Lab 5 –Inheritance and Polymorphism

Answer the following questions.

Instructor-led Demo:

1. Given any requirements, demonstrate inheritance, polymorphism, overriding and overloading program.

Exercise:

1. Implement a class named `Person` and two subclasses of `Person` named `Student` and `Employee`. Make `Faculty` and `Staff` subclasses of `Employee`. A person has a name, address, phone number, and email address. A student has a status (freshman, sophomore, junior, or senior). Define the status as a constant (Hint: Use Enum). An employee has an office, salary, and date-hired. Define a class named `MyDate` that contains the fields year, month, and day. A faculty member has office hours and a rank. A staff member has a title, override the `toString` method in each class to display the class name and the person's name.
    a. Furthermore from Q1, make `FullTime` and `PartTime` subclasses of `Staff`. Full time staff has a fixed salary whereas part time staff has a salary depending on worked hour. Implement this requirement that demonstrate the earning for both staff.
    b. Test your program. Demonstrate the result to the instructor.

2. The `Account` class is to model a bank account. An account has the properties account number, balance, and annual interest rate, and methods to deposit and withdrawal. Create two subclasses for checking and saving accounts. A checking account has an overdraft limit, but a savings account cannot go overdrawn. Test your program.

3. Enabling `GeometricObject` comparable, `Circle` and `Cylinder` are subclasses of `GeometricObject`. Modify the `GeometricObject` class to implement the `Comparable` interface, define the `max` method in the `GeometricObject` class. Write a test program that uses the `max` method to find the larger of two circles and the larger of two cylinders.
    a. Create a class named `ComparableCylinder` that extends `Cylinder` and implements `Comparable`. Implement the `compareTo` method to compare the cylinders on the basic of volume. Write a test class to find the larger of two instances of `ComparableCylinder` objects.
    b. Create an interface named `Colorable` having an abstract method named `howtoColor` method. Every class of a colorable object must implement the `Colorable` interface. Create a class named `Square` that extends `GeometricObject` and implements `Colorable`. Implement `howToColor` to display a message on how to color the square.