SQITK3073 BUSINESS ANALYTICS PROGRAMMING

2025/2026 A251

GROUP B

INDIVIDUAL REPORT

TECHNICAL MANUAL REPORT

PREPARED BY:

IMAN HAFIZ BIN MUHAMMAD FAIZ

(306827)

INTRODUCTION

The Malaysian Tax Input System is a simple Python program created to help users calculate their yearly income tax based on the official LHDN tax brackets. This project was developed as part of the Business Analytics Programming course to apply the basic programming skills learned in class. The system allows users to register with a User ID and IC number, log in securely using the last four digits of their IC, and insert their annual income and tax relief values. After that, the program automatically calculates the correct tax payable according to the progressive tax rates used in Malaysia. The tax information is then saved permanently into a CSV file using the pandas library, allowing users to view their tax history at any time. This project demonstrates how simple programming concepts, when combined, can be used to create a useful tool for everyday tasks such as tax calculation.

OBJECTIVE

The objectives of this project are:

1. To apply Python fundamentals such as variables, functions, loops, and conditionals.

2. To implement user login validation using IC Number.

3. To calculate Malaysian income tax using the latest tax brackets.

4. To store tax data using CSV files and pandas.

5. To design a user-friendly text-based menu system.

6. To follow the given requirements by structuring the project into main.py and functions.py.

BACKGROUND

In Malaysia, income tax is calculated based on a progressive system, where higher portions of income are taxed at higher rates. Every year, LHDN updates the tax brackets to ensure fairness and accuracy. Many people find it confusing to calculate tax manually because it requires understanding multiple income ranges and percentages. This project was designed to simplify the process by turning the tax rules into a Python program that performs the calculations automatically. With the use of functions, loops, and conditionals, the program is able to validate the user's IC number, calculate tax based on the correct income bracket, and save the results using the pandas library. By applying the programming techniques taught in class—such as dictionaries, file handling, and modular coding—the project shows how computer science principles can solve real-life problems in a practical and meaningful way.

METHODOLOGY/WORKFLOW
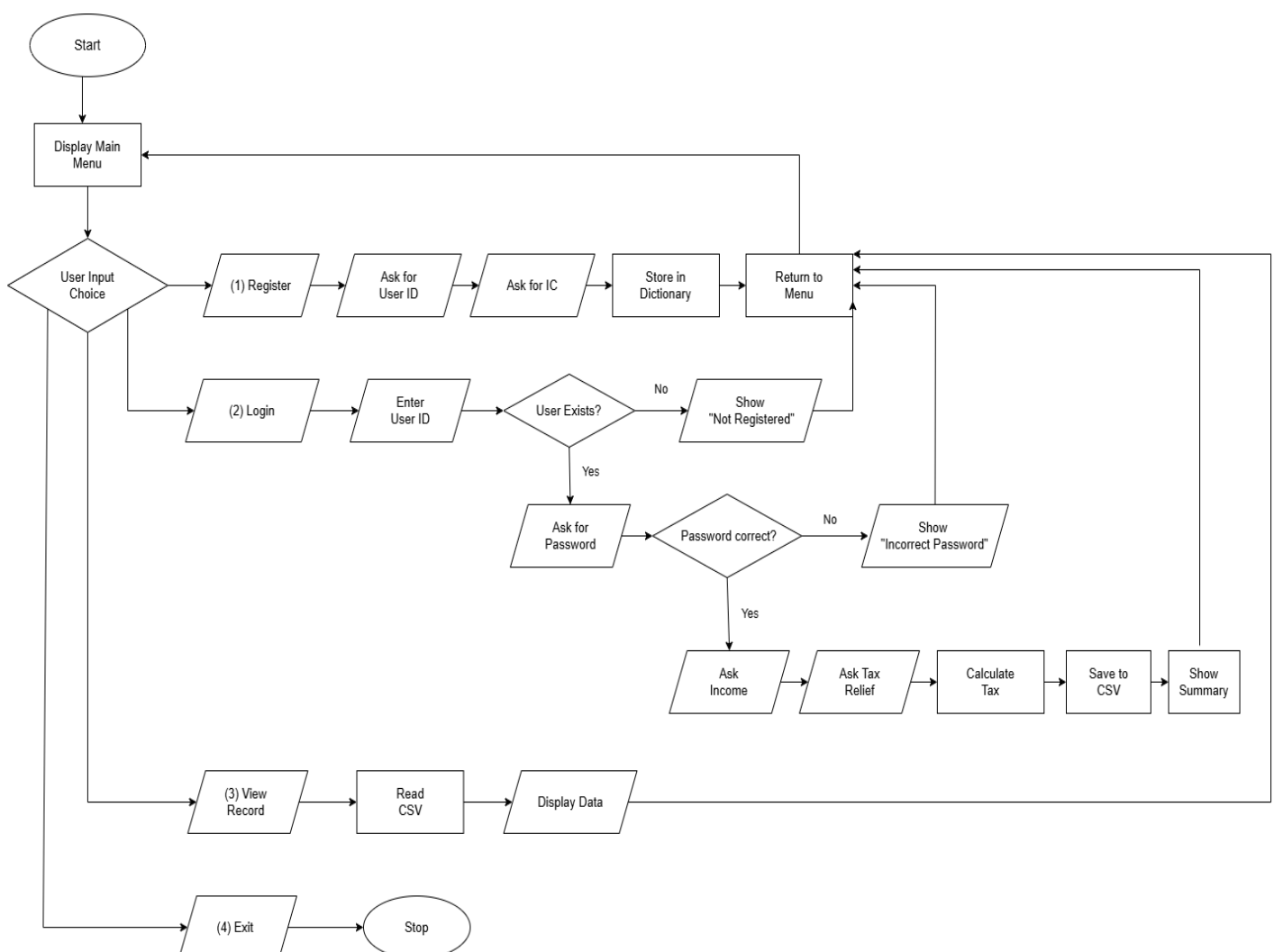
INITIAL SETUP & CONFIGURATION

