UNIVERSITI UTARA MALAYSIA

SEMESTER A241

SQITK 3073 BUSINESS ANALYTICS PROGRAMMING

INDIVIDUAL ASSIGNMENT : MALAYSIA TAX INPUT SYSTEM

PREPARED FOR:

DR. MOHD AAMIR ADEEB BIN ABDUL RAHIM

PREPARED BY: TAN XIN YI 300370

TABLE OF CONTENT

TABLE OF CONTENT	**2**
INTRODUCTION	**3**
OBJECTIVES	**4**
METHODOLOGY	**5-6**
INITIAL SETUP AND CONFIGURATION	**7-8**
FLOWCHART	**9**
CODE EXPLANATION	**10-13**
- **main.py**	**10-11**
- **function.py**	**12-13**
TROUBLESHOOTING AND FAQs	**14**
REFERENCE	**15**

INTRODUCTION

This project aims to develop a Malaysian Tax Input Program using Python. The primary objective is to create a functional and user-friendly application that enables users to calculate their tax payable based on the annual income inserted and eligible tax reliefs, in accordance with the Malaysian tax regulations. The system incorporates essential programming concepts including user authentication, data validation, file handling and modular code structure. By utilizing Python libraries such as pandas, the program can efficiently store and manage user data in CSV files. This project not only reinforces fundamental programming skills but also offers practical experience in developing real-world applications for financial data processing and analysis.

Why do I use Python?

Python is a high-level, multi purpose programming language renowned for its readability and adaptability. It's designed to be simple to learn and use, making it a popular choice for both new and seasoned developers.

OBJECTIVES

The primary objectives of this project are:

1. Implement user registration and authentication
   - Develop a secure login system using the user IC number as an unique identifier and password which is the last four-digit of IC number to manage access to the program.

2. Automate Tax Calculation
   - Enable users to calculate their tax payable based on their declared annual income and tax reliefs, following with the current Malaysian tax guidelines.

3. Data storage and Retrieval
   - Store and retrieve user tax data, including IC number, income, reliefs and calculated tax, using CSV files with the support of the pandas library.

4. Promote code modularity and reusability
   - Apply best practice by separating business logic into two Python files which are “mainind.py” for user interface and flow control and “functionind.py for reusable functions.

5. Improve user experience and reliability
   - Ensure input validation, error handling, and clear prompts to make the application reliable and user-friendly.

METHODOLOGY

The development of the Malaysian Tax Input Program was carried out by using the below methodology:

1. Requirements Analysis
    - The project began with analysis of the assignment, identifying the key functionalities needed for example, user registration, login verification, tax calculation based on tax relief, data storage and retrieval using CSV files.

2. Program Design
    - In this program, it has been divided into two main modules which is main.py which is used to manage the program flow, handle user interaction, and run the functions from the utility module.
    - Besides that, another module will be function.py which contains reusable functions for user verification, tax computation, file saving and file reading.

3. Function Implementation
    - In function.py, it were implemented to handle core tasks which are:
        - verify_user() : to validates the IC number and password
        - calculate_tax() : to computes tax payable based on user income and relief
        - save_to_csv() : to save the user's data to CSV file (IC number, income, tax relief and tax payable
        - read_from_csv() : to loads and displays previously stored tax data

4. User Interface and Flow
    - A console-based interface was built to interact with users. Users are prompted to register or log-in, enter their annual income and tax relief, users will receive an immediate tax computation feedback.

5. Data management using Pandas
    - The pandas library was utilized for handling CSV files to simplify data storage, retrieval and tabular presentation.

6. Code documentation and quality assurance
    - Comments and docstrings were added to explain the functionality of each part of the code, supporting readability and future maintenance.

INITIAL SETUP AND CONFIGURATION

To run the Malaysian Tax Input Program successfully, there are certain prerequisites and configurations are required.

System Requirements:

- Python version: To run this program, the latest version of Python will be Python 3.8 or above.

Required Software:

- Python Interpreter: Users can install latest Python version from https://www.python.org/downloads/
- Code Editor: There is also a Python-compatible IDE or text-editor for example VS Code.

