School of Computing & Information Technology

CSCI251/CSCI851 Advanced Programming Autumn 2022

Assignment 2 (Worth 1% (Part A) + 13% (Part B))

Due 11:55pm Friday in Week 10

Overview

This assignment is to be implemented using object oriented programming. It involves implementing a simulation of an election campaign and the election itself.

In addition to providing code, you need to submit a final report, which contains a final UML like class diagram reflecting the structure in your code, and text/tables/diagrams addressing the points below:

1. Describe qualitatively, so not numerically, the five (5) different issues that you have chosen.
2. Describe the three (3) political parties and specify the range of stances on each issue.
3. Describe the characteristics of your electorates, and how the stance distribution for the electorates is modelled.
4. The winner for an electorate is the candidate for that electorate who obtains the most votes. You need to describe how the number of votes for each candidate is obtained. There should be randomness in this process. While the process doesn't need to be particularly complicated, there should be dependence on the relative stances of electorates and candidates and on candidate popularity.
5. Describe the characteristics of the candidates and the qualitative impact they have.
6. Describe the characteristics of the other people associated with the political parties and the qualitative impact they have.
7. Describe what events you have chosen and how the event mechanisms qualitatively depend on the characteristics of the components described above.

General notes

These are some general rules about what you should and shouldn't do.

1. Your assignment should be sensibly organised with the same kind of expectations in this area as assignment one, although you may reasonably have more files for this assignment.
2. **Other than the initial command line input, the program should run without user input. In particular this means there shouldn't be pauses waiting for the user to press a key.**
3. Your code must compile on CodeBlocks or Capa with the compilation instructions you provide in `Readme.txt`.
4. You have to use classes, and should make use of encapsulation.
5. You must use inheritance. If you don't use inheritance you will lose a fair few marks.
6. You can use polymorphism, if it's appropriate.
7. You use operator overloading, it should be appropriate.
8. You can use your own data files to import for names or similar information.
9. Output should only be to standard out or standard error.

1 A political episode

The nation is on edge ahead of the upcoming election campaign. The leaders have presented their candidates and published manifestos detailing their plans for the country. You need to design and build a simulation of the campaign and subsequent election.

Your code should compile on CodeBlocks or Capa according to instructions you provide in a `Readme.txt` file.

If you run on Capa, once your program is compiled into the executable `APE`, it must run as follows:

```
./APE n m
```

where n is the number of electorates in the nation, and m is the number of campaigning days before the election. The value of n should be in the range 1 to 10 inclusive. The value of m should be in the range 1 to 30 inclusive.

On running, `APE` should report on each party, so their stance values and the relevant details of their candidates, leader, and other significant figures.

`APE` should also report on the state of the nation, so give details on the electorates including the stance distribution.

2 Components

You need to give consistent and appropriate definitions for a class hierarchy associated with the various components of the campaign and election.

2.1 Issues

You need to decide on five (5) issues of relevance, so concerns, for the nation. These can be real-world national issues, such as global warming, real-world local matters, such as road infrastructure, or not–so–likely issues, such as the distribution of cat food. For simplicity, we model a stance on an issue as being two–dimensional, with the dimensions being:

1. The significance of the issue.
2. The approach is taken to the issue. You don't need to specify what the approach means, and it's just a scale where similar values mean similar approaches and quite different values represent pretty different approaches.

Each party and electorate will have different stances on each. The leaders and candidates share the same stance with their parties. Events may change stance values.

2.2 Political parties and candidates

Each political party has the following officials:

1. One leader, not themselves a candidate for an electorate.
2. One candidate for each electorate.
3. One managerial team.

Each political party has a range across each stance. You define this for each of your three political parties. Initial stance values should be randomly generated somewhere in that range, both for the party, party candidates, and the party leader.

Both the leader and the candidates will have popularity and at least one other characteristic. It's up to you to determine how those are relevant, but both popularity and the other characteristic(s) must have some impact.

2.3 Electorates

You need to determine appropriate components to represent the electorates. Each will have a political stance distribution rather than a specific stance. It's up to you to decide how this works. If you choose to trivialise it to being a single stance, you still get full marks for this part for the candidates and parties. Using distribution will score additional marks.

2.4 Campaign days and events

Each day of the campaign involves the candidates and leaders attempting to convince the electorates to support them.

On every day of the campaign, for each electorate, there is a 50% chance of a local event:

1. Debate.
2. Candidates related event. You need two different types of these events.

3. Leaders related event. You need two different types of these events.

4. Issues related event. You need two different types of these events.

Leaders, candidates, and the managerial team will have some impact on the outcomes.

It is up to you to determine the likelihood of individual events, but they should be weighted towards less significant events. You should describe the events and their impact in your report.

Each campaign day `APE` should report the day, what has happened in each electorate, and the significant impacts that have occurred.

2.5 Election day and wrapping up...

Immediately before election day, you should report on each party, so their stance values and the relevant details of their candidates, leader, and financial figures.

You also need to report on the state of the nation, so give details on the electorates, including population and stance distribution.

For each candidate, you need to provide the breakdown of votes and identify the winner so the candidate with the most votes.

You then need to provide a national summary so how many electorates have been won by each party. If a party has won more than 50% of the electorates, they will form a government, and you can report the party leader as the new leader for the country. If no party has won more than 50% of the electorates, `APE` should declare a hung parliament.

Notes on submission

Submission is via Moodle.

Your code must compile on Capa or CodeBlocks with the instructions you provide.

Please submit your source, so .cpp and .h files, Readme.txt file, any data files used, report, and makefile if you have one, in a zip file `name_studentID_A2.zip`.

Make sure your report is in pdf and has your name, student number, and lab class marked clearly on it. The final diagram should be in your final report pdf and not in a distinct document. It may be ignored if it's a non–pdf format.

1. Late submissions will be marked with a 25% deduction for each day, including days over the weekend.

2. Submissions more than four days late will not receive marks, unless an extension has been granted.

3. If you need an extension apply through SOLS, if possible **before** the assignment deadline.

4. Academic misconduct is treated seriously.