<div align="center">

WIA1002/WIB1002 Data Structure

<u>Lab 1: Programming Fundamentals</u>

</div>

Instruction: Submit your solutions for all the questions in one zip file named Lab1-*yourName-yourMatricNum*.zip to Spectrum before next Thursday. You should submit the project folder for each question including the .txt file where applicable.

Question 1

Your task is to write a letter to me (your respective lecturer) that has two parts.

Part 1:

In the first part of your letter, you will introduce yourself. An example is as below:

Thursday, 8 February 2018.

My name is XXXX (your name) with matrix number, XXXXX (your matrix number). I am majoring in XXXX (your majoring). This is my first/second/third (specify) time taking the Data Structure subject. At the moment, I am feeling XXXX (your emotion) about taking this subject. This is because XXXX (describe the cause of your emotion about taking DS).

I acquired XXXX (your grade) for my previous Programming 1 course. It's not too bad. So, I think I can manage to get XXXX (the expected grade) for this DS subject this term. In order to do well in the subject, I will XXXX (what will you do). Wish me luck!!!

---

1. Create a text file that introduces you to me. Name this file as yourname_matrixNum.txt (i.e., Unaizah_WXX12345.txt). The content of the first part of your letter can be adapted from the example given above. (However, you are free to write on your own, so long you retain the information). Write/type directly on a Notepad and save them as a text file format (.txt).

2. Create a program to read the file (yourname_matrixNum.txt). Display the contents of the file in the output console. [Project name : ReadMyLetter_matrixNum]

Part 2:

You will append your letter (from part one) with a second part of the letter dated the day of the last lab session of DS (Thursday, 10 May 2018). Assuming you fast forward to the future, reflect and describe:

- How you have performed in the class?
- Are you happy with your performance?
- What has learning DS taught you / what did you learn from DS?
- Is there any change to your target grade?
- What you did well during the course?
- What could have been done better during the course?

3. Write the second part of the letter using the input from Console (Do not type on the txt file directly). To do this, you are required to modify your program above (2) by importing a Scanner class to read input from the keyboard. Your description outlined above should be appended (not overwritten) to the first part of your letter. Display the contents of the letter (part 1 and part 2) in the output console. The final letter should be something like below:

Thursday, 8 February 2018.

My name is XXXX (your name) with matrix number, XXXXX (your matrix number). I am majoring in XXXX (your majoring). This is my first/second/third (specify) time taking the Data Structure subject. At the moment, I am feeling XXXX(your emotion) about taking this subject. This is because XXXX (describe the cause of your emotion about taking DS).

I acquired XXXX (your grade) for my previous Programming 1 course. It's not too bad. So, I think I can manage to get XXXX (the expected grade) for this DS subject this term. In order to do well in the subject, I will XXXX (what will you do). Wish me luck!!!

Thursday, 10 May 2018

It's me again. Finally, it's the end of the term and the DS class has finished! I think I did XXX (your performance) in this course. ……….

Question 2

1. Write a program to read a text file (namely, text1.txt) that has a sequence of characters that are delimited (separated) by *special character* (i.e., comma, semi colons, spaces, etc.). The number of characters can vary from 1 to N. Your task is to :
   a. Calculate the number of characters retrieved from text1.txt without the special characters.
   b. Display all characters from the text file without the *special characters*.

2. Modify your program to perform 1a. and 1b. above on text2.txt, text3.txt and text4.txt

Example of text1.txt (character separated by comma)

```
S,U,C,E,S,S
A,c,c,o,m,p,l,i,s,h,m,e,n,t
B,r,i,l,l,i,a,n,t
C,r,e,a,t,i,v,e
D,e,t,e,r,m,i,n,a,t,i,o,n
E,n,c,o,u,r,a,g,i,n,g
F,o,c,u,s
```

Example of text2.txt (numbers separated by comma and space)

```
15, 2, 9, 78, 33, 61
198, 523, 91, 42, 13, 77
34, 45
```

Example of text3.txt ((real numbers separated by semicolon and space)

```
4.33; 2.51; 6.11; 2.33; 6.31
1.95; 3.67; 2.22
6.84; 5.04; 9.56; 0.92
```

Example of text4.txt (alphabets separated by numbers)

```
abc123def456ghi789jkl
```

Question 3

Implement a class named **Account** that contains:

