SOLON, MALT JOHN VIANNEY C.
CSIT321F3

(1)

a. OBJECT-ORIENTED PROGRAMMING

- Object-Oriented Programming (OOP) encapsulates data (attributes) and methods (behaviors) into objects, coupling them to promote information hiding. This means objects communicate through defined interfaces without knowing each other's internal implementations. In procedural languages like C, the focus is on actions and functions, while OOP languages like Java, the focus shifts to classes and objects. Classes define data and methods, and objects are instances of said classes. This approach emphasizes creating reusable and modular code, enhancing software engineering practices.

— JN

b. FOUR PILLARS OF OOP

① ABSTRACTION - Simplifies complex systems by exposing only essential features and hiding their implementation details. The selective display of information enables developers to focus on high-level interactions and functionality without getting bogged by intricate inner workings, enhancing clarity & manageability.

② ENCAPSULATION - Involves bundling data and methods within a single unit known as the object. Encapsulation protects the objects' internal state by restricting direct access and allows interaction through well-defined interfaces. This ensures data integrity and reduces the risk of unintended reference.

③ INHERITANCE - Promotes code reuse by allowing new classes or subclasses to inherit attributes and methods from existing classes. The hierarchal relationship streamlines creation and ensures consistency while reducing redundancy, as shared behaviors and characteristics need not to be defined.

— JN

④ POLYMORPHISM - Allows objects to be treated as instances of their parent class, enabling a single interface to represent different underlying forms. The flexibility allows methods to operate on objects of various classes and subclasses, facilitating dynamic method binding and enhancing the extensibility and scalability of the code.

Source

Deitel, H. & Deitel, P. (2002). JAVA How to Program. Pearson Education (Asia) PTE LTD. p.449

— JN 10/29/24

(2)

a. What is MVC?

The MVC (Model-View-Controller) framework is a design pattern used in software development to separate an application into 3 main components: Model, View, and Controller:

- Model: Manages the data and business logic of the application
- View: Handles the presentation layer, displaying data to the user
- Controller: Acts as an intermediary, processing user input and updating the Model and View

b. How do the three logical components of MVC work together?

1. User Interaction: The user interacts with the view (button click)
2. Controller Handling: The View sends user input to the Controller
3. Model Interaction: The Controller updates the Model based on the user input
4. Updating the View: The Model sends updated data back to the Controller
5. Rendering the View: The Controller passes this data to the View, updating the display