0360-212 Winter 2017 Lab Assignment 6

Due@Bb: Mar 26, 11:59pm

Remember to comment your code.

For all the questions below, if there is any ambiguity in the descriptions, make your own assumptions and document them in your code as comments.

Question 1:

Implement a superclass "Person". Make two subclasses, "Student" and "Instructor", that inherit from "Person". A person has a name and a year of birth. A student has a major, and an instructor has a salary. Write the class definitions, the constructors, and the methods "toString" for all classes. You are free to add other methods as you see fit. Supply a test program that tests these classes and methods.

Question 2:

Make a class "Employee" with a name and salary. Make a class "Manger" that inherits from "Employee". Add an instance field, named "department", of type String. Supply a method "toString" that prints the manger's name, department, and salary. Make a class "Executive" that inherits from "Manager". Supply appropriate "toString" methods for all classes. You are free to add other methods as you see fit. Supply a test program that tests these classes and methods.

Question 3:

Design a class named Triangle that extends the GeometricObject class we discussed in class. This class contains:

- Three double data fields named side1, side2, and side3 with default values 1.0 to denote three sides of the triangle.
- A no-arg constructor that creates a default triangle.
- A constructor that creates a triangle with specified side1, side2, and side3.
- The accessor methods for all three data fields.
- A method named getArea() that returns the area of this triangle. (google how to compute area of a triangle given three sides)

- A method named getPerimeter() that returns the perimeter of this triangle.
- A method named toString() that returns a string description for the triangle.

Write a test program to test the various date fields and methods.

Question 4:

Define a class named Payment that contains an instance variable of type double that stores the amount of the payment and appropriate accessor and mutator methods. Also create a method named paymentDetails that outputs an English sentence to describe the amount of the payment.

Next, define a class named CashPayment that is derived from Payment . This class should redefine the paymentDetails method to indicate that the payment is in cash. Include appropriate constructor(s).

Define a class named CreditCardPayment that is derived from Payment . This class should contain instance variables for the name on the card, expiration date, and credit card number. Include appropriate constructor(s). Finally, redefine the paymentDetails method to include all credit card information in the printout.

Create a main method that creates at least two CashPayment and two CreditCardPayment objects with different values and calls paymentDetails for each.

Question 5:

Define a class named Document that contains an instance variable of type String named text that stores any textual content for the document. Create a method named toString that returns the text field and also include a method to set this value.

Next, define a class for Email that is derived from Document and includes instance variables for the sender, recipient, and title of an email message. Implement appropriate accessor and mutator methods. The body of the email message should be stored in the inherited variable text. Redefine the toString method to concatenate all text fields.

Similarly, define a class for File that is derived from Document and includes a instance variable for the pathname. The textual contents of the file should be stored in the inherited variable text. Redefine the toString method to concatenate all text fields.

Finally, create several sample objects of type Email and File in your main method. Test your objects by passing them to the following subroutine that returns true if the object contains the specified keyword in the text property.

```
public static boolean ContainsKeyword(Document docObject, String keyword){

        if (docObject.toString().indexOf(keyword,0) >= 0)

                return true ;

        return false ;

}
```

Think about where to put the above method.