SUBDOMAIN 430.1 - INTRODUCTION TO PROGRAMMING

Competency 430.1.1: Control Structures - *The graduate develops working programs that use appropriate control structures and accurately evaluates execution paths in program code.*
Competency 430.1.2: Algorithm Design and Development - *The graduate designs and develops algorithms for problem solving and implements those algorithms using appropriate program code.*
Competency 430.1.3: Use of Data Structures - *The graduate develops working programs that use appropriate data structures for problem solving.*

Introduction:

As a competent programmer, your understanding of control structures, your ability to design and develop algorithms, and your proficient use of data structures will help you to design and develop applications to meet customer requirements. You will need to develop a Java application to meet the requirements of this assignment. This assignment requires a graphical user interface (GUI).

Note: It is recommended that you use NetBeans to develop this application and the NetBeans GUI Builder to develop your GUI. To download this program, follow the instructions found at the web link below.

You will use the following scenario to complete the task:

A friend has requested that you develop an application that will help keep track of earnings from a tutoring business. A graphical user interface (GUI) is required to ensure the application is user friendly. The application will accept earnings information. The user will enter minutes and earnings per each tutoring session. Upon request, a report can be generated providing average earnings per session, total tutoring minutes provided, and total earnings to date. The application should be well commented to make it easy to maintain.

Task:

Note: You will submit all .java and .class files in one zipped folder. If you developed this application using NetBeans, it is recommended that you zip and submit the entire NetBeans project folder.

A. Create a program that has a GUI and calculates earnings and tutoring time by doing the following:
   1. Create a GUI that allows a user to enter the following into labeled textfields:
      - Number of minutes for tutoring session (should be a positive decimal number)
      - Earnings in dollars and cents received (should be positive decimal number)
      a. GUI should be built using javax.swing components jButton, jLabel, jTextField, jTextArea, jFrame, and jPanel.
   2. Implement functionality for *each* of the following application buttons:
      - "Quit" button: Should exit application
      - "Enter" button: Accepts entered number of minutes and earnings; as earnings information is entered, it is stored in a two-dimensional array
      - "Run Report" button: Calculates total minutes of tutoring provided, average wages earned per hour, total earned to date, and determines if average wage per hour is below, average, or above minimum wage

Note: For an example, see the attached "Sample GUI."

3. Use a two-dimensional array to store the entered minutes tutored and earnings received information for *each* tutoring session, as illustrated in the attached "Two-Dimensional Array."
4. Implement the "Run Report" button to iterate through the two-dimensional array using a nested for-loop to display the earnings data to the jTextArea, as illustrated in the attached "Earnings Entered."
5. Implement the "Run Report" button to calculate the following:
   - Total minutes tutoring provided (as a decimal number)
   - Average wage per hour (in dollars and cents)
   - Total earnings (in dollars and cents)
   - Wage analysis, determined based on comparison of average per hour earnings and current minimum wage, using the following logic:
     - if average wages per hour is < minimum wage, then below average
     - if average wages per hour is ≥ minimum wage AND ≤ minimum wage × 2.00, then average
     - if average wages per hour is > minimum wage × 2.00, then above average
6. Display resulting calculations in GUI text area in the required format, as illustrated in the attached "Calculations Display."

   Note: All results from calculations should be viewable in the text area. Consider appending to the text area as you write out information.

7. Use exceptions to handle invalid user entries for the following situations:
   - If minutes entered is less than or equal to zero, throw exception.
   - Tutors should not tutor for longer than 4 hours per day. If minutes entered is greater than 60 × 4 = 240 minutes, throw exception.
   - Tutors must be paid for their tutoring service. If earnings entered is not greater than or equal to zero, throw exception.

   Note: The program should not terminate abruptly if any of these situations are encountered. The application should print a message "Invalid input. Please try again." and let the user continue using the application.

8. Use single-line and multiline comments to explain program flow.

B. When you use sources, include all in-text citations and references in APA format.

Note: When bulleted points are present in the task prompt, the level of detail or support called for in the rubric refers to those bulleted points.

Note: For definitions of terms commonly used in the rubric, see the Rubric Terms web link included in the Evaluation Procedures section.

Note: When using sources to support ideas and elements in a paper or project, the submission MUST include APA formatted in-text citations with a corresponding reference list for any direct quotes or paraphrasing. It is not necessary to list sources that were consulted if they have not been quoted or paraphrased in the text of the paper or project

Note: No more than a combined total of 30% of a submission can be directly quoted or closely paraphrased from sources, even if cited correctly. For tips on using APA style, please refer to the APA Handout web link included in the General Instructions section.