PW 02 : Basic concept of OOP

Exercise 01: From Procedural to Object-Oriented Programming / Calculating the Area and Perimeter of a Rectangle

This exercise aims to help you understand the difference between:

- Procedural programming (using the C language), and
- Object-Oriented Programming (OOP) (using Java),

through a simple mathematical problem: calculating the area and perimeter of a rectangle.

Part 1 – Procedural Version (C Language)

Write a C program named **rectangle_procedural.c** that calculates:

- the area of a rectangle, and
- the perimeter of a rectangle.

Your program should:

1. Declare two variables length and width (type double).
2. Define a constant named NUMBER_OF_SIDES with the value 4.
3. Implement two functions:
   - calculateArea(double length, double width)
   - calculatePerimeter(double length, double width)
4. In the main() function, assign values to length and width, then display:
   - the length and width,
   - the calculated area,
   - the calculated perimeter.

In this version, data and functions are kept separate (this is the procedural approach).

Part 2 – Object-Oriented Version (Java Language)

Write a Java program named **RectangleOOP.java** that performs the same calculation, but this time following the object-oriented programming model.

Your program should:

1. Define a **class** named RectangleOOP.
2. Declare two **variables**: length and width (type double).
3. Define a **constant** named NUMBER_OF_SIDES with the value 4.
4. Create two **methods**:

- calculateArea()
- calculatePerimeter()

5. In the main(String[] args) method:
   - Assign values to length and width.
   - Call the two methods to display the results.

In this version, both data and operations are grouped together inside the same class (this is the essence of object-oriented programming.)

Part 3 – Reflection: Comparing the Two Approaches

After writing both programs on your notebook and testing them, answer the following questions:

1. What are the **main differences** between the procedural (C) and object-oriented (Java) versions?
2. Where are the **data** and **functions/methods** located in each version?
3. What are the **advantages** of procedural programming?
4. What are the **advantages** of object-oriented programming?
5. What are the **disadvantages** of each approach?

Exercise 02:

We want to model a small training management system for a training center. Each training session is taught by one instructor and attended by several learners (up to a maximum of 3).

Classes to Create:

1. *Class Learner*

1. Declare the attributes:
   - String name
   - String firstName
   - String email
2. Create two constructors:
   - A **default constructor** (initializing attributes with "Unknown" or "not.defined@mail.com")
   - A **parameterized constructor** to initialize all attributes.
3. Add a method displayInfo() that prints the learner's first name, last name, and email.

2. *Class Instructor*

1. Declare the attributes:
   - String name
   - String specialty
2. Create a parameterized constructor to initialize these attributes.
3. Create a method displayInfo() that prints the instructor's information.

3. Class Training

1. Declare the attributes:
   - String title
   - Instructor instructor
   - Learner[] learners (array of size 3)
   - int nbLearners
   - static int trainingCount (to count how many training sessions have been created)
2. Create two constructors:
   - A constructor with parameters (title, instructor)
   - A constructor with a single parameter (title), where the instructor is set to null.

Each time a new training session is created, increment the static counter.

3. Add a method addLearner(Learner a):
   - Add the learner to the array if there is space available.
   - Display a message if the training is already full.
4. Add a method displayTraining():
   - Display the training title.
   - Display the instructor (if it exists).
   - Display the list of enrolled learners.
5. Add a static method displayCount() that prints the total number of trainings created.

4. Class MainApp (main class)

1. Declare two Training objects without instantiating them.
2. Create an Instructor object and associate it with a training.
3. Create three Learner objects (one of them using the default constructor).
4. Add the learners to the training sessions.
5. Display the information of each training.
6. Call the static method displayCount().

5.Questions to Answer:

1. What is the difference between:
   - the declaration of an object,
   - the creation of an object, and
   - the instantiation of an object? (Illustrate your answer with an example from your code.)
2. What is the difference between an instance method and a static method? Give an example of each in your program.
3. What is the purpose of the Learner[] learners array in the Training class? Why not use a single Learner variable instead?
4. Why can a static method be called without creating an object? Give a concrete example from your code.