

Capital Gains Tax:
Tax Rate : 45.0%
Capital Gains Tax : 450.0
Profit : 550.0
```

If the user enters “yes,” the program will continue, and the user will be prompted to enter the initial amount of investment they are willing to make followed by amount for sequential year.

```
Would you like to invest? [yes/no] yes
Enter initial investment amount (cannot exceed $550.0): $
```

Upon success, the program will print Projected Profits with the following data:

- Projected yearly profit they are going to receive.
- Projected total profit they are going to receive each year.

```
Would you like to invest? [yes/no] yes
Enter initial investment amount (cannot exceed $550.0): $500
Enter investment amount after year 1: $1000
Enter investment amount after year 2: $6000
Choose the Cryptocurrency to invest in
1 for Best Coin (predicted profit rates 18%)
2 for Simple Coin (predicted profit rates 12%)
3 for Fast Coin (predicted profit rates 15%)
[type 1/2/3]: 1
You Selected BestCoin
Predicted Profit for Investment in BestCoin
Years   |YearlyProfit          |TotalProfit
--------|----------------------|---------------
1       |$90.00                |$90.00
2       |$270.00               |$360.00
3       |$1350.00              |$1710.00
```

In the end, the program will print all available data for the user.

```
-----------------------------------------------

User Details
Name : Min Thu Khaing
Annual Salary : 340000.0
Residential Status : No

Crypto Currency
Buying Price : 5000.0
Selling Price : 9000.0
Number of years held : 4

Capital Gains Tax:
Tax Rate : 45.0%
Capital Gains Tax : 450.0
Profit : 550.0

Predicted Profit for Investment in BestCoin
Years   |YearlyProfit          |TotalProfit
--------|----------------------|---------------
1       |$90.00                |$90.00
2       |$270.00               |$360.00
3       |$1350.00              |$1710.00
```

II. Manual Calculation Validation

To ensure the accuracy of our calculations, we manually computed the CGT (Capital Gains Tax) and actual profit for the provided scenarios. We compared these manual calculations, which can be seen in the Calculation Section, with the results generated by the program. Here are the comparisons:

CGT Calculation for Cryptocurrency Transactions:

- Manual CGT Calculation: $450 (refer to Calculation Section)
- Program's CGT Calculation: $450 (Match)
- Result: The program's calculated CGT matches the manual calculation.

Actual Profit Calculation for Cryptocurrency Transactions:

- Manual Actual Profit Calculation: $550 (refer to Calculation Section)
- Program's Actual Profit Calculation: $550 (Match)
- Result: The program's calculated actual profit matches the manual calculation.

Predicted Total Profit over Three Years:

- Manual Prediction: $1,710 (refer to Calculation Section)
- Program's Prediction: $1,710 (Match)
- Result: The program's predicted total profit matches the manual calculation.

The program consistently produced correct results, confirming its effectiveness in calculating CGT and profit for cryptocurrency transactions.

III. Testing the Program with Invalid Inputs

Extensive testing was conducted to evaluate the program's error-handling abilities, covering scenarios like using special characters or numbers in names, inputting negative values, leaving fields empty, and exceeding specified ranges.

1. Name

First: Using numbers.

```
What's your name? [put your name hit enter] 42
Please enter a valid name (letters only, no numbers or special characters).
What's your name? [put your name hit enter]
```

Second: Using special characters.

```
What's your name? [put your name hit enter] Mommy! Amera Hpone ^_^
Please enter a valid name (letters only, no numbers or special characters).
What's your name? [put your name hit enter]
```

Third: Using an empty input.

```
What's your name? [put your name hit enter]
Please enter a valid name (letters only, no numbers or special characters).
What's your name? [put your name hit enter]
```

2. Salary

First: Using special characters.

```
Your Annual Salary? [input number only] 7500&
Please Enter a valid number in correct format!
Your Annual Salary? [input number only]
```

Second: Using numbers other than a positive value.

```
Your Annual Salary? [input number only] -133700
Please Enter a valid number in correct format!
Your Annual Salary? [input number only]
```

Third: Using an empty input.

