Project Based Lab on OOPs

Voting System

A report submitted in partial fulfilment of requirements of the Lab Project of

Bachelor of Technology

In

Computer Science and Engineering (CSE)

By

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Name of the Student | Sec | R.No | Id.No |
| 1 | P.Tejaswi | B | 17 | 12003072 |
| 2 | G.Balaram | B | 48 | 12003381 |
| 3 | Ch.Durga Manohar Reddy | B | 15 | 12003056 |
| 4 | V.Sanjay | B | 46 | 12003360 |

Under the Guidance of

**Dr. D. Rajeswara Rao (Professor)**

****

**Department of Computer Science and Engineering**

**KLUniversity, Vaddeswaram, Guntur Dt. (2013-14)**



**K.L. UNIVERSITY**

**Green fields, Vaddeswaram, Guntur Dist**.

CERTIFICATE

This is certify that this project work entitled “Voting System” being submitted to Department of Computer Science and Engineering (CSE), KlUniversity by P.Tejaswi (12003072), G.Balaram (12003381), Ch.Durga Manohar Reddy (12003056), V.Sanjay (12003360) is a bonified work carried out by them in Department of Computer Science Engineering

Project supervisor Head of the Department

Dr.D.Rajeswara Rao (Proff)

**ACKNOWLEDGMENT**

We express my sincere gratitude to “Dr. D. Rajeswara Rao” for encouraging and guiding us to undertake this project work.

We express my deep sense of gratitude to Dr.V.Srikanth (HOD,CSE) and Dr.D.Rajeswara Rao (Proff) and our beloved course lecturers of department for their encouragement.

We would like to express our special gratitude and thanks to all the faculty and staff for giving me such attention and time. We would like to express our gratitude towards our parents for their kind co-operation and encouragement which helped me in completion of this project.

Our thanks and appreciations also go to our friends in developing the project and people who have willingly helped me out with their abilities.

P.Tejaswi (12003072)

G.Balaram (12003381)

Ch.Durga Manohar Reddy (12003056)

V.Sanjay (12003360)

**INDEX**

1. **Abstract** …………………………………………………………….…….5
2. **SRS-Software and Hardware Requirements**...........................................6
3. **Class Hierarchy**…………………………………………………………..7
4. **Description of each class with data members & member functions**….8-9
5. **Modules**…………………………………………………………………..10-15
6. **Source Code**……………………………………………………………...16-21
7. **Screen shots**………………………………………………………………22-23
8. **Conclusion**………………………………………………………………..24
9. **References**……………………………………………………………..….24

**Abstract**

Our project Voting system is developed using c++ programming language. The output of this project is to display winner of the contest by displaying no. of votes that company has got and the majority the company has got when compared to other companies. In this program the details of the voter are maintained in the array and each voter was given a voter id that is unique. When the voter enters the voter id, the program will check that an individual is not allowed to vote more than once i.e. once a voter has finished voting, his/her voter id is disabled. Data is much secured in this program as we are using the c++ programming which will not allow its data to flow around. The total number of votes for each of the company is calculated and then we will count which company has got the highest votes. Company which has got the highest votes are displayed as the winner and we will display how many votes that company has got.

**SRS-Software and Hardware Requirement Specification**

**Hardware Requirements**:

RAM: 4GB minimum requirement: 1 GB

Processor: AMD A8-4500APU with HD graphics

OS: windows 7 professional

HARD DISK: 40 GB or more

**Software Requirements**:

User Interface: Turboc30 version of c++ programming

**Class Hierarchy**

In this project we get multiple inheritance as a class inherit the attributes of two classes.

class voter class cand

protected:

cand

public:

cand()

protected:

id,wardno,gender

public:

voter()

showdata()

private:

big,big2,voterid,majority

public:

cast()

count(),declare()

class voting:public voter,public cand

**Class Hierarchy showing multiple Inheritance**

**Description of each class with data members & member functions**

**class voter**

Data members:

id[1000]:contains the voter id of the voters

wardno[1000]:contains the wardno of the voter with reference to voter id

voted[1000]:used to check if that voter has completed his voting

gender[1000]:contains the gender of voter with reference to voterid

Function members:

voter():

