CPT113 Programming Assignment 1
Semester 2, Academic Session: 2023/2024

Prepared by: Dr. Nur Hana Samsudin

My Health Journal

Course Outcomes (CO), Program Outcomes (PO), Taxonomy Level (LT) and Soft Skills	*At the end of this course the students will be able to:*	PO	LT	SS	Assessment Methods
	Use data structures in problem solving and object-oriented programming.	PO1	C3		Test (13), Final Exam (PA)
	Analyze the requirements and design for object oriented programming and problem solving.	PO2	C4	CTPS	Test (13), Final Exam (PA)
	Construct object-oriented programming with the appropriate data structures.	PO3	P4	CTPS	Assignment (03)

1. Programming Scenario

can we add input files: so we can set password for admin can we edit input file, so that we can determine the activity level from user, or we just prompt the activity level from user

You are required to design and implement a **console-based project "My Health Journal"** using **Object-Oriented Programming in C++**. The system must record and manage health information for **multiple users**, focusing on body measurements, diet, workouts, and progress tracking.

- the POV of our system now is from user
* now we need to implement POV from admin

for the progress tracking, we need to do for daily or weekly - logically from our POV we think that is daily, cuz if weekly, how can we ensure the users come log into the system everyday

Core Functionalities:

- **Body Measurements:** Store and update weight, height, BMI, etc.

BMR, TDEE, GOAL

- **Diet Protocols:**
  - Read meals from input files.

done: distributed breakfast, lunch, dinner and snacks

  - Ensure minimum calorie intake (≥1200 for females, ≥1500 for males).

done for this requirement, but we only need to fulfill this requirement right, or we need to make sure our calorie must greater than BMR and lower than TDEE for loss/ greater than TDEE for gain

  - Generate or allow user-selected daily meals.

auto-generated suggestion (for daily meals)

  - Provide suggestions if calorie limits are not met or exceeded.

done: must choose breakfast, lunch, dinner Ques: Snacks is it not compulsory, as now we implement if we compulsory for user to choose breakfast, lunch ang dinner, and we prompt user whether they want to add snacks or not, then after than display the total calories intake, then it will compare the meal that user choose with the auto generated suggestion(then here we use operator overloading to compare)

- **Workout Calculations:**

done we have show max duration for workout and total calories burn

  - Calculate workout-based calorie burn.
  - Cap daily burn (max 400 for females, 500 for males).

- **Progress Tracking:**

we assume this feature is when add new user

  - Allow users to set goals (lose/maintain/gain weight).
  - Suggest calorie intake or workout plans accordingly.

adjusted TDEE

Minimum Program Requirements (OOP Concepts):

breakfast,lunch,dinner,snack

1. **Inheritance:** Must include base and derived classes with meaningful relationships.
2. **Composition:** Derived classes must include objects as private members.

Person has a BodyMetric

use together with operator overloading

3. **Friend Function:** Include one justifiable friend function.

compare system suggested meal and user selected meal (calories) && leaderboard to compare user BMI

4. **Operator Overloading:** Implement suitable operator overloads (e.g., combine/compare objects).
5. **Class Interaction:** Ensure logical interactions among classes.
6. **Menu-Driven Interface**.

2. Program Specifications

- Must be executable on **Google Colab** (in .ipynb format).
- Include a **UML Diagram** and the **input files** (using %%writefile).
- Only **one object** in **main()** (arrays of objects allowed).
- main () is the main function, **NOT** a function used to call other functions, that you declare all the objects inside it.
- Emphasize **data encapsulation** and **information hiding**.
- Must use **multiple file inclusion** (header, implementation, and main).
- Files must be read **once at start**, and saved **once at termination** (processing must not done on the file directly).

can we more than 3 file exp now have Person.h,BodyMetric.h and main.cpp, we would like to add more file is it possible

Key Features to Include:

- Use of **inheritance** and **composition**. Once two classes established one relationship, they cannot have another relationship between themselves.
- Using **friend function.** Friend **function(s)** usage is justifiable. Not simply used because you can.
- **Operator overloading**.
- Text file-based **input/output** for persistence.
- **Input validation** for user inputs.
- A clear, user-friendly **menu** and program **flow**.
- Program is interactive and do not terminate prematurely.
- **Proper commenting** in source code.

Need to change do you want continue

Important Note 1: There will be a Q&A session for each group after the assignment due date. Marks can differ slightly.

Important Note 2: **Do not submit the Google Colab link!** Download the complete Google Colab file(.ipynb) from your Google Drive, then submit the files. Only one person per group needs to submit. Failure to comply means that you are not submitting the assignment despite providing a link for the submission.

Important Note 3: Your program complexity and clarity will also contribute towards your marks.

3. Documentation Format and Guidelines

No external report required. Instead, document your program inside the **Google Colab notebook** using **text cells** and **code cells**.

Your notebook must include:

- Program scenario.
- Description of the program requirements.
- The problem your program is trying to solve using object-oriented programming.
- Analysis of the problem.
- How to run your program

Program specification and deliverables.

- Describe all inputs/outputs/processes/constraints/assumptions of your program.
- Your codes (separated into header, implementation and main files).
- Sample of cases tested on your program (you can use different input files for different executions).
- The UML diagram of your class structure (including inheritance, composition, friend function components) must be readable from .ipynb file. At the same time, you can also uploaded the file separately (make sure it is public) in case you worried about readability but the diagram need to be combined and compressed with the .ipynb file and submit as a zip file. You do not need to do the flowchart of the program.

4. Program Restrictions

You **must not** use:

Remove static

- Global variables (except const).
- STL containers: <vector>, <list>, <queue>, or other build-in structure library
- Preprocessors not taught in class.
- Public access specifier for member variables. change dietprotocol.h, workout.h
- Friend classes.
- main() function as dummy function (call another function and create multiple object in that called function).

You **must**:

- Read from input files **and** from user interaction.
- Use multiple files inclusion.
- Make composition objects private.

⚠️ Plagiarism is an academic crime. If caught all members of the groups involved will obtained an '**F**'.

5. Dates

- This assignment questions is released on 26th April 2025 at 00:00
- This assignment due is on 23rd May 2025 at 23:59
- You need to make a selection of the demo session from 16th May 2025. The selection of demo sessions will be closed on 23rd May 2025 at 12:00 noon. If there is any issue with the slots, kindly inform me **BEFORE** 21st May 2025.
- All group members must attend the demo session.
- Group failed to book a slot within the time stated will be considered not completing their Assignment 1.