```
Your Annual Salary? [input number only]
Please Enter a valid number in correct format!
Your Annual Salary? [input number only]
```

3. Residential Status

First: Using numbers.

```
Are you resident of Australia? [yes/no] 42
Invalid input. Please enter 'yes' or 'no'.
Are you resident of Australia? [yes/no]
```

Second: Using special characters.

```
Are you resident of Australia? [yes/no] #$$
Invalid input. Please enter 'yes' or 'no'.
Are you resident of Australia? [yes/no]
```

Third: Providing both yes and no at the same time.

```
Are you resident of Australia? [yes/no] yes no
Invalid input. Please enter 'yes' or 'no'.
Are you resident of Australia? [yes/no]
```

Fourth: Using an empty input.

```
Are you resident of Australia? [yes/no]
Invalid input. Please enter 'yes' or 'no'.
Are you resident of Australia? [yes/no]
```

4. Buying Price and Selling Price

First: Using special characters.

```
Buying price [type in positive number] : $1@@0
Please Enter a valid number in correct format!
Buying price [type in positive number] : $
```

Second: Using numbers other than a positive value.

```
Buying price [type in positive number] : $0
Please enter a valid positive number.
Buying price [type in positive number] : $
```

Third: Using an empty input.

```
Buying price [type in positive number] : $
Please Enter a valid number in correct format!
Buying price [type in positive number] : $
```

Fourth: Using a value less than or equal to the Buying Price.

```
Buying price [type in positive number] : $5000
Selling price [type in positive number] : $4999
Please enter a value greater than Buying Price
Selling price [type in positive number] : $
```

5. The number of years held.

First: Using special characters.

```
Number of years held [type in positive number] : 1*2
Please Enter a valid number in correct format!
Number of years held [type in positive number] :
```

Second: Using numbers other than a positive value.

```
Number of years held [type in positive number] : -42
Please Enter a valid number in correct format!
Number of years held [type in positive number] :
```

Third: Using an empty input.

```
Number of years held [type in positive number] :
Please Enter a valid number in correct format!
Number of years held [type in positive number] :
```

Fourth: Using decimal values.

```
Number of years held [type in positive number] : 1.1
Please Enter a valid number in correct format!
Number of years held [type in positive number] :
```

6. Prompt to Invest.

First: Using numbers

```
Would you like to invest? [yes/no] 69
Invalid input. Please enter 'yes' or 'no'.
Would you like to invest? [yes/no]
```

Second: Using special characters.

```
Would you like to invest? [yes/no] &42
Invalid input. Please enter 'yes' or 'no'.
Would you like to invest? [yes/no]
```

Third: Providing both yes and no.

```
Would you like to invest? [yes/no] yes no
Invalid input. Please enter 'yes' or 'no'.
Would you like to invest? [yes/no]
```

Fourth: Using an empty input.

```
Would you like to invest? [yes/no]
Invalid input. Please enter 'yes' or 'no'.
Would you like to invest? [yes/no]
```

7. Input for Initial Investment Amount

First: Using special characters.

```
Enter initial investment amount (cannot exceed $30375.0): $#20492
Please Enter a valid number in correct format!
Enter initial investment amount (cannot exceed $30375.0): $
```

Second: Using numbers other than a positive value.

```
Enter initial investment amount (cannot exceed $30375.0): $-42
Please Enter a valid number in correct format!
Enter initial investment amount (cannot exceed $30375.0): $
```

Third: Using an empty input.

```
Enter initial investment amount (cannot exceed $30375.0): $
Please Enter a valid number in correct format!
Enter initial investment amount (cannot exceed $30375.0): $
```

Fourth: Using a value greater than the profit the user made after taxation.

```
Enter initial investment amount (cannot exceed $30375.0): $30376
Invalid input. Initial investment amount cannot exceed $30375.0.
Enter initial investment amount (cannot exceed $30375.0): $
```

8. Input for Second Year

First: Using special characters.

```
Enter investment amount after year 1: $####
Please Enter a valid number in correct format!
Enter investment amount after year 1: $
```

Second: Using numbers less than zero.

```
Enter investment amount after year 1: $-1337
Please Enter a valid number in correct format!
Enter investment amount after year 1: $
```

Third: Using an empty input.