It is constructor which is used to register the voter details. The main thing to check in this registration is whether that voter id is already registered or not.If it is already registered we will not register it again. If that voter id is not registered we will register that voter details

void showdata(int)

It is used to display the voter details of a particular voter with reference to voterid

**class cand**

Data members:

can[5]-used to count the no.of voted each company has got

Function members:

cand():

It is construct which initially assigns the no.of votes of each company as 0

**class voting:public voter,public cand**

Data members:

voterid: reads the voterid of user during casting the vote which is used to pass as argument to getdata() function of voter class which display the voter details of that particular voter id and also used to check if that voter has already casted his vote

big: used to find the highest votes

big2: used to find second highest votes

majority: used to calculate the difference of first and second highest votes

Function members:

void cast():

This function mainly deals about casting vote from the user. First we will read voter id from the user and then we will check if his voter id is matching with any in data storage class and if he is using his voter for the first time. If it is so then we will display his details and display the choice of companies for voting and read the vote from the user and add it to the company he voted for. If it is not so (i.e., if he is using his voter more than once or his voter id is not in the registered id numbers) then we will display error message

void count():

The function mainly deals about finding the highest votes. After the voting process is totally completed we will check the company which has got highest votes .And we also find the company which has got the second highest votes. Then we will calculate the difference between first highest and the second highest to find the majority of votes the first highest company has got

void declare():

This function mainly deals about the declaring the results. After finding out the first highest votes we will check which company has got these highest votes and then we will declare the name of the company as the winner of the voting. We will display the name of the company and the total number of votes the company has got and the majority (i.e., difference of first highest and the second highest) votes the winning company has got

**Modules**

We divided the entire program into 4 modules and each module is completed by each of the members of our team.

Module 1 G.Balaram (12003381)

Module 2 P.Tejaswi (12003072)

Module 3 Ch.Durga Manohar Reddy (12003056)

Module 4 V.Sanjay (12003360)

Where the modules are

Module 1 registration of voter details

Module 2 casting the vote

Module 3 Counting the votes

Module 4 Declaration of results

**Module-1**

**Registration of voter details**

In order to start a voting process first thing to be done is getting the details of the voters like voterid, ward number and gender of the voter. We will use data storage class (array) to read all the details from the user. The main thing we have to check in this registration is than no voter can register his voter id more than once. So we will develop a project in such a way that once he has registered his voter id he cannot register it once again

Pseudo code:

void voter::voter(){

Enter no.of voters; cin>>n;

for(i=0;i<n;i++)

{

Enter voterid id[i];

for(j=0;j<i;j++)

if(id[j]==id[i])

check=1;

if(check==1)

{

This voter id already registered

i--;

check=0;

}

else

enter voter details

}}}

void cand::cand(){

for(i=0;i<5;i++)

can[i]=0;}

**Module-2**

**Casting the vote**

This module mainly deals about casting vote from the user. First we will read voter id from the user and then we will check if his voter id is matching with any in data storage class and if he is using his voter for the first time. If it is so then we will display his details and display the choice of companies for voting and read the vote from the user and add it to the company he voted for. If it is not so (i.e., if he is using his voter more than once or his voter id is not in the registered id numbers) then we will display error message

Pseudo code:

void voter::showdata(int id1)

{

int i;

for(i=0;i<n;i++)

{

if(id[i]==id1)

{

cout<<"\nwardno is:"<<wardno[i]<<"\ngender is:"<<gender[i];

}

}

void voting::cast()

{

int i,ch;

cout<<"\nenter voter id:";

cin>>voterid;

for(i=0;i<n;i++)

{

if(voted[i]!=0 && id[i]==voterid)

{

cout<<"\nyour voting is over ";

}

else if(id[i]==voterid && voted[i]==0)

{

voted[i]=1;

showdata(voterid);

cout<<"\nCOMPANIES ARE";

cout<<"\n1\tACER";

cout<<"\n2\tHP";

……………………….

cout<<"\nEnter your choice";

cin>>ch;

switch(ch)

{

case 1:

{

can[0]++;

break;

}

case 2:

{

can[1]++;

break;

}

………………

}

}

}

**Module-3**

**Counting the votes**

The module mainly deals about finding the company which has got the highest votes. After the voting process is totally completed we will check the company which has got highest votes .And we also find the company which has got the second highest votes. Then we will calculate the difference between first highest and the second highest to find the majority of votes the first highest company has got

Pseudo code:

void voting::count()

{

big=0,big2=0,majority=0;

int i;

big=can[0];

for(i=1;i<5;i++)

{

if(can[i]>big)

{

big=can[i];

}

}

for(i=0;i<5 ;i++)

{

if(can[i]>big2 && can[i]!=big)

{

big2=can[i];

}

majority=big-big2;

}

**Module-4**

**Declaration of results**

After finding out the first highest votes we will check which company has got these highest votes and then we will declare the name of the company as the winner of the voting. We will display the name of the company and the total number of votes the company has got and the majority (i.e., difference of first highest and the second highest) votes the winning company has got

Pseudo code:

void voting::declare()

{

if(big==can[0])

cout<<"\nWINNER IS ACER";

if(big==can[1])

cout<<"\nWINNER IS HP";

……………………………………………

cout<<"\nvotes are"<<big<<"\nmajority is "<<majority;

}

**Source Code**

#include<iostream.h>

#include<conio.h>

class voter

{

protected:

int id[1000],wardno[1000],voted[1000];

char gender[1000];

public:

int n;

voter();

void showdata(int id1);

};

class cand

{

protected:

int can[5];

public:

cand();

};

class voting:public voter,public cand

{

private:

int voterid,big,big2,majority;

public:

int cast();

void count();

void declare();

};

void voter::voter()

{

int i,check=0,j;

cout<<"\nenter no.of voters";

cin>>n;

for(i=0;i<n;i++)

{

cout<<"\nEnter"<<i+1<<"voter details:";

cout<<"\nEnter voterid:";

cin>>id[i];

for(j=0;j<i;j++)

{

if(id[j]==id[i])

check=1;

}

if(check==1)

{

cout<<"\nThis voter id already registered";

i--;

check=0;

}

else

{

cout<<"\nEnter wardno:";

cin>>wardno[i];

cout<<"\nEnter gender:";

cin>>gender[i];

voted[i]=0;

}

}

}

void cand::cand()

{

int m;

for(m=0;m<5;m++)

can[m]=0;

}

void voter::showdata(int id1)

{

int i;

for(i=0;i<n;i++)

{

if(id[i]==id1)

{

cout<<"\nwardno is:"<<wardno[i]<<"\ngender is:"<<gender[i];

}

}

}

int voting::cast()

{

int k,ch,voterid=0;

int j=6;

cout<<"\nenter voter id:";

cin>>voterid;

for(k=0;k<n;k++)

{

if(voted[k]!=0 && id[k]==voterid)

{

cout<<"\nyour voting is over ";

}

else if(id[k]==voterid && voted[k]==0)

{

voted[k]=1;

voter::showdata(voterid);

cout<<"\nCOMPANIES ARE";

cout<<"\n1\tACER";

cout<<"\n2\tHP";

cout<<"\n3\tDELL";

cout<<"\n4\tSONY";

cout<<"\n5\tAPPLE";

cout<<"\nEnter your choice";

cin>>ch;

switch(ch)

{

case 1:

can[0]++;

break;

case 2:

can[1]++;

break;

case 3:

can[2]++;

break;

case 4:

can[3]++;

break;

case 5:

can[4]++;

break;

default:

cout<<"\nEnter valid choice";

}

}

else

cout<<" ";

}

return j;

}

void voting::count()

{

big=0,big2=0,majority=0;

int i;

big=can[0];

for(i=1;i<5;i++)

{

if(can[i]>big)

big=can[i];

}

for(i=0;i<5 ;i++)

{

if(can[i]>big2 && can[i]!=big)

big2=can[i];

}

majority=big-big2;

}

void voting::declare()

