Grade 12 Assignment #6 Java Programming

Classroom Grades Analyzer

In this project, you will create a simple tool to help analyze classroom grades stored in an ArrayList. At the end of this project, the analyzer you build will be able to retrieve the classroom average.

Upon completion, feel free to explore and add additional functionality to your grades analyzer.

Tasks:

There are 23 Tasks to complete for this assignment:

1. Open the BlueJ IDE editor to create the code needed for the program called Classroom Grades Analyzer.

2. First set up the BlueJ IDE. Create a project (file) with a public class called gradeAnalyzer.

3. Import the ArrayList (function) class. To use the ArrayList function, it must first be imported. The ArrayList class is part of a Java package. All imported

functions must occur at the top of the program before the public class declaration. Import the correct package by typing:

```
import java.util.ArrayList;
```

Note: Java provides built-in classes (pre-defined functions that can be used). Some are readily available (like the System.out.println() and math operators +-&≤÷=) but that does not mean they are ALL readily available in ALL Java programs. Many classes are stored in Libraries called Java packages. The correct Java package must first be imported into a Java program before a certain class (function) can be used.

4. Next, create a class called GradeAnalyzer, under the import declarations

5. Now create a GradeAnalyzer constructor. You can leave the contents of the constructor empty.

6. Let's create a method that will return the average of all grades. Create a method called getAverage. It should return an int.

7. The getAverage method should accept an ArrayList parameter that holds integers. Call the parameter grades.

   Hint: the parameter is written as: ArrayList<Integer> grades.

8. The first thing the method should do is check to see that the ArrayList it's analyzing is not empty. Create an if statement that checks if the size of grades is less than 1.

9. Inside of the if block, print out a friendly error message to the user indicating that the ArrayList is empty. On the next line, return 0.

10. Otherwise, in an else block, find the average of all grades. To find the average, first we will need the sum of all grades. Create an int called sum and set it equal to 0. We will update this variable as we sum the grades.

11. Create a for each loop block that iterates through each grade in the grades ArrayList.

12. Inside of the for each loop block, update sum. Set it equal to sum plus grade. This for each loop will add up all the grades.

13. Outside of the for each loop, calculate the average of the grades. Create an int called average and set it equal to the sum divided by the size of grades.

14. On the next line, print out the average.

15. Finally, on the next line, return the average.

16. Next, create the main method for this program. This is the main body of programming code that is executed when run. Note: the main method must be defined *exactly the same way* every time it is created. Refer back to the lesson if you need to review the main method.

In this assignment, the main part of the code is extremely simple. It will focus on creating an array then adding elements to that array. Then creating an object that belongs to the class through the class constructor. Once an object is created, that object can call the methods (specialized functions) that belong to the class. The main part of the code will simply call the main method and print out the results to the screen.

17. Inside of main create an ArrayList object that stores integers and call it myClassroom.

18. Next, add the grades 98, 92, 88, 75, 61, 89 and 95 to myClassroom.

19. On the next line, create a GradeAnalyzer object called myAnalyzer.

20. Now call the getAverage method on myAnalyzer and specify myClassroom as the argument (parameter).

    If you completed this project correctly, the output should show a class average of 85. Feel free to explore more with the program.

    Feel free to explore more with the program. What are some ways in which the program could be improved?

    For example, write methods that determine both the lowest or highest class grade.

Also, as the program is right now there are no enough printed out statements to the screen (System.out.println()). As is the user only sees an 85 and has no idea what is going on. This is not acceptable.

21. It would be helpful to describe to other developers what this small Java program does. Write some comments that describes what this program does.

    a. Use multi-line comments to (/* comment in the middle of */):
        i. Write one at the top of the code (before the public class and import designations) that gives a quick intro/description the assignment.
        ii. Write one at the top of the code (before the public static void main(String[] args)) that summarizes the program.
    b. Use a single line comment to (//then comment):
        i. Create 3 comments anywhere you deem necessary or important in the code. Remember the comment is to highlight or explain what is going on or what is being done in a particular way or used and why.
        ii. Identify each of the Methods created by indicating its purpose.
        iii. Explain why the class constructor GradeAnalyzer is Empty (no code inside the block).
        iv. Identify the last line of code (anything that ends with a curly bracket }) in every function by writing “End of BLAH-BLAH function”.
        v. Identify the end of the program.

22. Once your program is complete, make sure to test it using the BlueJ IDE (do not submit a program that does not work).

23. Lastly, upload (drag and drop) your assignment to the portal. Look for your name under the Assignment 6 webpage.