```
Enter investment amount after year 1: $
Please Enter a valid number in correct format!
Enter investment amount after year 1: $
```

9. Input for Third Year

First: Using special characters.

```
Enter investment amount after year 2: $418(*
Please Enter a valid number in correct format!
Enter investment amount after year 2: $
```

Second: Using numbers other than a positive value.

```
Enter investment amount after year 2: $-42
Please Enter a valid number in correct format!
Enter investment amount after year 2: $
```

Third: Using an empty input.

```
Enter investment amount after year 2: $
Please Enter a valid number in correct format!
Enter investment amount after year 2: $
```

10. Input for Coin Selection

First: Using special characters.

```
[type 1/2/3]: *@
Please Enter a valid number in correct format!
[type 1/2/3]:
```

Second: Using numbers outside the range of 1 to 3.

```
[type 1/2/3]: 0
Invalid input. Please enter a number between 1 and 3.
[type 1/2/3]: 4
Invalid input. Please enter a number between 1 and 3.
[type 1/2/3]:
```

Third: Using an empty input.

```
[type 1/2/3]:
Please Enter a valid number in correct format!
[type 1/2/3]:
```

Conclusion

In conclusion, the development and implementation of the Capital Gains Tax (CGT) calculation program in Java for the SENG1110 Programming Assignment 1 have yielded promising results, supported by comprehensive testing and analysis. The program has effectively met its objectives of accurately calculating CGT and profits from cryptocurrency transactions, as well as projecting future profits based on user inputs.

Testing the program with valid inputs demonstrated its capability to process user data accurately. For instance, when provided with realistic inputs such as purchase price, selling price, and the duration of investment, the program consistently computed CGT percentages and actual profits after taxation in alignment with real-world calculations. This verification against known CGT formulas bolstered confidence in the program's accuracy and reliability.

Through simulated user interactions and inputs, such as providing personal details and transaction data, we observed accurate computation of CGT percentages and post-tax profits. For instance, when prompted to input data user named “Min Thu Khaing” with an annual salary of $340,000 and non-resident status in Australia and given transaction details where the buying price was $5,000, selling price was $9,000, and the investment was held for 4 years, the program correctly calculated a CGT rate of 45.0%, resulting in a capital gains tax of $450.0 and a profit of $550.0.

Furthermore, when presented with the option to invest, the program successfully processed investment amounts over three years, ensuring they did not exceed the available profit. Subsequently, it allowed selection of a cryptocurrency for investment, providing predicted profit rates for each option. Upon choosing "Bestcoin" with predicted profit rates of 18%, the program accurately projected yearly and total profits over the investment period, demonstrating its predictive capabilities and reliability.

Additionally, testing the program with invalid inputs further validated its robustness. For instance, deliberate attempts to input erroneous data, such as using numbers or special characters in the name field or entering negative values for salary, were effectively identified, and rejected by the program. Such meticulous error handling mechanisms ensure that only valid inputs are processed, safeguarding against computational errors.

Finally, the comparison between the program's calculations and manual calculations using real-world scenarios provided a benchmark for assessing its accuracy. By verifying the program's outputs against manually calculated results, discrepancies were minimized, affirming the correctness of the implemented algorithms and logic.

Appendix

Appendix A : Sample Code Snippets

Algorithm for Retrieving and Validating Input String.

```
1.  /**
2.   * Retrieves and validates input from the user.
3.   *
4.   * @param prompt         The message prompting the user for input.
5.   * @param console        The Scanner object used for input.
6.   * @param isName         Boolean value whether it's name or not.
7.   * @param regex          The regular expression used to validate the input.
8.   * @param invalidMessage The message displayed when the input does not match the
9.   *                       Regex.
10.  * @return The validated input from the user.
11.  */
12.
13. private String getValidatedInput(String prompt, Scanner console, boolean isName,
    String regex,
14.                                  String invalidMessage) {
15.     String input;
16.     do {
17.         System.out.print(prompt);
18.         if (isName)
19.             input = console.nextLine();
20.         else
21.             input = console.nextLine().toLowerCase();
22.         if (!input.matches(regex)) {
23.             System.out.println(invalidMessage);
24.         }
25.     } while (!input.matches(regex));
26.     return input;
27. }
```

Algorithm for Retrieving and Validating Number.

