Assignment 1

Due time: 02/10/2022, 6:00pm

Total credits: 100, 5 questions

Submission guide:

1. Create folder and name it with the format FirstName_LastName_Aassigment1 for example Chunyu_Yuan_Assignment1

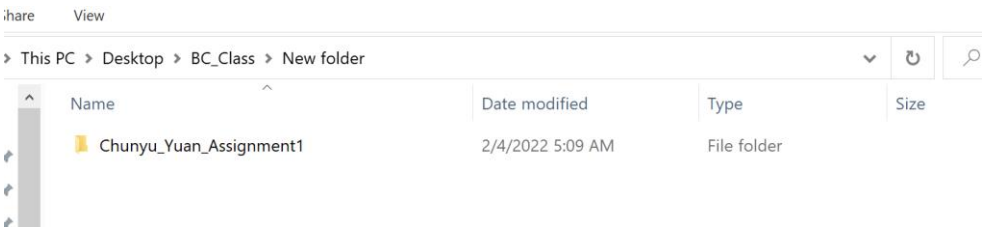
2. Inside the folder, you should have 4 java files and one image for UML, the image file can be png, jpg, jpeg

3. compress your file to .zip format and submit it to the blackboard,

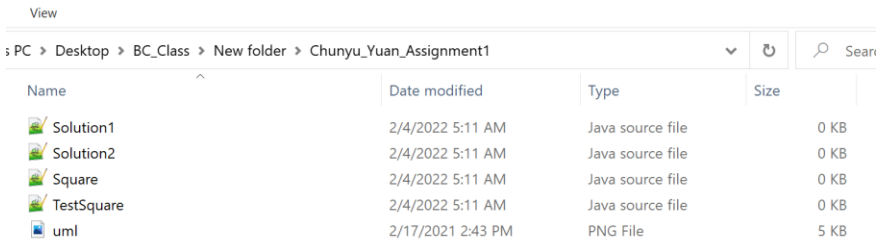
4. if you have any question, please send email to cyuan1@gradcenter.cuny.edu

Below are examples about the folder contend (Square.java, TestSquare.java, Solution1.java, Solution2.java, uml.png):

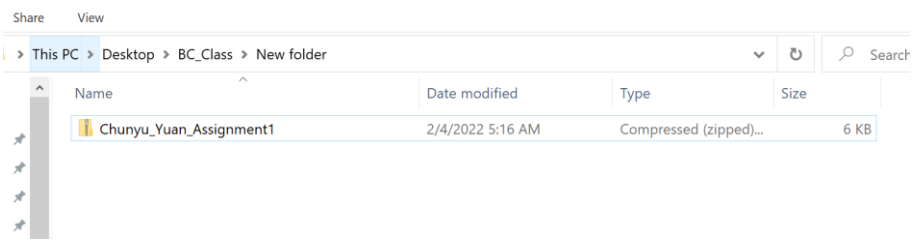
Snapshot1:



Snapshot2:



Snapshot3:



1. design and implement Square Class (20 credits)

Requirements:

- **One data field: side, type double, initialize it to 10**
- **Two constructors: one constructor with argu, one constructor without constructor**
- **three methods:**
  - **setSide, void**
  - **getArea, return type double**
  - **getPerimeter, return type double**

2. design and implement TestSquare Class (40 credits)

Requirements:

- **declare one Square object square1 using one constructor without argu**
- **print out its side using variable reference**
- **print out its Area using method getArea**
- **print out its perimeter using method getPerimeter**
- **using setSide to set its side to 20**
- **print out its side using variable reference**
- **print out its Area using method getArea**
- **print out its perimeter using method getPerimeter**

- **declare one Square object square2 using one constructor with argu 40**
- **print out its side using variable reference**
- **print out its Area using method getArea**
- **print out its perimeter using method getPerimeter**
- **using setSide to set its side to 80**
- **print out its side using variable reference**
- **print out its Area using method getArea**
- **print out its perimeter using method getPerimeter**

(make sure you can compile your program and get the results)

3. draw UML for the Square Class(question 1) (10 credits)

Requirement: make it similar to the Circle example in the slide

4. Question 1 (15 credits)

Given an integer array `nums` of length `n`, you want to create an array `ans` of length `2n` where `ans[i] == nums[i]` and `ans[i + n] == nums[i]` for `0 <= i < n` (**0-indexed**).

Specifically, `ans` is the **concatenation** of two `nums` arrays.

Return *the array* `ans`.

Example 1:

```
Input: nums = [1,2,1]
Output: [1,2,1,1,2,1]
Explanation: The array ans is formed as follows:
- ans = [nums[0],nums[1],nums[2],nums[0],nums[1],nums[2]]
- ans = [1,2,1,1,2,1]
```

Example 2:

```
Input: nums = [1,3,2,1]
Output: [1,3,2,1,1,3,2,1]
Explanation: The array ans is formed as follows:
- ans =
[nums[0],nums[1],nums[2],nums[3],nums[0],nums[1],nums[2],nums[3]]
- ans = [1,3,2,1,1,3,2,1]
```

Below is your start code, finish the method "public int[] answer(int[] nums)", don't need to write the main method inside the class Solution

```
class Solution1 {
    public int[] answer(int[] nums) {
      //Your method

    }
}
```

5. Question 2 (15 credits)

You are given an `m x n` integer grid `accounts` where `accounts[i][j]` is the amount of money the `i`th customer has in the `j`th bank. Return *the **wealth** that the richest customer has.*

A customer's **wealth** is the amount of money they have in all their bank accounts. The richest customer is the customer that has the maximum **wealth**.

Example 1:

```
Input: accounts = [[1,2,3],[3,2,1]]
Output: 6
Explanation:
1st customer has wealth = 1 + 2 + 3 = 6
2nd customer has wealth = 3 + 2 + 1 = 6
Both customers are considered the richest with a
wealth of 6 each, so return 6.
```

Example 2:

```
Input: accounts = [[1,5],[7,3],[3,5]]
Output: 10
Explanation:
1st customer has wealth = 6
2nd customer has wealth = 10
3rd customer has wealth = 8
The 2nd customer is the richest with a wealth of 10.
```

Example 3:

```
Input: accounts = [[2,8,7],[7,1,3],[1,9,5]]
Output: 17
```

Below is your start code, finish the method "public int maximumWealth(int[][] accounts)", don't need to write the main method inside the class Solution

```
class Solution2 {
    public int maximumWealth(int[][] accounts) {
        // your method
    }
}
```