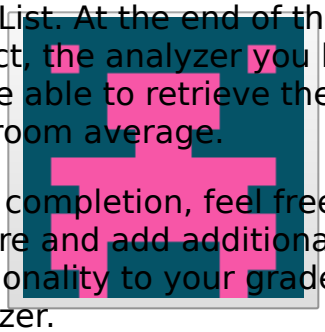
Objective
Learn Java

Classroom Grades Analyzer

Learn Java

- Upgrade to Pro
- Connected to Codecademy



Drag the edges to resize the window.

Classroom Grades Analyzer

2/21 Complete

Up Next

Get Help

Tasks
2/21 Complete
Mark the tasks as complete by checking them off

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.

If you get stuck during this project, check out the **project walkthrough video**, which can be found at the bottom of the page after the final step of the project.

In Projects, you can keep track of your progress as you go throught the tasks. Check each item as you complete it.

1.

Note: Java provides built-in classes, but that does not mean they are readily available in all Java programs. Some classes are stored in Java packages. The correct Java package must first be imported into a Java program before a certain class can be used.

The `ArrayList` class is part of a Java package. On line 1, import the correct package by typing:

GradeAnalyzer.java

```
xxxxxxxxxx
11

1
import java.util.ArrayList;
2
class GradeAnalyzer {
3


4
  public GradeAnalyzer() {
5


6
  }
7


8
  public static void main(String[]args) {
9


10
  }
11
}
```

Save

Reset Project

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

2.

Next, create a class called `GradeAnalyzer`.

3.

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

4.

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

5.

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

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

6.

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`.

7.

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`.

8.

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.

9.

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

10.

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

11.

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`.

12.

On the next line, print out the average.

13.

Finally, on the next line, return the average.

14.

Next, create the `main` method for this program.

15.

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

16.

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

17.

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

18.

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

19.

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

Optional: Write methods that determine both the lowest or highest class grade.

20.

Add comments near the top of the program that describe what the `GradeAnalyzer` program does.

21.

Click here for a video walkthrough from our experts to help you check your work!

Report a Bug
If you see a bug or any other issue with this page, please report it here.

Objective
Tasks
2/21 Complete
Report a Bug
Objective
Tasks
2/21 Complete
Report a Bug