

# CS101- Algorithms and Programming I

## Lab 09

---

**Lab Objectives:** ArrayLists. Classes and objects.

---

For all labs in CS 101, your solutions must conform to the CS101 style guidelines (rules!)

### PART 1:

1. Write a Java program that inputs a set of integer values from the user and stores them in an ArrayList. Create a new ArrayList that is the same length as the original list, where each element in the new list contains the sum of all elements in the original list, up to the current element, which are divisible by a number input by the user.

Your program should define and make use of the following helper methods:

- o `void removeDuplicates(ArrayList<Integers>):` removes duplicates from the parameter ArrayList
- o `ArrayList<Integers> fillList():` fills and returns an ArrayList with values input by the user.
- o `void sumDivisible(ArrayList<Integers>, ArrayList<Integers>,int):` takes 2 ArrayLists as parameters, the first list contains a set of Integer values, the second list should be updated to contain the sum of the elements divisible by the int value passed as a parameter.

### Sample Run:

Enter values:

4 16 15 25 230 27 34 19 5 10 6 \*

Enter divisor: 5

List: [4, 16, 15, 25, 230, 27, 34, 19, 5, 10, 6]

Sum List: [0, 0, 15, 40, 270, 270, 270, 270, 275, 285, 285]

Sum List (no duplicates): [0, 15, 40, 270, 275, 285]

## PART 2:

1. Download from Moodle the classes, `Player.java` that stores information about a Player (name, games played, high score), and the class `FlashCard.java` that contains data and functionality related to `FlashCard` objects. Review the functionality of these classes. You may want to create objects/test methods to see how they work.
2. Create a class, `FlashCardQuiz.java` that has the following data and methods:

### Data Members:

- `public static Scanner` to read keyboard input. Use this Scanner for all keyboard input in class and in application.
- `ArrayList, cardList`, to store the list of all `FlashCards`.
- `ArrayList, quizList`, to store `FlashCards` for the current quiz.
- A `Player` representing the Player of the quiz.

### Methods:

- **Constructor:**
  - takes a file name and a `Player` as parameters, initialize the `Player` to the one passed as a parameter.
  - Calls the method `loadCards()` to load the data from the file into the `cardList`. (see below)
  - Calls the `initializeQuiz()` method to initialize a `FlashCardQuiz`. (see below)
  - **Note:** because the Constructor calls the `loadCards()` method that reads data from a file, after the method declaration you must include: `throws FileNotFoundException`
- **loadCards():** takes a string filename as a parameter and loads the `FlashCards` from the file into the `cardList ArrayList`. Because method reads data from a file, the method must also throw `FileNotFoundException`.
- **initializeQuiz() :** prompts user to input the number of `FlashCards` they would like the quiz to have. Then choose n random cards from the `cardList` and add to the `quizList`. There should be no duplicate questions in the `quizList`.
- **takeQuiz() :** using `FlashCard` functionality, displays the `FlashCards` one by one, and prompts player for the answer. Each correct answer earns 1 point (you can change the scoring if you like). Display the correct answer for all incorrect guesses. At the end of the quiz, display the result and update the `Player` data (games played/high score).

3. Create a class, `FlashCardApp.java`, that does the following:
  - Creates an `ArrayList` of `Players`.
  - Displays a menu, prompting the user to take the quiz or view the player information.
  - If the user chooses to Take Quiz, a `FlashCardQuiz` is created. You may use data from the sample file: "turkish\_english\_words.txt". Because of this the main method must also throw a `FileNotFoundException`.
  - Each `FlashCardQuiz` has a `Player`. `Players` are uniquely identified by their player names. If the `Player` who wishes to take the quiz already exists in the `players ArrayList`, they should be added to the `FlashCardQuiz` as the `Player`, otherwise the new `Player` may be used to initialize the `FlashCardQuiz`. All new `Players` should be added to the list of `Players`.

## NOTES:

- Opening and closing Scanners for keyboard input in multiple classes can cause runtime exceptions. To avoid these exceptions, you have declared a public, **static** Scanner object in your `FlashCardQuiz` class. In your application you should initialize this Scanner object (`FlashCardQuiz.in = new Scanner(System.in)`), and close this object before the application terminates. You should also use the same Scanner in the `FlashCardQuiz` class.
- Below is a sample run of the application. You may use the classes above to develop your own custom application as well. For example, you may wish to generate quizzes according to question difficulty, you may choose to score your quizzes in different ways, you may choose a different file with different questions, etc. You may wish to display English words and have the user input their Turkish translations.

## Sample Run:

```
1 - Play Game
2 - View Player Information
3 - Quit
Choice: 1
Enter player name: Lori
Enter the number of flash cards to generate: 4
*****
*               *
*   insan      *
*               *
*****

Answer: human
Correct!
*****
*               *
*   ticaret    *
*               *
*****

Answer: business
Wrong Answer
*****
*               *
*   trade      *
*               *
*****

*****
*               *
*   varmak     *
*               *
*****

Answer: arrive
Correct!
```

```
*****
*
*      gecmek      *
*
*****
```

Answer: pass

Correct!

Your Score: 3/4

1 - Play Game

2 - View Player Information

3 - Quit

Choice: 1

Enter player name: Lori

Enter the number of flash cards to generate: 5

```
*****
*
*      baglamak    *
*
*****
```

Answer: tie

Wrong Answer

```
*****
*
*      connect     *
*
*****
```

```
*****
*
*      bu           *
*
*****
```

Answer: this

Wrong Answer

```
*****
*
*      that      *
*
*****
```

```
*****
*
*      urun      *
*
*****
```

Answer: product

Correct!

```
*****
*
*      karsilastirma      *
*
*****
```

Answer: compare

Correct!

```
*****
*
*      aglamak      *
*
*****
```

Answer: cry

Correct!

Your Score: 3/5

1 - Play Game  
2 - View Player Information  
3 - Quit  
Choice: 2

Player Name: Lori

Games Played: 2

High Score: 3

1 - Play Game  
2 - View Player Information  
3 - Quit  
Choice: 1

Enter player name: Ali

Enter the number of flash cards to generate: 4

```
*****
*           *
*    log    *
*           *
*****
```

Answer: ?

Wrong Answer

```
*****
*           *
*    log    *
*           *
*****
```

```
*****
*           *
*   lutfen  *
*           *
*****
```

Answer: please

Correct!

```
*****
*           *
*   dogmus  *
*           *
*****
```

Answer: born

Correct!

```
*****
*           *
*   toplam  *
*           *
*****
```

Answer: total

Correct!

Your Score: 3/4

1 - Play Game

2 - View Player Information

3 - Quit

Choice: 2

Player Name: Ali

Games Played: 1

High Score: 3

1 - Play Game

2 - View Player Information

3 - Quit

Choice: 3