

# SWE 206

## LAB 1

**Lab topic: Introduction**

**Section No: 52**

**Groups members:**

**Ahmed Almakhlouq 202017740**

**Ridah Al-Moslem 202017940**

## REQUIREMENTS:

- The program should generate groups with random students.
- Create groups with a specific size.
- Create specific numbers of groups.
- Add the rest of the students to a randomly chosen group.

## DESIGN:

- Initialize an empty ArrayList to have the names.
- Read the file and add it to the ArrayList.
- Create a 'fixedSizeGroups' method that has our 'ArrayList' and the size of each group as an argument.
- Create a 'fixedNumberGroups' method that has our 'ArrayList' and the number of groups as an argument.

## TESTING:

**CHOICE 1: CREATING A SPECIFIC NUMBER OF GROUPS.**

**CHOICE 2: CREATING GROUPS WITH A FIXED NUMBER OF STUDENTS.**

```
Enter the path of the list file: U:\\Term222\\ICS202-SWE206-ISE291-MATH208-GS332\\SWE206\\SWE-206\\Labs\\Lab1\\List.txt
Enter the way of distributing students:
  1) By a fixed number of groups.
  2) By a fixed maximum students.
Choice: 1
Enter the number of groups: 6
[[Ismail, Jasser], [Ahmad, Gamal], [Hamad, Naif], [Mohammed, Kamal], [Faisal, Bader], [Dawwd, Qassim, Rabeh, Sami, Tala, Yusuf]]
PS C:\\Users\\hamad> & 'C:\\Program Files\\Java\\jre1.8.0_351\\bin\\java.exe' '-cp' 'C:\\Users\\hamad\\AppData\\Local\\Temp\\vscode\\_86463\\j
Enter the path of the list file: U:\\Term222\\ICS202-SWE206-ISE291-MATH208-GS332\\SWE206\\SWE-206\\Labs\\Lab1\\List.txt
Enter the way of distributing students:
  1) By a fixed number of groups.
  2) By a fixed maximum students.
Choice: 2
Enter the maximum number of students: 5
[[Sami, Qassim, Jasser, Naif, Hamad], [Kamal, Yusuf, Gamal, Ahmad, Tala], [Rabeh, Bader, Ismail, Mohammed, Dawwd], [Faisal]]
```

# IMPLEMENTATION:

Main method:

```
public static void main(String[] args){
    try{
        Scanner input = new Scanner(System.in);    Resource leak: 'input' is never closed
        System.out.print("Enter the path of the list file: "); // Enter the path.
        String fileName = input.next();
        File students = new File(fileName); // "U:\Term222\ICS202-SME206-ISE291-MATH208-GS332"\SME206\SME-206\Labs\Lab1\List.txt"
        Scanner reader = new Scanner(students);    Resource leak: 'reader' is never closed
        ArrayList studentList = new ArrayList<>();
        while(reader.hasNext()){
            String student = reader.next();
            studentList.add(student);
        }
        System.out.print("Enter the way of distributing students:\n 1) By a fixed number of groups.\n 2) By a fixed maximum students.\nChoice: ");
        int way = input.nextInt();
        switch (way) {
            case 1 :
                System.out.print("Enter the number of groups: ");
                int groupsNumber = input.nextInt();
                System.out.println(fixedNumberOfGroups(studentList, groupsNumber));
                break;
            case 2 :
                System.out.print("Enter the maximum number of students: ");
                int maxSize = input.nextInt();
                System.out.println(fixedSizeGroups(studentList, maxSize));
                break;
        }
    }
    catch(FileNotFoundException ex){
        System.out.println(ex.getMessage());
    }
}
```

fixedSizeGroups method:

```
// Creating groups with a fixed number of students.
public static ArrayList<ArrayList<String>> fixedSizeGroups( ArrayList<String> names, int size){
    int listSize = names.size();
    ArrayList<ArrayList<String>> groupsArray = new ArrayList<ArrayList<String>>();
    while (!names.isEmpty()){
        ArrayList<String> group = new ArrayList<>();
        for(int z = 0; z < size ; z++){
            int random = (int)(Math.random() * listSize);
            group.add(names.get(random));
            names.remove(random);
            listSize--;
            if (names.isEmpty()){
                break;
            }
        }
        groupsArray.add(group);
    }
    return groupsArray;
}
```

fixedNumberOfGroups method:

```
// Creating fixed number of groups.
public static ArrayList<ArrayList<String>> fixedNumberOfGroups( ArrayList<String> names, int numOfGroups){
    int listSize = names.size();
    int numberOfStudentInEachGroup = 0;
    if(listSize % numOfGroups == 0){
        numberOfStudentInEachGroup = listSize / numOfGroups;
    }
    else{
        while(listSize % numOfGroups != 0){
            listSize--;
        }
        numberOfStudentInEachGroup = listSize / numOfGroups;
    }

    ArrayList<ArrayList<String>> groupsArray = new ArrayList<ArrayList<String>>();
    for(int z = 0; z < numOfGroups - 1; z++){
        ArrayList<String> group = new ArrayList<>();
        for(int i = 0; i < numberOfStudentInEachGroup; i++){
            int random = (int)(Math.random() * listSize);
            group.add(names.get(random));
            names.remove(random);
            listSize--;
        }
        groupsArray.add(group);
    }
    groupsArray.add(names);
    return groupsArray;
}
```