

CS-150 2023

4th Assignment - Live

05/05/2023

18:00 – 20:00

Using the rules of inheritance implement a program in which users have to select a political party and a candidate to vote for.

1. *Party Classes (50%)*

Implement 2 subclasses of the `PoliticalParty` class, which is provided below. The two subclasses should be called `BlueParty`, and `RedParty`, each of them represents a different electoral party. Also, two structs, `Info` and `Nominee`, are provided, where the information of each party and each candidate is stored. Each subclass should be able to assign values to the variables **partyInfo**, **Name**, **Nominees**, through its constructor.

The `PoliticalParty` class contains 2 virtual functions (**GetTotalScore**, **GetNomineesVotes**) that work differently in each subclass.

- For `BlueParty`, **GetTotalScore** calculates the total number of votes divided by the total number of candidates. While for `RedParty` it counts the total number of votes of all candidates.
- For `BlueParty`, **GetNomineesVotes** returns all candidates and the number of votes they have sorted in *ascending* order based on the name, while for `RedParty` **GetNomineesVotes** returns the same information sorted by the votes of each candidate, in ascending order.

```

class PoliticalParty{
protected:
    Info partyInfo;
    string Name;
    vector<Nominee> Nominees;

public:
    virtual int GetTotalScore() =0;
    virtual vector<string> GetNomineesVotes() = 0;

    Info GetPartInfo(){
        return partyInfo;
    }

    void SetPartyInfo(Info partyInfo){
        this->partyInfo = partyInfo;
    }

    void SetNominees(vector<Nominee> noms){
        Nominees = noms;
    }

    void IncreaseVote(int idx){
        Nominees[idx].Votes++;
    }

    string GetAllInfo(){
        //Add code here
    }
};

```

```

struct Info{
    int foundedYear;
    string chairmanName;
    string Ideology;
};

struct Nominee{
    string Name;
    int Votes;
};

```

2. *Init & Run (50%)*

The program should initially create a RedParty object and a BlueParty object. It will then read from the RedParty.txt and BlueParty.txt files the available candidates and their current votes and store the information in the vector Nominees. It will then ask the user to select a political party and will print out all the information that is provided from the **GetAllInfo()** function. Finally, the user will choose one of the candidates. The program ends when the user gives the value -1.

The GetAllInfo() function should return a string in the following format:

```
Political Party: RedParty  
Founded: 1955  
ChairmanName: Giorgos Spirakis  
Ideology: We like red.  
Score: 15  
Nominees:  
[0]: Papadimitropoulos Michalis 12  
[1]: Lefterakias Lefteris 8  
....
```

Instructions

Deliver all the code files (.h, .cpp) you made for the exercise and optionally CMakelists.txt, all through the elearn submit, as well as all previous exercises so far.

It is mandatory in classes that declarations and definitions are in separate files.