```
1.  /**
2.    * Retrieves and validates numerical input from the user.
3.    *
4.    * @param minimum              The minimum allowed value for the input.
5.    * @param maximum              The maximum allowed value for the input. (put -1
6.    *                             to
7.    *                             ignore maximum value check)
8.    * @param acceptEqualToMinimum Boolean value whether going to accept equal
9.    *                             amount to minimum amount.
10.   * @param prompt               The message prompting the user for input.
11.   * @param console              The Scanner object used for input.
12.   * @param regex                The regular expression used to validate the
13.   *                             input.
14.   * @param invalidMessage       The message displayed when the input is invalid.
15.   * @return The validated numerical input from the user.
16.   */
17.
18. private double getValidatedNumInput(double minimum, double maximum, boolean
    acceptEqualToMinimum, String prompt,
19.                                     Scanner console, String regex,
20.                                     String invalidMessage) {
21.     double value = -1; //Setting value to -1 to pass the minimum value check, also
    compiler check, never set it to 0,
```

```
22.      String input;
23.      do {
24.          System.out.print(prompt);
25.          input = console.nextLine();
26.          if (!input.matches(regex)) {
27.              System.out.println("Please Enter a valid number in correct format!");
28.              continue;
29.          }
30.          value = Double.parseDouble(input);
31.          if (value <= minimum && !acceptEqualToMinimum || value < minimum &&
    acceptEqualToMinimum
32.                  || (maximum != -1 && value > maximum)) {
33.              System.out.println(invalidMessage);
34.          }
35.      } while (value <= minimum && !acceptEqualToMinimum || value < minimum &&
    acceptEqualToMinimum
36.              || (maximum != -1 && value > maximum));
37.      return value;
38.  }
```

The yearly Profit and Total Profit calculation is hard coded since we only need three subsequent years.

```
1.  /**
2.   * Calculates the predicted profits for the investment based on the selected
3.   * coin and deposits.
4.   */
5.  public void calculateInvestment() {
6.      /*
7.       * Years Yearly profit Total Profit
8.       * 1 | $500* 0.15 = $75                    | $75
9.       * 2 | ($500 + $1000) * 0.15 = $225        | $75 + $225 = $300
10.      * 3 | ($500 + $1000 + $500) * 0.15 = $300 | $75 + $225 + $300 = $600
11.      */
12.
13.
14.     /*
15.      * Predicted Profit for Investment in Fast Coin
16.      * Years | YearlyProfit | TotalProfit
17.      * ______|______________|_____________
18.      * 1     | $75          | $75
19.      * 2     | $225         | $300
20.      * 3     | $300         | $600
21.      */
22.
23.     double predictedProfitRate;
24.
25.     predictedProfitRate = switch (coinSelection) {
26.         case 1 -> 0.18;
27.         case 2 -> 0.12;
28.         case 3 -> 0.15;
29.         default -> throw new IllegalStateException("Unexpected value: " +
    coinSelection);
30.     };
31.
32.     yearOneProfit = (year1Deposit) * predictedProfitRate;
33.     yearTwoProfit = (year1Deposit + year2Deposit) * predictedProfitRate;
34.     yearThreeProfit = (year1Deposit + year2Deposit + year3Deposit) *
    predictedProfitRate;
35.
36.     yearOneTotalProfit = yearOneProfit;
37.     yearTwoTotalProfit = yearOneProfit + yearTwoProfit;
38.     yearThreeTotalProfit = yearOneProfit + yearTwoProfit + yearThreeProfit;
39. }
```

Algorithm for calculating Capital Gains Tax