Directory Structure:

- Before starting at the code, a few files are needed for progress. Firstly, a individual_asm_program/ folder will be open to stored other files. In the folder, there will be main.py, function.py, msia_tax_data.csv and an autorun.cmd file.

Python libraries

- The only external library required in this project is pandas. It can be install using the following command: *pip install pandas*.

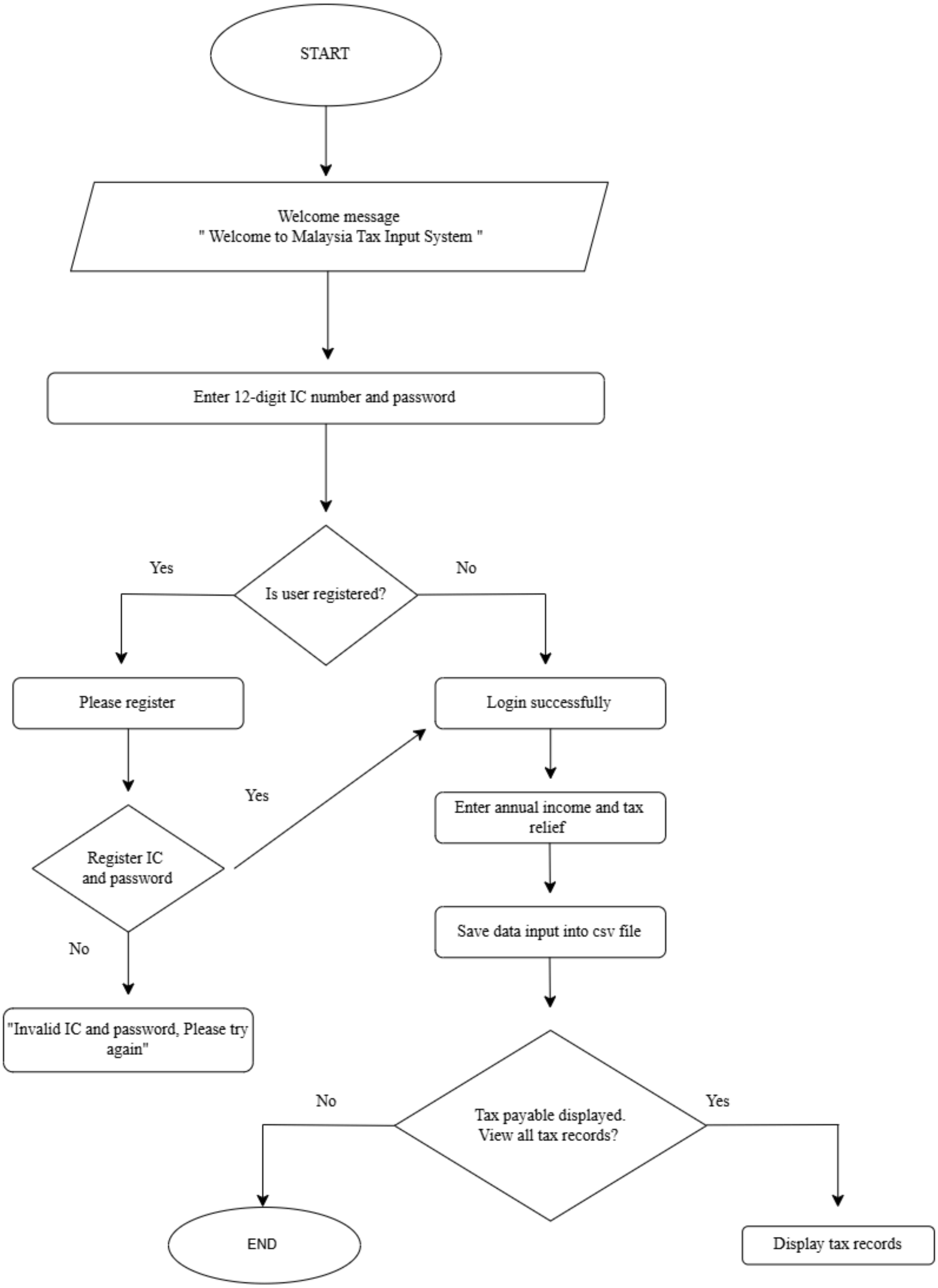
Running the program

- Open command prompt and navigate the project directory using cd: for example, *cd C:\Users\Fion\python program\individual asm*
- To run the program, we will use *pythonmainind.py*

