**Assessment item 2**

**Assessment 2: File Processing and GUI**

**Value:** 20%

**Due Date:** 26-Aug-2018

**Return Date:** 14-Sep-2018

**Length:**

**Submission method options:** Alternative submission method

**Task**

**Task 1: Tax Management System (30 marks)**

Write a Java program that will calculate tax for the employees of a company called XYZ. The main menu of the tax calculation program is as follows.

Welcome to Tax Management System of XYZ

Please select one of the following options:  
1. Calculate tax  
2. Search tax  
3. Exit

When **1** is selected then the program will calculate the tax of an employee based on the annual income of the employee and tax rates on the income. The tax rates on the income is stored in a file called **taxrates.txt**. The program needs to read the **taxrates.txt** file and store the information in proper data structure. If the **taxrates.txt** file does not exist in the directory of the source code then the program should ask to provide the **taxrates.txt** file as an input. The format of the **taxrates.txt** file is as follows.

| **Taxable Income** | **Tax on Income** |
| --- | --- |
| 0 – $18,200 | 0 |
| $18,201 – $37,000 | 19c for each $1 over $18,200 |
| $37,001 – $87,000 | $3,572 plus 32.5c for each $1 over $37,000 |
| $87,001 – $180,000 | $19,822 plus 37c for each $1 over $87,000 |
| $180,001 and over | $54,232 plus 45c for each $1 over $180,000 |

The program will take user inputs on Employee ID (4-digit number) and the annual income of the employee (floating-point number with two decimal places). Based on the annual income of the employee the program will then calculate the tax (using the information in **taxrates.txt** file) of the employee. For  example, if the annual income of an employee is $100000.00 then the tax of the employee=19822+(100000 - 87000)\*0.37= 20303.00.

After calculating the tax of an employee the program will then write the Employee Id, taxable income and tax into a file called **taxreport.txt**. The format of the**taxreport.txt** file is as follows.

| **Employee ID** | **Taxable Income** | **Tax** |
| --- | --- | --- |
| 1111 | 100000.00 | 20303.00 |
| 2222 | 90000.00 | 19933.00 |

Once the tax calculation is done for one employee then the program will ask if XZY wants to calculate the tax for another employee, if yes then the above process will continue again. The program will calculate the tax for as many employees as XYZ wants. However, if XZY does not want to calculate the tax for another employee then main menu will be displayed.

When **2** is selected then the program will search the tax for an employee using the employee id in the **taxreport.txt** file. However, if the **taxreport.txt** file does not exist in the directory of the source code then the program should ask to provide the **taxreport.txt** file as an input.

If the**taxreport.txt** file contains the multiple tax for the same employee then the program will get the latest tax amount of that employee. If the **taxreport.txt** file does not contain the employee id then the program should give an warning message that the **taxreport.txt** file does not contain the tax of that employee.

Once searching tax (based on employee id) is done for one employee then the program will ask if XZY wants to search tax for another employee, if yes then the above process will continue again. The program will search tax for as many employees as XYZ wants. However, if XZY does not want to search tax for another employee then main menu will be displayed.

 When **3** is selected then the program will exit.  
 File read and write operations need to be done properly. You need to use proper data structure. Input validation also needs to be done.