**OOP Lab Test 1**

**Rules:**

* This is an individual test. No collaboration or communication with other students during the test.
* You can use the following websites during the test. <https://www.processing.org/reference/> <https://github.com/skooter500/DT228-OOP>. No use of Google/Facebook or any other external resources.
* At the end of the test, zip up the folder with your solution and submit through webcourse only. Do not email me solutions.

In this test, you will be loading a dataset of politician’s expenses and writing a processing sketch to analyse and graph the data. The data is tab separated and formatted as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Party** | **Constituency** | **Amount returned** | **Amount claimed** |
| Buttimer, Jerry | FG | Cork South-Central | 586.14 | 50645.3 |
| … |  |  |  |  |

Create a new Processing sketch and save it. If your student number is D111111 and your name is Alan Moore, your sketch should be called:

D111111\_Alan\_Moore

You can get the file you need to complete the test from here:

<https://raw.githubusercontent.com/skooter500/DT228-OOP/master/Processing%20Examples/Expenses/data/expenses.txt>

This should be saved in a folder called **data** (note the lower case) inside your sketch folder.

1. Create a class called **Expense** with appropriate fields, default and parameterised constructors. You can use floats for the currency values. Add an additional field called total of type float. This class will be used to encapsulate a single row of data from the file. Also add a field called colour of type color.

(10 marks)

1. Create a class called **PartyExpense** with a field for the party name and a field for total. Also add a field called colour of type color.

(10 marks)

1. In your sketch file, create two global variables called **expenses** and **partyExpenses**. These should be ArrayList’s of Expense and PartyExpense types.

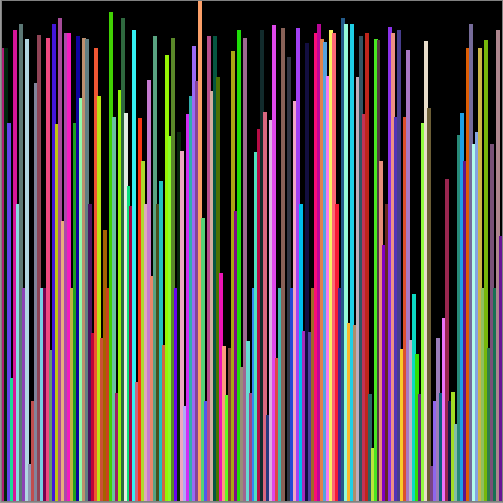
(5 marks)

1. Write a method called **loadExpenses.** This method should populate the two array lists from the file of data. The total field in the expense class is calculated as being amount claimed – amount returned. Assign a random colour to each Expense object and PartyExpense object. Hint: You will need to compare strings to see if a matching PartyExpense object exists in the partyExpenses array list. The best way to do this is by using the equals method on the String class:

if (expense.party.**equals**(partyExpenses.get(j).name))

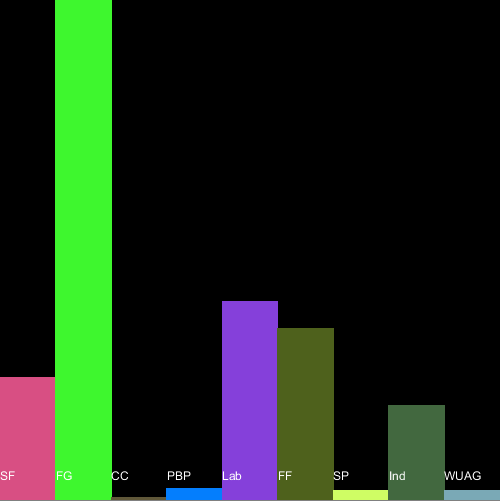
(30 marks)

1. Write a method called **drawExpenseBars** thatdraws abar chart of the expenses array list. This graph should fill the width and height of the sketch. It should look something like this:



(20 marks)

1. Write a method called **drawPartyExpenseBars** thatdraws abar chart of the partyExpenses array list. On this graph you should print the party name. This graph should fill the width and height of the sketch. It should look something like this:



(20 marks)

1. Write code to switch between displaying the two graphs in the sketch when you press 0 and 1.

(5 marks)