Initial Execution Behaviour

- If it is the user's first time, the program will prompt for ID and IC number for registration.
- It will state that the password must be the last 4-digit of the IC number.
- If a user is registered, the user can log in directly using ID and password, the program will show that user prompt to input income and tax relief to compute tax payable.

FLOWCHART


START
Welcome message
" Welcome to Malaysia Tax Input System "
Enter 12-digit IC number and password
Yes
Is user registered?
No
Please register
Login successfully
Yes
Enter annual income and tax relief
Register IC and password
Save data input into csv file
No
"Invalid IC and password, Please try again"
No
Tax payable displayed. View all tax records?
Yes
END
Display tax records

CODE EXPLANATION : main.py

```
import os
import csv
from functionind import verify_user,
calculate_tax, save_to_csv, read_from_csv
```

Import os
Import csv for data store
Import variables from functionind to run necessary functions of the program

```
def mainind():
   #define the CSV file where the data will
be stored
  csv_filename = os.path.abspath
("C:\Users\Fion\python program\individual
asm\msia_tax_data.csv")
```

This code defines where the csv path of the data stored when user input data information.

```
   #confirm full file path for debugging
   print(f"[DEBUG] Tax data will be saved
to : {os.path.abspath(csv_filename)}")
```

This is to ensure that the tax data imputed is saved in the right CSV path.

```
 #check if user registered
   registered_users = read_from_csv
(csv_filename)
```

This is to confirm whether the user is registered or not where code will read data from a CSV file.

```
   #check if user is registered, else prompt to
registration
   user_found = False
   while not user_found:
      ic_number = input("Enter your 12 digit
IC number: ")
      password = input("Enter the last 4 digit
of your IC number as password: ")
```

When a user is not found in a CSV file, users are prompted to register. To register, users need to insert a 12-digit IC number and password which is the last 4-digit of the IC.

```
 if registered_users is None or ic_number
not in registered_users['ID'].values:
         print("New user. Please register.")
         if verify_user(ic_number, password):
            print("Registration successfully")
            registered_users =
read_from_csv(csv_filename)
            user_found = True
         else:
```

In this, the verified user will run to verify whether the user record is in a CSV file. After verifying, it will show that users have registered successfully.

If a user inserts an invalid IC number which is less than 12-digit, it will print “Invalid IC number and password. Please try again.”

  print("Invalid IC number and password. Please try again.")

```
        else:
 if verify_user(ic_number, password):
                print("Login successful")
                user_found = True
            else:
                print("Invalid password. Please try
again")
```

If the user inserts the correct IC and password, he successfully logs in and proceeds to the next session.

```
# After successfully login, proceed to tax
calculation
    try:
        income = float(input("Enter your
annual income (in RM):"))
        tax_relief = float(input("Enter your tax
relief amount (in RM):"))
    except ValueError:
        print("Please enter valid number for
income and tax relief")
        return
```

After successfully login, it will proceed to tax calculation where users need to input their annual income, tax relief.
The amount that users input must be integer or float else it will print “Please enter valid number for income and tax relief”

```
 #calculate tax payable
    tax_payable = calculate_tax(income,
tax_relief)
    print (f"Your tax payable is: RM
{tax_payable:.2f}")
```

After users insert their annual income and tax relief, tax payable will be calculated and print with “ Your tax payable is RM *x*

```
 #store user data in CSV file
    data_to_save = [ic_number, password,
income, tax_relief, tax_payable]
    print(f"[DEBUG] Saving this data :
{data_to_save}")
    save_to_csv(data_to_save, csv_filename)
```

The data inserted and tax calculated will be saved to CSV.

```
 #display all tax records
    show_record = input("Would you like to
view all tax record? (yes/no):").lower()
    if show_record == "yes":
        tax_record =
read_from_csv(csv_filename)
        print(tax_record)
    else:
        print("Thank you!")
if name == "main":
    maind()
```

After saving all data into a CSV file, users can choose to display its overall tax record. If the user chooses yes, the tax record will be shown, else it will print “Thank you” and go back to the main page.