{

if(big==can[0] && big>=1)

cout<<"\nWINNER IS ACER";

if(big==can[1] && big>=1)

cout<<"\nWINNER IS HP";

if(big==can[2] && big>=1)

cout<<"\nWINNER IS DELL";

if(big==can[3] && big>=1)

cout<<"\nWINNER IS SONY";

if(big==can[4] && big>=1)

cout<<"\nWINNER IS APPLE";

cout<<"\nvotes are"<<big<<"\nmajority is "<<majority;

}

int main()

{

clrscr();

cout<<"\nEnter voter details:";

voting v1;

int j=0,close;

clrscr();

while(j!=1)

{

v1.cast();

cout<<"\nenter 1 if voting is over else enter 0";

cin>>close;

if(close==1)

j=1;

}

v1.count();

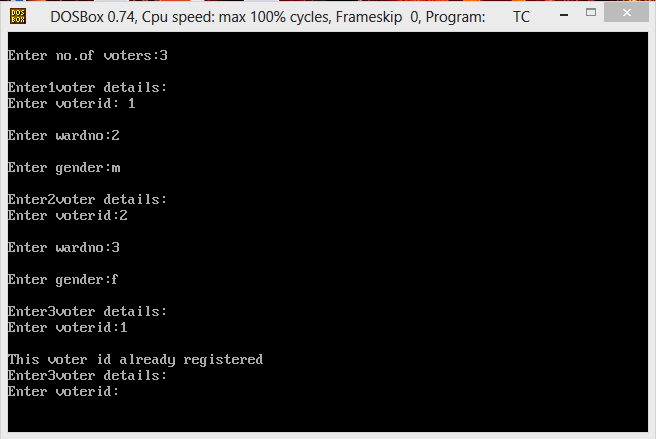
v1.declare();

getch();

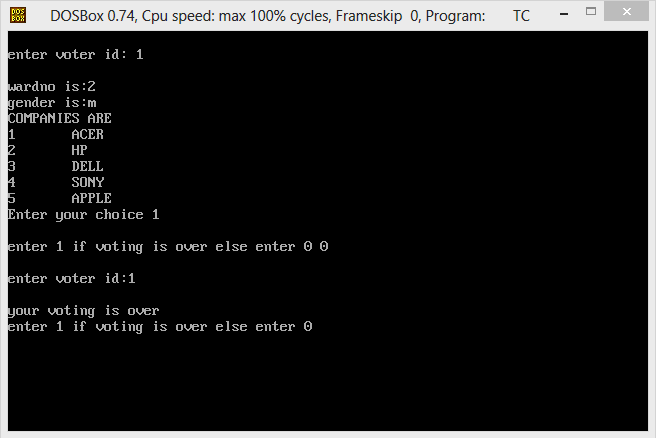
return 0;

}

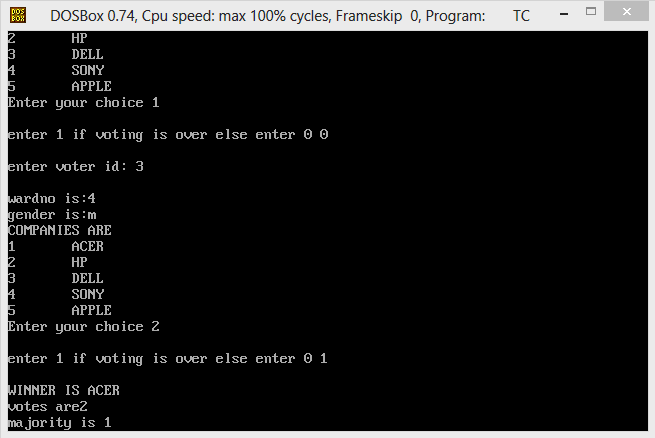
**Ouput**

****

**Output showing the registration of voter details**

****

**Output showing the choice of companies to user after reading his voter id**

****

**Output showing the result to the user**

**Conclusion**

This ends out voting system project in ‘c++’ which is a small attempt in big ocean of c++ programming. Here using the concept of inheritance and constructors we had been completed and come up with the desired output of what we want. With this project using c++ concepts we had converted the general process of voting system in to program based voting system.

**References**

<http://www.technoexam.com/>

http://electrofriends.com/source-codes/software-programs