WIA1002/WIB1002 Data Structure

Lab 1: Programming Fundamentals (Sample Answer)

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, 19 March 2021.
>
> 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., JamesBond_U1234567.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]

Sample Answer:

```
import java.io.BufferedReader;
import java.io.FileReader;


public class ReadMyLetter {
    public static void main(String[] args)
    {
       System.out.println("Reading File from Java code");
```

```
        //Name of the file
       // String fileName="C:\\Users\\Unaizah\\Documents\\NetBeansProjects\
MyLetter\\src\\Unaizah_WXX12345.txt"; //give full url, but only work on
windows platform
        String fileName="Unaizah_WXX12345.txt"; //file stored at the same level
as src folder
        try{

           //Create object of FileReader
           FileReader inputFile = new FileReader(fileName);

           //Instantiate the BufferedReader Class
           BufferedReader bufferReader = new BufferedReader(inputFile);

           //Variable to hold the one line data
           String line;

           // Read file line by line and print on the console
           while ((line = bufferReader.readLine()) != null)   {
             System.out.println(line);
           }
           //Close the buffer reader
           bufferReader.close();
        }catch(Exception e){
           System.out.println("Error while reading file line by line:" +
e.getMessage());
        }
      }
}
```

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, 18 June 2021). 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, 19 March 2021.

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, 18 June 2021

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. …….

Sample Answer:

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.util.Scanner;


public class ReadMyLetter {
    public static void main(String[] args)
    {
       System.out.println("Reading File from Java code");
       //Name of the file
      // String fileName="C:\\Users\\Unaizah\\Documents\\NetBeansProjects\MyLetter\\
src\\Unaizah_WXX12345.txt"; //give full url, but only work on windows platform
       String fileName="Unaizah_WXX12345.txt"; //file stored at the same level as src
folder
       try{

          //Create object of FileReader
          FileReader inputFile = new FileReader(fileName);

          //Instantiate the BufferedReader Class
          BufferedReader bufferReader = new BufferedReader(inputFile);

          //Variable to hold the one line data
          String line;

          System.out.println("Before user input. Content in text file only.");
          // Read file line by line and print on the console
          while ((line = bufferReader.readLine()) != null)   {
            System.out.println(line);
          }
```

```
           //Close the buffer reader
           bufferReader.close();

           Scanner input = new Scanner(System.in);

            System.out.println("\n\nEnter your second part of the letter :\n");
            String strInput = input.nextLine();

            //PrintWriter output = new PrintWriter(fileName); // this one overwrites
exisiting content
            //output.println(strInput); //write new string input on file

            FileWriter outputFile = new FileWriter(fileName, true);

            BufferedWriter output = new BufferedWriter(outputFile);
            output.write(strInput);
            output.close();


            System.out.println("\n\nAfter user input. String and content in text
file.");

            FileReader inputFileAfterAppend = new FileReader(fileName);
            BufferedReader bufferReaderAfterAppend = new
BufferedReader(inputFileAfterAppend);

            String line2;

           // Read file line by line and print on the console
           while ((line2 = bufferReaderAfterAppend.readLine()) != null)   {
             System.out.println(line2);
           }

             //Close the buffer reader
           bufferReaderAfterAppend.close();

       }catch(Exception e){
           System.out.println("Error while reading file line by line:" +
e.getMessage());
       }
     }
}
```

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
```

Sample Answer:

ReadText1File.java

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.util.StringTokenizer;

//Read text1.txt (character separated by comma)
public class ReadText1File {
    public static void main(String[] args) {
       String fileName="text1.txt";
       System.out.println("Reading File from text file : " + fileName + "\n");
       try{
          //Create object of FileReader
          FileReader inputFile = new FileReader(fileName);

          //Instantiate the BufferedReader Class
          BufferedReader bufferReader = new BufferedReader(inputFile);
```

```
          //Variable to hold the one line data
          String line;
          int counter=0;

          // Read file line by line and print on the console
          while ((line = bufferReader.readLine()) != null)   {
            System.out.println("Line " + (counter+1) + " is : " + line);
           // StringTokenizer st = new StringTokenizer(line);

            //Approach 1
           System.out.println("Split by comma using StringTokenizer:");
           StringTokenizer st2 = new StringTokenizer(line, ",");
           int numOfChar = 0;
           while(st2.hasMoreElements()) {
               System.out.println(st2.nextElement());
               numOfChar++;
           }

           System.out.println("Number of characters: "+ numOfChar+"\n");

           //Approach 2
           System.out.println("Split by comma using split method:");
           String[] tokens = line.split(",");
           int tokenCount = tokens.length;
           for (int j = 0; j < tokenCount; j++) {
               System.out.println(tokens[j]);
           }