Code explanation : function.py

```
import pandas as pd
import os
import csv
```

Import pandas for dataframe
Import os
Import csv for data storing

```
# to verify user
def verify_user(ic_number, password):
   """Verify the user's IC number and password."""
   if len(ic_number) == 12 and ic_number.isdigit():
      if password == ic_number[-4:]:
         return True
      else:
         return False
   return False
```

This is to verify the user input IC number and password. The IC number must be 12-digit and the password must be the last 4-digit of the IC number else it is not verified.

```
# to calculate tax
def calculate_tax(income,tax_relief):
   """Calculate tax payable based on income and tax relief."""
   taxable_income = income - tax_relief
   if taxable_income <= 50000:
      return taxable_income * 0.01
   elif taxable_income <= 100000:
      return taxable_income * 0.05
   elif taxable_income <= 150000:
      return taxable_income * 0.1
   else:
      return taxable_income * 0.15
```

This is to calculate tax payable for users. The tax rates for the current year is input and will be divided based on their annual income.

```
# to save user's data into csv file
def save_to_csv(data, filename):
   """save data to a CSV file."""
   file_exists = os.path.isfile(filename)
   with open(filename, mode= 'a', newline= '') as file:
      writer = csv.writer(file)
```

This is the place where data will be saved. The right path of the CSV is placed here and every new user will automatically open a new line in the CSV file.

```
 if not file_exists:
           writer.writerow(["ID", "password using IC number", "Income(RM)", "Tax Relief(RM)", "Tax payable (RM)"])
       writer.writerow(data)
```

If a user file does not exist in a CSV file, a new line of user’s ID, password, income in RM, tax relief in RM, tax payable in RM will be saved in a new row.

```
# to read data from CSV
def read_from_csv(filename):
   """Read data from a CSV file and return it as pandas Dataframe"""
   if os.path.exists(filename):
       return pd.read_csv(filename)
   return None
```

This is for a program to read files from CSV to check whether user existence and for tax records to be shown.

TROUBLESHOOTING AND FAQs

In troubleshooting, it will provide some solutions to common issues users might face while running or interacting with the Malaysian Tax Input Program.

1. Get an error on “ModuleNotFoundError: No module named ‘pandas’ ”
    - This means that the pandas library is not installed. It can install by running *pip install pandas*

2. My IC number is rejected during registration or login
    - You will need to ensure that you entered exactly 12 digits for the IC number and the password is correctly entered as the last 4-digits of the same IC number.
    - You will also need to make sure that there are no extra spaces or characters.

3. Did not find the CSV file after saving.
    - Ensure that the program has write permission in the directory where it is being executed.
    - Make sure that the file named “msia_tax_data.csv” is in the same folder as the .py file for the project, unless it should be specified in the code. For example: *“csv_filename = os.path.abspath(r"C:\Users\Fion\python program\individual asm\msia_tax_data.csv")*

4. Tax amount seems wrong.
    - You will need to make sure the annual income and tax reliefs are input correctly.
    - The tax is computed as (Annual Income - Tax Relief) x Tax Rate.

REFERENCE

LEMBAGA HASIL DALAM NEGERI MALAYSIA (LHDN) （2025， January 23). *Global Minimum Tax* https://www.hasil.gov.my/

Python software Foundation (2025, May 7). *Python 3.13.3 documentation* https://docs.python.org/

Pandas Documentation (2024, September 20). *Pandas version 2.2.3* https://pandas.pydata.org/docs

SQITK3073 - Business Analytics Programming Course Material