INSY 4305 ADVANCED APPLICATION DEVELOPMENT

ASSINGMENT 4

100 points

1. INSTRUCTIONS

- **Due date is Apr 5, 11:59 pm. Late submissions will get 0 points. NO EXCUSE!**
- **PLEASE, ONLY USE TECHNIQUES THAT WE LEARNED IN CHAPTER 8 AND CHAPTER 9.**
- In this assignment, you are expected to create **five** Java applications.
- Each question is **independent** of each other.
- You will **upload five Java files** on Blackboard. **If you do not upload .java files, your answers will not be evaluated.**
- **Do not forget to add comments to explain how your codes are working! Comments must be brief, clear, and understandable. Do not write long sentences!!!**
- **Write your codes individually! Do not copy of any of them from someone else!**

2. GRADING POLICY

- **Case 1:**
  - For each question:
  - I will compile your .java files. **If any compilation error occurs, 5 pts will be deducted.**
  - After that, I will check your algorithms whether they are correct or not. For example; if it says find odd and even numbers. I will check whether it really finds both even and odd numbers. **This part will be evaluated based on your work.**
  - Additionally, comments will be checked whether they clearly and briefly explain what you have done. **If comments are missing or not clear, enough, or brief 3 pts will be deducted.**

- **Case 2:**
  - For each question:
  - If there is not any compilation error:
    - I will try each case scenario stated in each question. For example; if it says find odd and even numbers. I will try both even and odd numbers. **This part will be evaluated based on your work.**
    - Additionally, comments will be checked whether they clearly and briefly explain what you have done. **If comments are missing or not clear, enough, or brief 3 pts will be deducted.**
- **Case 3:**
  - **If you do not upload a .java file, I will not evaluate your answer.**
- **Case 4:**
  - **If it is determined that you copy the codes from someone else, you will get 0 pt.**

QUESTIONS

1. (55 pts) Create a class named ***AccountSavings***. This class has a **static double** variable which stores the annual interest rate for all account holders. The name of variable is ***annualInterestRate***. The class also has another **double** variable named ***savingsBalance*** which stores balance for current account.

a. Write a constructor to create an account with specified balance. Add a validation whether the balance is greater than 0.0 or not. If it is less than 0.0 then throw illegal argument exception.

b. Write a ***non-static calculateMonthlyInterest*** method to calculate the monthly interest by multiplying the savingsBalance by annualInterestRate divided by 12 – the interest should be added to savingsBalance.

c. Write a ***static*** method named ***modifyInterestRate*** to set the annual interest rate. Add a validation whether the rate is greater than equal to 0.0 and less than or equal 1.0. Otherwise, throw illegal argument exception.

d. Write a ***toString*** method which returns savingsBalance in a string format.

After that, create ***AccountSavingsTest*** class. Create **two objects** from the class ***AccountSavings*** with balances $2000.00 and $3000.00. Then, set the interest rate to 4%, then calculate the monthly interest rate for each 12 months for each object and print the new balances with toString method for each object. **UPLOAD AccountSavings.java AND AccountSavingsTest.java.**

Example Output: (**you can create a similar output**)

Note: you should see a period (.) instead of a comma.

```
Monthly balances for one year at .04
Balances:
             Saver 1   Saver 2
Base        $2000,00  $3000,00
Month 1:    $2006,67  $3010,00
Month 2:    $2013,36  $3020,03
Month 3:    $2020,07  $3030,10
Month 4:    $2026,80  $3040,20
Month 5:    $2033,56  $3050,33
Month 6:    $2040,33  $3060,50
Month 7:    $2047,14  $3070,70
Month 8:    $2053,96  $3080,94
Month 9:    $2060,81  $3091,21
Month 10:   $2067,68  $3101,51
Month 11:   $2074,57  $3111,85
Month 12:   $2081,48  $3122,22
```

2. (45 pts) Create a class ***Person*** which is a super class. The class includes four private String instance variables: first name, last name, social security number, and state. Write a constructor and get methods for each instance variable. Also, add a toString method to print the details of the person. After that create a class ***Teacher*** which extends the class, Person. Add a private instance variable to store number of courses. Write a constructor and a get method for the number of courses and override toString method for Teacher. Then, create ***PersonTest*** class and create an object from the class Teacher and an object from the class Person. Display the details of Teacher and Person (use toString methods). **UPLOAD Person.java AND Teacher.java AND PersonTest.java.**

Example Output:

```
Displaying Teacher
Ezgi Akar
socialSecurityNumber: 222-222-222
state: Texas
number of courses: 3
Displaying Person
Brad Pitt
socialSecurityNumber: 333-333-333
state: Los Angeles
```