```
1.  /**
2.   * Calculates the capital gains tax (CGT) based on user's financial information.
3.   */
4.  public void calculateCgt() {
5.
6.      double profit;
7.      double profitForCGT;
8.      double totalAnnualIncome;
9.      taxRate = 0.0;
10.
11.     // CALCULATION
12.     /*
13.      * Profit = Selling price – Buying price
14.      * Profit for CGT = Profit / Number of years cryptocurrency is held
15.      * Total Annual income = Annual salary + Profit for CGT
16.      */
17.
18.     profit = sellingPrice - buyingPrice;
19.     profitForCGT = profit / years;
20.     totalAnnualIncome = annualSalary + profitForCGT;
21.
22.     /*
23.      * Find the tax rate for Total Annual income as Tax rate.
24.      *
25.      * Tax rates – residents
26.      * $0 – $18,200 0%
27.      * $18,201 – $45,000 19%
28.      * $45,001 – $120,000 32.5%
29.      * $120,001 – $180,000 37%
30.      * Over $180,001 45%
31.      *
32.      * Tax rates – non-residents
33.      * $0 – $120,000 32.5%
34.      * $120,001 – $180,000 37%
35.      * Over $180,001 45%
36.      */
37.     if (resident) {
38.         if (totalAnnualIncome >= 0 && totalAnnualIncome <= 18200) {
39.             taxRate = 0;
40.         } else if (totalAnnualIncome >= 18201 && totalAnnualIncome <= 45000) {
41.             taxRate = 0.19;
42.         } else if (totalAnnualIncome >= 45001 && totalAnnualIncome <= 120000) {
43.             taxRate = 0.325;
44.         } else if (totalAnnualIncome >= 120001 && totalAnnualIncome <= 180000) {
45.             taxRate = 0.37;
46.         } else if (totalAnnualIncome >= 180001) {
47.             taxRate = 0.45;
48.         }
49.     } else {
50.         if (totalAnnualIncome >= 0 && totalAnnualIncome <= 120000) {
51.             taxRate = 0.325;
52.         } else if (totalAnnualIncome >= 120001 && totalAnnualIncome <= 180000) {
53.             taxRate = 0.37;
54.         } else if (totalAnnualIncome >= 180001) {
55.             taxRate = 0.45;
56.         }
57.
58.     }
59.     /*
60.      * Calculation
61.      * CGT = Tax rate * Profit for CGT
62.      * Actual Profit = Profit for CGT - CGT
```

