CS 501B – Introduction to JAVA Programming
Fall 2023 Semester
Due: 10/06/2023 Friday at 11:59 PM

Instructions:

1. You are not allowed to use any package/library unless stated otherwise.
2. There is no skeleton given for this assignment.
3. You are required to follow questions and additional requirements.
4. We will be testing your code on hidden test cases.
5. Test your code with your own test cases.
6. Please comment your name and CWID in the first two lines of every **`.java`** file.
7. Add comments in your code about what you are doing.
8. You are required to zip only **`.java`** file as FirstName_LastName_Assignment#.zip (Ex: Roushan_Kumar_Assignment3.zip).
9. This assignment covers topics from week 5.
10. Students are not allowed to collaborate with classmates and any other people outside. All work must be done individually. Any work having evidence of showing academic dishonesty violation is subjected to zero for the assignment.

Penalty:

1. 10 marks will be deducted for invalid format of file / assignment submission.
2. If you submit after the due date then 10 marks will be deducted for every day after the due day.
3. If you submit an assignment after two weeks from the due date then you will get zero marks.
4. You will receive zero, if code doesn't compile / run.

Questions:
Each question carries **25** points and total points is **100.**

1. Create a Java class called **`Calculator`** for performing basic arithmetic operations.

   Fields:
   - This class does not necessarily require any fields, as all operations are done via methods.

   Include:
   - No-argument constructor

   Implement the following methods:

- **`public double add(double a, double b)`**: Takes two double values a and b as parameters and returns their sum.

- **`public double subtract(double a, double b)`**: Takes two double values a and b as parameters and returns the result of a - b.

- **`public double multiply(double a, double b)`**: Takes two double values a and b as parameters and returns their product.

- **`public double divide(double a, double b)`**: Takes two double values a and b as parameters and returns the result of a / b.

2. Create a Java class named **`Student`** for storing student information and calculating grades.

   Fields:
   - **`String name`**: The name of the student.
   - **`int rollNumber`**: The roll number of the student.
   - **`double marks1, marks2, marks3`**: The marks in three subjects.

   Include:
   - **No-argument constructor**: Initializes the **`name`** to an empty string and **`rollNumber`, `marks1`, `marks2`, `marks3`** to zero.
   - **Constructor with parameters**: Initializes all fields with given values.

   Implement the following methods:

   - **`public double calculateAverage()`:** Calculates and returns the average of **`marks1`, `marks2`,** and **`marks3`.**

   - **`public char grade()`:** Calculates and returns the grade based on the average marks. The grading scheme is as follows:
     - A: More than 100
     - B: 80 to 100
     - C: Less than 80

3. Create a Java class named **`BankAccount`** for simulating basic bank operations..

   Fields:
   - **`String accountNumber`**: The account number.

- **`String accountHolder`:** The name of the account holder**.**
- **`double balance`:** The current balance.

Include:

- **No-argument constructor**: Initializes **`accountNumber`** and **`accountHolder`** to empty strings and **`balance`** to zero.
- **Constructor with parameters**: Initializes all fields with given values.

Implement the following methods:

- **`public void deposit(double amount)`:** Increases the **`balance`** by the given **`amount`.**

- **`public void withdraw(double amount)`:** Decreases the **`balance`** by the given **`amount`,** ensuring that the balance does not go below zero.

4. Create a Java class named **`Circle`** for calculating the area and circumference of a circle. Take **`π`** as 3.14.

   Fields:

   - **`double radius`:** The radius of the circle**.**

   Include:

   - **No-argument constructor**: Initializes the **`radius`** to 1.0.
   - **Constructor with parameters**: Initializes the **`radius`** with the given value..

   Implement the following methods:

   - **`public double area()`:** Calculates and returns the area of the circle.

   - **`public double circumference()`:** Calculates and returns the circumference of the circle.

Additional Requirement for All Questions:

- Draw the UML diagram for each class then implement the class. **Submit** a UML diagram in PDF format named **`class_name_UML.pdf`** (example for question 1: **`Calculator_UML.pdf`**).

- Perform each question in a separate **`.java`** file. Name of the file should be **`class_name.java`** and the name of the class is already mentioned in each question.

Ex: for question 1, file name will be **`Calculator.java`** and class name will be **`Calculator`.**
Follow the same convention for other questions.

- Write five test cases for each method of each question in a separate **`.java`** test file. Print the input and output in each test case. Sample of question 1 is given below for this.

  File Name : `class_nameTest.java` -> `CalculatorTest.java`
  ClassName : `class_nameTest` -> `CalculatorTest`

  CalculatorTest.java

```
CalculatorTest {
        public static void main(String[] args){
                        // Test Case
                }
}
```

  Follow the same convention for all other questions for writing your own test case and **submit** it also.