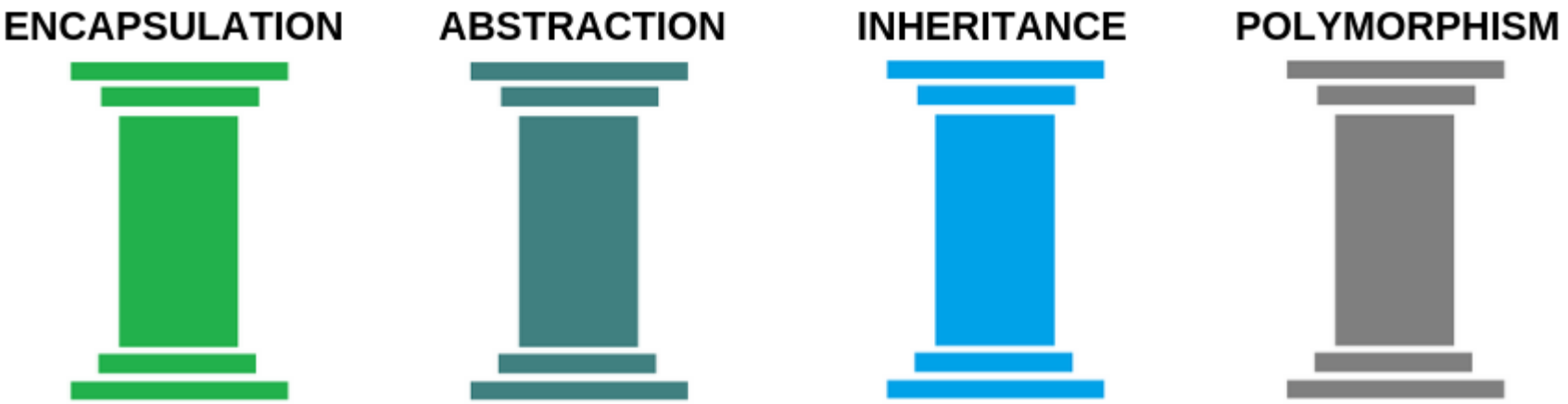
NCP2103: Object-Oriented Programming (Java Programming)

Pillars of Object Oriented Programming

Module 9

Errol John M. Antonio
Assistant Professor
Computer Engineering Department
University of the East – Manila Campus

Four (4) Pillars of OOP



Encapsulation

- Encapsulation could be referred to in simple terms as **data binding**.
- It is all about binding the data variables and functions together in a class. In encapsulation, the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class.
- Sometimes refer to it as **data hiding**.

Encapsulation

- To achieve encapsulation in Java,
  - Declare the variables of a class as private.
  - Provide public getter and setter methods to modify and view the variables values.

Abstraction

- Abstraction is selecting data from a larger pool to show only the relevant details to the object.
- The process of hiding the implementation details from the user, while only the functionality will be provided to the user.
  - In other words, the user will have the information on what the object does instead of how it does it.
- In Java, abstraction is accomplished using Abstract classes and interfaces.

Abstraction

- Abstract classes **may or may not contain abstract methods**, i.e., methods without body.
- If a class has **at least one abstract method**, then the class must be declared **abstract**.
- If a class is declared abstract, **it cannot be instantiated**.
- To use an abstract class, you have to inherit it from another class, provide implementations to the abstract methods in it.
- If you inherit an abstract class, you have to provide implementations to all the abstract methods in it.

Inheritance

- The process where one class acquires the properties (methods and fields) of another.
- The class which inherits the properties of other is known as **subclass** (derived class, child class) and the class whose properties are inherited is known as **superclass** (base class, parent class).

Polymorphism

- The word "**polymorphism**" in simple terms means "**many forms**". It is the ability of an object to take on many forms.
- Polymorphism in Java has two types: **Compile time polymorphism (static binding)** and **Runtime polymorphism (dynamic binding)**.
  - **Method overloading** is an example of static polymorphism, while **method overriding** is an example of dynamic polymorphism.

End of Module.

REFERENCE:

Adeniyi, M. (2019). Deciphering the 4 pillars of Object Oriented Programming (OOP). https://medium.com/@mayokunadeniyi/deciphering-the-4-pillars-of-object-oriented-programming-oop-c2e670c4ad48