1. Install Python from website.

2. Install pandas via "`pip install pandas`".

3. Ensure the project is organized into a single folder.
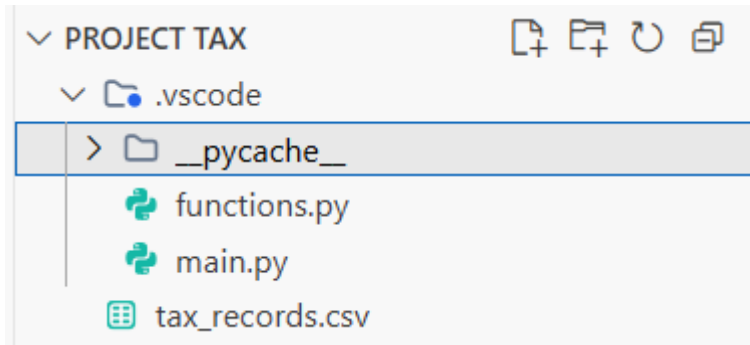
4. Ensure correct file path in `main.py`

5. Run the program using `main.py`

6. Testing the program.

BASIC OPERATIONS

1. **USER REGISTRATION**
   - Function: The user provides a User ID and an IC number (12-digit identification number) during the registration process.
   - Action: The entered User ID and IC number are stored in the `registered_users` dictionary, mapping the User ID to the IC number.
   - Outcome: Upon successful registration, the user is prompted to log in.

2. **USER LOGIN**
   - Function: The user must enter their User ID to log in. After validating the ID, the system asks for the password (which is the last 4 digits of the IC number).
   - Action: The IC number stored during registration is used to verify the password entered by the user. If the password matches, login is successful.
   - Outcome: If the login is successful, the user can proceed to input their Annual Income and Tax Relief.

3. **TAX CALCULATION**
   - Function: After a successful login, the user is prompted to enter their Annual Income and Tax Relief (amount of relief they qualify for).
   - Action: The system calculates the Chargeable Income (Annual Income - Tax Relief) and computes the Tax Payable based on the chargeable income.
   - Outcome: The tax summary, including the chargeable income, tax relief, and tax payable, is displayed to the user.

   Tax Bracket Breakdown:
   1. Bracket 1 (0 – 5000): Tax rate 0% (No tax for the first RM 5000.)
      We don't need to do anything for this bracket because no tax is applied here.
   2. Bracket 2 (5001 – 20000): Tax rate 1%
      For the next RM 15000 (5001 to 20000), we calculate 1% tax:

```python
# 5001 – 20000 : 1% on next 15000
if chargeable > 5000:
    tax += min(chargeable - 5000, 15000) * 0.01
```

This checks whether the chargeable income is greater than 5000. The chargeable income is the income that remains after subtracting allowable tax reliefs. If the income is greater than 5000, the tax calculation for the next range will be executed. If not, this block will be skipped, meaning no tax is calculated for this portion.

This part calculates the portion of income above 5000. If the chargeable income is 15000, this would give you 10000 (the income that is taxed).