■ A private **int** data field named **id** for the account (default **0**).
■ A private **double** data field named **balance** for the account (default **0**).
■ A private **double** data field named **annualInterestRate** that stores the current interest rate (default **0**). Assume all accounts have the same interest rate.
■ A private **Date** data field named **dateCreated** that stores the date when the account was created.
■ A no-arg constructor that creates a default account.
■ A constructor that creates an account with the specified id and initial balance.
■ The accessor and mutator methods for **id**, **balance**, and **annualInterestRate**.
■ The accessor method for **dateCreated**.
■ A method named **getMonthlyInterestRate()** that returns the monthly interest rate.
■ A method named **getMonthlyInterest()** that returns the monthly interest.
■ A method named **withdraw** that withdraws a specified amount from the account.
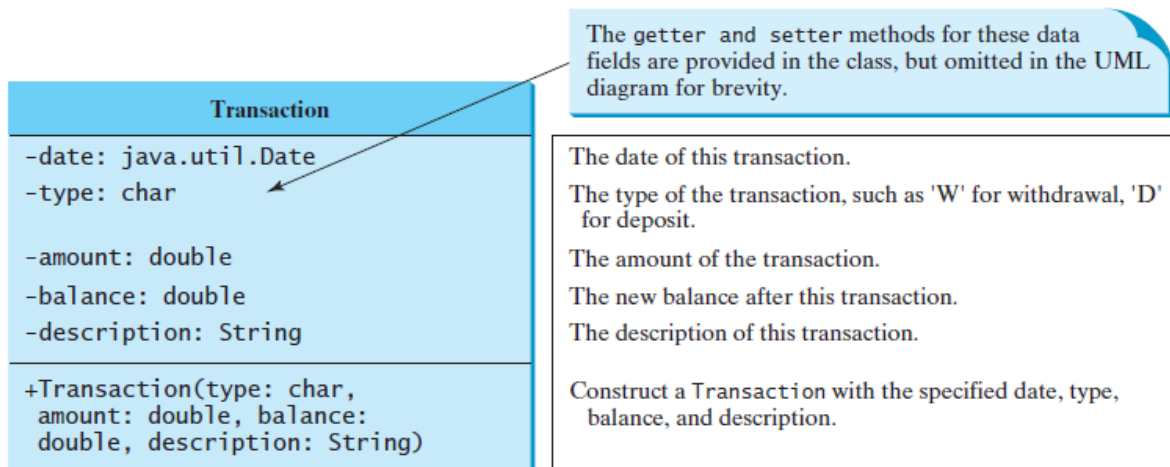■ A method named **deposit** that deposits a specified amount to the account.

(*Hint*: The method **getMonthlyInterest()** is to return monthly interest, not the interest rate. Monthly interest is **balance * monthlyInterestRate**. **monthlyInterestRate** is **annualInterestRate / 12**. Note that **annualInterestRate** is a percentage, e.g., like 4.5%. You need

to divide it by 100.) Write a test program that creates an **Account** object with an account ID of 1122, a balance of $20,000, and an annual interest rate of 4.5%. Use the **withdraw** method to withdraw $2,500, use the **deposit** method to deposit $3,000, and print the balance, the monthly interest, and the date when this account was created.

Question 4

An **Account** class was specified in Question 3. Design a new **Account1** class as follows:
- Add a new data field **name** of the **String** type to store the name of the customer.
- Add a new constructor that constructs an account with the specified name, id, and balance.
- Add a new data field named **transactions** whose type is **ArrayList** that stores the transaction for the accounts. Each transaction is an instance of the **Transaction** class. The **Transaction** class is defined as shown in figure below.

The getter and setter methods for these data fields are provided in the class, but omitted in the UML diagram for brevity.

Transaction	
-date: java.util.Date	The date of this transaction.
-type: char	The type of the transaction, such as 'W' for withdrawal, 'D' for deposit.
-amount: double	The amount of the transaction.
-balance: double	The new balance after this transaction.
-description: String	The description of this transaction.
+Transaction(type: char, amount: double, balance: double, description: String)	Construct a Transaction with the specified date, type, balance, and description.

- Modify the **withdraw** and **deposit** methods to add a transaction to the **transactions** array list.
- All other properties and methods are the same as in Question 3.

Write a test program that creates an **Account1** object with annual interest rate **1.5%**, balance **1000**, id **1122**, and name **George**. Deposit $30, $40, and $50 to the account and withdraw $5, $4, and $2 from the account. Print an account summary that shows account holder name, interest rate, balance, and all transactions.