**Te Hoe Rorohiko**

Department of Computing

Bachelor of Information and Communication Technologies

Graduate Diploma of Information and Communication Technologies

BCSE 101 Software Engineering

Assignment THREE

**Programming Assignment**

Semester Two 2017

Due date: Friday 10 November 2017

Time: 5.00pm

Student Name/ID



CPIT and its faculty members reserve the right to use electronic means to detect and help prevent plagiarism. Students agree that when submitting this assignment, it may be subject to submission for textual similarity review to Turnitin.com.

This assignment is worth 25% of the total marks for BCSE101.

This paper has three (3) pages including the cover sheet.

TASKS

The goal for this assignment is to extend the previous assignment so that it calculates the winner of an electorate seat and analyses voting trends.

1. **Code Comparison or UML Diagrams [10 marks]**

*Either*

* Compare your code with the model answer.
* Put YOUR code through https://standardjs.com/demo.html

What got warning and what was fixed?

NOTE: You will need to add the following comments at the top each file to configure the warnings

/\* global Election, View \*/

***Exact globals to use varies with file***

*Or*

* If you did not submit a first programming assignment which worked up to getElectorates then you must:
  + draw a UML class diagram of the provided model answer for the previous assignment. In this diagram show:
    - class names
    - attribute names
    - attribute type see <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures>
    - method names
    - relationships
    - relationship type arrowhead
    - relationship name
  + draw a UML sequence diagrams of getElectorates in the provided model answer for the previous assignment (start at Election and follow the logic down to Electorate). In this diagram show:
    - objects
    - lifelines
    - activations
    - messages to other objects
    - reflexive messages
    - loops
    - loop conditions

You may use <https://www.websequencediagrams.com/>

1. **Starting Analysis [10 marks]**

**You have: Electorates with links to their winning MP and their Party, Parties with their Party Lists and total Party Vote.**

* Draw a ***UML*** ***dynamic diagram*** showing how a seats will be allocated to party list members.
* Draw an ***analysis level class diagram*** of what classes and attributes and methods are needed to support **allocating seats to Parties.**
  + In the diagram show only classes (boxes with names), relationships (lines), multiplicities (0s, 12, and \*s) and new methods (names only in the bottom of the class boxes).

1. **Implement** Election.allocateSeats **[40 marks]**
2. **Implementation of New Reports [40 marks]**

Who won?

* Produce an Election status display which shows how many seats each Party gets.
  + Produce a display that shows which list candidates get to be MPs.

What happened?

* + Analyse Electorates comparing 2014 and 2017. Which stayed with the same Party? Which changed Party?
  + Analyse Party Vote by Electorate? Which got more Red/Blue? Which got less Red/Blue?