- min(chargeable - 5000, 15000): This takes the smaller of two values:
- The amount above 5000 in the chargeable income.
- 15000 because the tax is only calculated on the next 15000 of chargeable income (from 5001 to 20000).

If the chargeable income is 18000, this would result in 13000 (because 18000 - 5000 = 13000, and the cap is 15000).

If the chargeable income is 25000, it would use 15000 because the tax is only calculated on the next 15,000, even though the chargeable income is higher.

* 0.01: This multiplies the taxable amount by 1% (or 0.01 in decimal form) to compute the tax due for that portion of income.

Example 1:

• If the chargeable income is 18000:

• chargeable - 5000 = 18000 - 5000 = 13000

• min(13000, 15000) gives 13000 (since 13000 is less than the cap of 15000).

• 13000 * 0.01 = 130 (This means the tax payable on this portion is RM130).

Example 2:

• If the chargeable income is 25000:

• chargeable - 5000 = 25000 - 5000 = 20000

• min(20000, 15000) gives 15000 (since 15000 is the cap for this range).

• 15000 * 0.01 = 150 (This means the tax payable on this portion is RM150).

Similarly, the same concept is applied to the other tax brackets. For each income range, the system calculates the tax payable on the portion of chargeable income that falls within that range, using the appropriate tax rate. The tax is computed based on the applicable portion of the income, with the use of the min() function ensuring that only the relevant portion is taxed for each bracket."

4. SAVING TAX TO CSV

- Function: After the tax calculation, the system creates a data dictionary that contains the user's IC number, Income, Tax Relief, and Tax Payable.
- Action: The data dictionary is saved to a CSV file (named `tax_records.csv`) using the `save_to_csv()` function.
- Outcome: The tax record is successfully saved, and the user is notified that the record has been stored.

5. VIEWING SAVED TAX RECORDS

- Function: Users can view previously saved tax records.
- Action: The system reads the contents of the CSV file (`tax_records.csv`) using the `read_from_csv()` function.
- Outcome: If records exist, they are displayed in a table format. If no records are found, the user is notified that no records exist.

TROUBLESHOOTING

1. **Program shows "User not registered!**

   The entered User ID does not exist in the registered_users dictionary.

   <u>How to fix:</u>

   - Make sure you have registered first via Option 1, Register in the main menu.
   - Check that you typed the User ID exactly as created.
   - Restart the program if needed and register again.

2. **Program shows "Incorrect password!"**

   The password entered does not match the last 4 digits of the IC number stored during registration. The login process calls verify_user(ic, password) to check this.

   <u>How to fix:</u>

   - Ensure you are entering only the last 4 digits of your 12-digit IC as the password.
   - Check that you used the correct IC number during registration.
   - If you are unsure, re-register with a new User ID and correct IC number.

3. **Program shows "No records found." when viewing tax records**

   read_from_csv(filename) returned None. This usually means:

   - The file tax_records.csv does not exist yet, or
   - The file exists but is empty or in an unexpected format.

   <u>How to fix:</u>

   - First, complete at least one successful login and tax calculation so that save_to_csv(data, filename) can create or update tax_records.csv.
   - Make sure the program has permission to read and write in the folder where main.py is located.
   - If the CSV file has been manually edited and corrupted, delete tax_records.csv and allow the system to recreate it by running a new calculation.

FREQUENTLY ASKED QUESTIONS (FAQs)

Q1. What is my password for login?

Your password is the last 4 digits of your 12-digit IC number. During login, verify_user(ic, password) checks that the entered password matches these last 4 digits.

Q2. What format should I use for my IC number?

The system expects a 12-digit IC number with digits only, for example: 010203041234. Do not include dashes or spaces (010203-04-1234).

Q3. Why do I have to register before logging in?

Registration stores your User ID and IC number in the registered_users dictionary. During login, the program checks whether the entered User ID exists in this dictionary. If you skip registration, the system cannot find your User ID and will show "User not registered!".

REFERENCES

Jiahao. (2025, July 21). *Tax Calculator 2025 (YA 2024) | Malaysia Personal Tax Relief*. E Stream MSC. *https://www.sql.com.my/income-tax-calculator-malaysia/ya-2024/*

Tax Rate | Lembaga Hasil Dalam Negeri Malaysia. (n.d.). Lembaga Hasil Dalam Negeri Malaysia. *https://www.hasil.gov.my/en/individual/individual-life-cycle/income-declaration/tax-rate/*

W3Schools.com. (n.d.). *https://www.w3schools.com/python/python_functions.asp*