           System.out.println("Number of characters : " + tokenCount+"\n");

           counter++;
          }
          bufferReader.close();//Close the buffer reader

       }catch(Exception e){
          System.out.println("Error while reading file line by line:" +
e.getMessage());
       }

    }
}
```

ReadTextFileNew.java

```
public class ReadTextFileNew {
//Process all the 4 text files

    public static void main(String[] args) {
       String[] fileNameArray = {"text1.txt", "text2.txt", "text3.txt",
"text4.txt"};
       //text1.txt (character separated by comma)
       //text2.txt (numbers separated by comma and space)
       //text3.txt ((real numbers separated by semicolon and space)
       //text4.txt (alphabets separated by numbers)
       String[] delimiterArray = {",", ", ", "; ", "\\d+"};
       for (int i = 0; i < fileNameArray.length; i++)
       {
            System.out.println("Reading File from text file : " + fileNameArray[i]
+ "\n");
            try{
               //Create object of FileReader
```

```
                FileReader inputFile = new FileReader(fileNameArray[i]);

                //Instantiate the BufferedReader Class
                BufferedReader bufferReader = new BufferedReader(inputFile);

                //Variable to hold the one line data
                String line;
                int counter=0;

                // Read file line by line and print on the console
                while ((line = bufferReader.readLine()) != null)   {
                  System.out.println("Line " + (counter+1) + " is : " + line);
                 // StringTokenizer st = new StringTokenizer(line);
                 if (i!=3) // because delimiter "\\d+" does not work for
StringTokenizer
                     tokenizerApproach(line, delimiterArray[i]);

                 splitApproach(line, delimiterArray[i]);
                 counter++;
                }
                bufferReader.close();//Close the buffer reader

            }catch(Exception e){
                System.out.println("Error while reading file line by line:" +
e.getMessage());
            }
        }
    }

    public static void tokenizerApproach(String s, String delimiter){
        //Approach 1
        System.out.println("Split using StringTokenizer:");
        StringTokenizer st2 = new StringTokenizer(s, delimiter);
        int numOfChar = 0;
        while(st2.hasMoreElements()) {
            System.out.println(st2.nextElement());
            numOfChar++;
        }
        System.out.println("Number of characters: "+ numOfChar+"\n");
    }

     public static void splitApproach(String s, String delimiter){
        //Approach 2
        System.out.println("Split using split method:");
        String[] tokens = s.split(delimiter);
        int tokenCount = tokens.length;
        for (int j = 0; j < tokenCount; j++) {
            System.out.println(tokens[j]);
        }
        System.out.println("Number of characters : " + tokenCount+"\n");
    }
}
```

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.

Sample Answer:

```
public class Exercise09_07 {
  public static void main (String[] args) {
    Account account = new Account(1122, 20000);
    Account.setAnnualInterestRate(4.5);

    account.withdraw(2500);
    account.deposit(3000);
    System.out.println("Balance is " + account.getBalance());
    System.out.println("Monthly interest is " +
      account.getMonthlyInterest());
    System.out.println("This account was created at " +
      account.getDateCreated());
  }
}

class Account {
  private int id;
  private double balance;
  private static double annualInterestRate;
  private java.util.Date dateCreated;

  public Account() {
    dateCreated = new java.util.Date();
  }

  public Account(int newId, double newBalance) {
    id = newId;
    balance = newBalance;
```

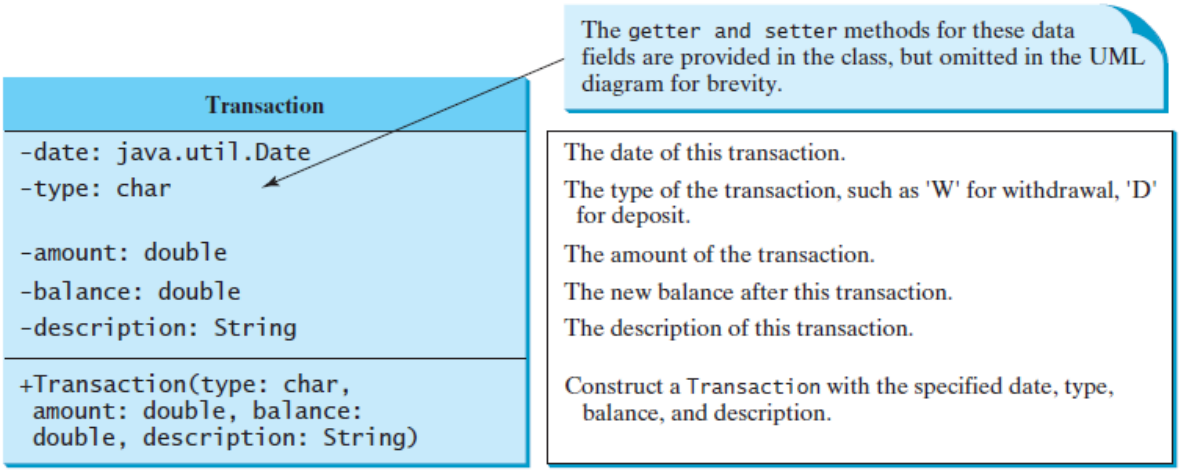
```
    dateCreated = new java.util.Date();
  }

