Documento anónimo

Parcial2Solution2018.pdf

EXAMENES

3° Informática Industrial I

Grado en Ingeniería Electrónica Industrial y Automática

Escuela Politécnica Superior
Universidad Carlos III de Madrid

montero espinosa
Academia Universitaria

academia universitaria

En nuestra academia TE OFRECEMOS:

LOS MEJORES PROFESORES

Son los protagonistas: Calidad, preparación y motivación es lo que define a nuestro equipo. Ingenier@s y licenciad@s se ciñen extrictamente a los contenidos de la Uni, y lo mejor, van al grano.

CERCA DE TI

Nos ubicamos cerca de las Universidades, para que no pierdas tiempo.

QUE EL RITMO NO PARE

Todas nuestras clases se imparten tanto presenciales como online.

A NUESTROS APUNTES, LOS QUIERE TODO EL MUNDO

Los más completos, claros y ordenados apuntes de teoría.
También tenemos la mayor colección de ejercicios de examen resueltos.

Montero Espinosa, patrocinador oficial de tus aprobados.

¡Infórmate! 689 71 67 71

Exercise 1 – 4 points

Given the screen output shown at the end of the text, complete as required in the comments of the code (it could be a fragment of code, a reserved word or nothing at all). The strings shown in the output are printed by the methods which include them as required in provided code. You are required to implement all the functions shown in the declaration.

contest.h

```
#include <iostream>
using namespace std;

struct Candidate { // [1] Candidate structure
   string name;
   int points;
};

class Contest {
public:
   Contest(_____){ // [2] Parametrized constructor
      // [3] Initialize class attributes

      // [4] Allocate required space for candidates

      // [5] Initialized candidates list with received names

   }

   _____ showWinner(_____);// [6] Method to compute and announce the winner

   _____ voteFor(_____); // [7] Method to add votes to candidates

   _____ operator += (_____); // [8] Operator +=

   _____ operator << (_____); // [9] Operator <<

   ~Contest();// [10]  Destructor

private:
   string _name;
   Candidate* _candidates;
   int _candidates_number;
};
```

main.cpp

```
#include <iostream>
#include "contest.h"

using namespace std;

int main(){
   string spain_candidates[]{"Sweden", "France", "Cyprus"};
   string france_candidates[]{"Sweden", "Spain", "Cyprus"};
   string sweden_candidates[]{"France", "Spain", "Cyprus"};
   string cyprus_candidates[]{"Sweden", "France", "Spain"};
   string eurovision_candidates[]{"Cyprus", "France", "Spain", "Sweden"};

   cout << "Welcome to Eurovision 2019!" << endl;

   // Getting votes from Spain
   Contest spain_voting("Spain", spain_candidates, 3);
   cout << "It is time for Spain to vote!" << endl;
   spain_voting.voteFor("France", 9);
   spain_voting.voteFor("Sweden", 10);
   spain_voting.voteFor("Cyprus", 12);

   // Getting votes from France
   Contest france_voting("France", france_candidates, 3);
   cout << endl << "Muchas gracias! Let's go to Paris now!" << endl;
   france_voting.voteFor("Sweden", 9);
   france_voting.voteFor("Spain", 10);
   france_voting.voteFor("Cyprus", 12);

   // Getting votes from Sweden
   Contest sweden_voting("Sweden", sweden_candidates, 3);
   cout << endl << "Merci beaucoup! We are going to listen now to our swedish friends..." << endl;
   sweden_voting.voteFor("Spain", 9);
   sweden_voting.voteFor("France", 10);
   sweden_voting.voteFor("Cyprus", 12);

   // Getting votes from Cyprus
   Contest cyprus_voting("Cyprus", cyprus_candidates, 3);
   cout << endl << "Tack! Finally, let's travel to Cyprus to get the last votes of the night" << endl;
   cyprus_voting.voteFor("France", 9);
   cyprus_voting.voteFor("Spain", 10);
   cyprus_voting.voteFor("Sweden", 12);

   cout << endl << "Thank you all for your participation. Let's look at the final ranking" << endl;

   // Add all country votes to get final results
   Contest final_results("Eurovision 2019", eurovision_candidates, 4);
   final_results+=spain_voting;
   final_results+=france_voting;
   final_results+=sweden_voting;
   final_results+=cyprus_voting;
   cout << final_results << endl;
   final_results.showWinner();

   return 0;
}
```

The result of the execution of the code produces the following output on the screen:

```
Welcome to Eurovision 2019!
It is time for Spain to vote!
        - 9 points go to... France!
        - 10 points go to... Sweden!
        - 12 points go to... Cyprus!

Muchas gracias! Let's go to Paris now!
        - 9 points go to... Sweden!
        - 10 points go to... Spain!
        - 12 points go to... Cyprus!

Merci beaucoup! We are going to listen now to our swedish friends...
        - 9 points go to... Spain!
        - 10 points go to... France!
        - 12 points go to... Cyprus!

Tack! Finally, let's travel to Cyprus to get the last votes of the
night
        - 9 points go to... France!
        - 10 points go to... Spain!
        - 12 points go to... Sweden!

Thank you all for your participation. Let's look at the final ranking
Eurovision 2019 Results:
Name    Votes:
Cyprus  36
France  28
Spain   29
Sweden  31

The winner, with a total of 36 points is... Cyprus! Congratulations!!
```

Solution

```
#include <iostream>

using namespace std;

struct Candidate { // [1] Candidate structure
   string name;
   int points;
};

class Contest {
public:
   Contest(string name, string candidate_names[], int candidates_number){ // [2] Parametrized
constructor
// [3] Initialize class attributes
      _name = name;
      _candidates_number = candidates_number;
      _candidates = new Candidate[_candidates_number]; // [4] Allocate required space for
candidates

      // [5] Initialized candidates list with received names
      for(int i=0; i<_candidates_number; i++){
         _candidates[i].name = candidate_names[i];
         _candidates[i].points = 0;
      }
   }

   void showWinner(){ // [6] Method to compute and announce the winner
      string winner = "";
      int points = -1;
      for(int i=0; i<_candidates_number; i++){
         if(_candidates[i].points>points){
            points = _candidates[i].points;
            winner = _candidates[i].name;
         }
      }
      cout << "The winner, with a total of "<< points <<  " points is... " << winner << "!
Congratulations!!"<< endl;
   }

   bool voteFor(string country, int points){ // [7] Method to add votes to candidates
      for(int i=0; i<_candidates_number; i++){
         if(_candidates[i].name.compare(country)==0){
            cout << "\t- " << points << " points go to... " << country << "!" << endl;
            _candidates[i].points=_candidates[i].points+points;
            return true;
         }
      }
      return false;
   }

   Contest& operator +=(const Contest& p){ // [8] Operator +=
      for(int i=0; i<_candidates_number; i++){
         string current_name = _candidates[i].name;

         for(int j=0; j<p._candidates_number; j++){
            if(p._candidates[j].name==current_name){
               _candidates[i].points = _candidates[i].points+p._candidates[j].points;
```

```
                break;
            }
          }
        }
        return *this;
    }

    friend ostream& operator<<(ostream& os, const Contest& p){ // [9] Operator <<
        os << p._name << " Results: " <<endl;
        os << "Name\tVotes: " <<endl;
        for(int i=0; i<p._candidates_number; i++){
            os<< p._candidates[i].name<<"\t"<<p._candidates[i].points << endl;
        }
        return os;
    }

    ~Contest(){ // [10]  Destructor
        if(_candidates_number>0){
            delete [] _candidates;
        }
    }

private:
    string _name;
    Candidate* _candidates;
    int _candidates_number;
};
```