```
63.      */
64.     cgt = taxRate * profitForCGT;
65.     actualProfit = profitForCGT - cgt;
66. }
```

Appendix B : Test Cases

Test Case	Description	Expected Result	Actual Result	Pass/Fail
Start Program – Valid Inputs	User starts executing the program and provides valid input for user details and transaction data. Program validates and processes inputs, then prints CGT details and prompts for investment choice.	Correct computation of CGT and post-tax profit	Correct computation of CGT and post-tax profit	Pass
Continue with Investment	If the user chooses to invest, the program prompts for the initial investment amount and sequential yearly amounts. User provides valid inputs. Program calculates and prints projected profits.	Accurate prediction of future profits	Accurate prediction of future profits	Pass
Print Available Data	The program prints all available data for the user at the end of execution.	All the available user data printed to console	All available user data printed as expected	Pass
Name - Using Numbers	User inputs numbers for their name. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Name - Using Special Characters	User inputs special characters for their name. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Name - Empty Input	User inputs an empty name. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Salary - Special Characters	User inputs special characters for their salary. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Salary - Negative Numbers	User inputs a negative salary value. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Salary - Empty Input	User inputs an empty salary value. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Residential Status - Numbers	User input numbers for their residential status. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Residential Status - Special Characters	User inputs special characters for their residential status. The	Error message indicating invalid input	Error message displayed for invalid input	Pass

	program detects invalid input and prompts for valid input.			
Residential Status - Yes/No Conflict	User provides both 'yes' and 'no' for residential status. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Residential Status - Empty Input	User inputs an empty residential status. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Buying Price - Special Characters	User inputs special characters for the buying price. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Buying Price - Negative Numbers	User inputs a negative buying price value. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Buying Price - Empty Input	User inputs an empty buying price value. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Buying Price - Less than Selling Price	User inputs a buying price greater than or equal to the selling price. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Years Held - Special Characters	User inputs special characters for the years held. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Years Held - Negative Numbers	User inputs a negative value for the years held. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Years Held - Empty Input	User inputs an empty value for the years held. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Years Held - Decimal Values	User inputs decimal values for the years held. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Prompt for Invest - Numbers	User inputs numbers for the invest prompt. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Prompt for Invest - Special Characters	User inputs special characters for the invest prompt. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Prompt for Invest - Yes/No Conflict	User provides both 'yes' and 'no' for the invest prompt. The	Error message indicating invalid input	Error message displayed for invalid input	Pass

	program detects invalid input and prompts for valid input.			
Prompt for Invest - Empty Input	User inputs an empty value for the invest prompt. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Input for First Year - Special Characters	User inputs special characters for the first-year investment amount. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Input for First Year - Negative Numbers	User inputs a negative value for the first-year investment amount. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Input for First Year - Empty Input	User inputs an empty value for the first-year investment amount. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Input for First Year - Exceeds Profit	User inputs an amount greater than the profit after taxation for the first-year investment amount. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Input for Second Year - Special Characters	User inputs special characters for the second-year investment amount. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Input for Second Year - Negative Numbers	User inputs a negative value for the second-year investment amount. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Input for Second Year - Empty Input	User inputs an empty value for the second-year investment amount. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Input for Third Year - Special Characters	User inputs special characters for the third-year investment amount. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Input for Third Year - Negative Numbers	User inputs a negative value for the third-year investment amount. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Input for Third Year - Empty Input	User inputs an empty value for the third-year investment amount. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Input for Coin Selection - Special Characters	User inputs special characters for the coin selection. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass

Input for Coin Selection - Out of Range	User inputs a number outside the range of 1 to 3 for the coin selection. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass
Input for Coin Selection - Empty Input	User inputs an empty value for the coin selection. The program detects invalid input and prompts for valid input.	Error message indicating invalid input	Error message displayed for invalid input	Pass

Appendix C : Tax Rate Tables

Taxable Income	Tax Rate (Residents)	Tax Rate (Non-Residents)
$0 - $18,200	0%	32.50%
$18,201 - $45,000	19%	32.50%
$45,001 - $120,000	32.50%	32.50%
$120,001 - $180,000	37%	37%
Over $180,001	45%	45%

Appendix D : Predicted Profit Rate

Cryptocurrency	Predicted Profit Rate
Bestcoin	18%
Simplecoin	12%
Fastcoin	15%

Appendix E : CGT Calculation Formula

1. Calculation of Profit from Cryptocurrency Transaction(P_{crypto}):
$$P_{crypto} = P_{selling} - P_{buying}$$
2. Calculate Profit for CGT(P_{cgt}):
$$P_{cgt} = \frac{P}{T_{years}}$$
3. Calculate Total Annual Income(I_{annual}):
$$I_{annual} = S_{annual} + P_{cgt}$$

4. Calculate Capital Gains Tax(CGT):

$$CGT = \tau \; x \; P_{cgt}$$

Where τ is the tax rate determined by the function $f(I_{annual})$, which is based on total annual income.

5. Calculate Actual Profit(P_{actual}):

$$P_{actual} = P_{cgt} - CGT$$

Appendix F : Profit Projection Formula

$$P_{yearly}(i) = \begin{cases} I_{initial} \times r, & for\ i = 1 \\ \left(I_{initial} + \sum_{j=1}^{i-1} I_j\right) \times r, & for\ i > 1 \end{cases}$$

$$P_{total}(i) = \begin{cases} P_{yearly}(i), & for\ i = 1 \\ P_{total}(i-1) + P_{yearly}(i), & for\ i > 1 \end{cases}$$

Appendix G : Glossary

Term	Definition
CGT	Capital Gains Tax
Java	A high-level programming language developed by Sun Microsystems (now owned by Oracle)
Regular Expression	A sequence of characters that define a search pattern used in input validation and text processing
Algorithm	A set of step-by-step instructions designed to perform a specific task or solve a particular problem.
Input Validation	The process of ensuring that data entered by the user meets specified criteria or conforms to a predefined format. Input validation helps maintain the integrity and accuracy of the program's computations by preventing invalid or unexpected inputs from being processed.
Prediction	The act of estimating or forecasting future outcomes based on existing data or trends.
Error Handling	The practice of identifying, managing, and resolving errors or exceptions that may occur during program execution.