  public int getId() {
    return this.id;
  }

  public double getBalance() {
    return balance;
  }

  public static double getAnnualInterestRate() {
    return annualInterestRate;
  }

  public void setId(int newId) {
    id = newId;
  }

  public void setBalance(double newBalance) {
    balance = newBalance;
  }

  public static void setAnnualInterestRate(double newAnnualInterestRate) {
    annualInterestRate = newAnnualInterestRate;
  }

  public double getMonthlyInterestRate() {
    return (annualInterestRate / 1200);
  }

  public double getMonthlyInterest() {
    return balance * getMonthlyInterestRate();
  }

  //public double getMonthlyInterest() {
  //  return balance * (annualInterestRate / 1200);
  //}

  public java.util.Date getDateCreated() {
    return dateCreated;
  }

  public void withdraw(double amount) {
    balance -= amount;
  }

  public void deposit(double amount) {
    balance += amount;
  }
}
```

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.



■ 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.

Sample Answer:

```
public class Exercise11_08 {
  public static void main (String[] args) {
    Account1.setAnnualInterestRate(1.65);

    Account1 account = new Account1("George", 1122, 1000);
    account.deposit(30);
    account.deposit(40);
    account.deposit(50);

    account.withdraw(5);
    account.withdraw(4);
    account.withdraw(2);

    System.out.println("Name: " + account.getName());
    System.out.println("Annual interest rate: " + Account1.getAnnualInterestRate());
    System.out.println("Balance: " + account.getBalance());

    java.util.ArrayList<Transaction> list = account.getTransactions();

    System.out.printf("%-35s%-15s%-15s%-15s\n", "Date", "Type", "Amount", "Balance");

    for (int i = 0; i < list.size(); i++) {
      Transaction transaction = (Transaction)(list.get(i));
      System.out.printf("%-35s%-15s%-15s%-15s\n", transaction.getDate(),
          transaction.getType(), transaction.getAmount(), transaction.getBalance());
```

```
    }
  }
}

class Account1 {
  private int id;
  private String name;
  private double balance;
  private static double annualInterestRate;
  private java.util.Date dateCreated;
  private java.util.ArrayList<Transaction> transactions = new java.util.ArrayList<>();

  public Account1() {
    dateCreated = new java.util.Date();
  }

  public Account1(String name, int id, double balance) {
    this.id = id;
    this.name = name;
    this.balance = balance;
    dateCreated = new java.util.Date();
  }

  public int getId() {
    return this.id;
  }

  public double getBalance() {
    return balance;
  }

  public java.util.ArrayList<Transaction> getTransactions() {
    return transactions;
  }

  public String getName() {
    return name;
  }

  public static double getAnnualInterestRate() {
    return annualInterestRate;
  }

  public void setId(int id) {
    this.id =id;
  }

  public void setBalance(double balance) {
    this.balance = balance;
  }

  public static void setAnnualInterestRate(double annualInterestRate) {
    Account1.annualInterestRate = annualInterestRate;
  }

  public double getMonthlyInterest() {
    return balance * (annualInterestRate / 1200);
  }

  public java.util.Date getDateCreated() {
    return dateCreated;
  }
```

```
  public void withdraw(double amount) {
    balance -= amount;
    transactions.add(new Transaction('W', amount, balance, ""));
  }

  public void deposit(double amount) {
    balance += amount;
    transactions.add(new Transaction('D', amount, balance, ""));
  }
}

class Transaction {
  private java.util.Date date;
  private char type;
  private double amount;
  private double balance;
  private String description;

  public Transaction(char type, double amount, double balance,
      String description) {
    date = new java.util.Date();
    this.type = type;
    this.amount = amount;
    this.balance = balance;
    this.description = description;
  }

  public java.util.Date getDate() {
    return date;
  }

  public char getType() {
    return type;
  }

  public double getAmount() {
    return amount;
  }

  public double getBalance() {
    return balance;
  }

  public String getDescription() {
    return description;
  }
}
```