ACIT 2515 – Group Assignment 2

Group 13

- **How does your design implement the four pillars of OOP (abstraction, encapsulation, inheritance and composition, and polymorphism)?**

Abstraction: The Abstract class "Person" has 2 functions which return Type and Description for a Person object. However, they are not implemented an actual return but raising an Error because those functions are not implemented properly. Consequently, subclasses (Patient and Doctor) can reuse the abstract methods and extend them based on their concept.

Encapsulation: The Patient and Doctor classes, for example, all attributes are marked as private within the class so that it would prevent users to access the private attributes or methods accidentally. However, those private methods and attributes can be accessed intentionally with appropriate naming convention.

Inheritance: The Person class is a parent class of Doctor and Patient classes. Those subclasses inherit all attributes and methods which are already declared in the Person class (first_name, last_name, get_type(), get_name(), etc). Moreover, they can extend themselves with more private attributes and methods which are not declared in Person class.

Composition: The Department class has a relationship with the Person class in the way that one department can have many types of Person objects. That means the Department class contains an object of Person class. They are in a relationship, but the Department class does not inherit any methods or attributes from Person class.

Polymorphism: The three classes Person, Doctor, Patient have the same 2 methods get_type() and get_description(). However, when calling two methods for each class, the outputs are different based on how the methods are defined within the class itself. In other words, the Patient and Doctor classes override the get_type() and get_description() from the Parent class.

- **Why are your classes good abstractions (i.e., models) of the real-world entities they represent?**

Our abstractions are good for represent real-world entities (different type of person) because the Person class only contains basic information for a given certain person, not all people in a department. There are many types of people and they have different attributes and behaviors to each other. By giving general inner details, other subclasses (Patient and Doctor) can extend their public interface (get_type, get_income, get_bill, etc) from the inner details which are defined in Person class.

Moreover, there might have different type person in a department (Nurse, Security Guard, Technician, etc) that have different behaviors but they still have the same inner details such as name, day of birth, description, id, etc.