

OBJECT ORIENTED PROGRAMMING PROJECT

SEMESTER-III

**PRESENTED BY:**

**ABHISHEK SHUKLA (14103226)**

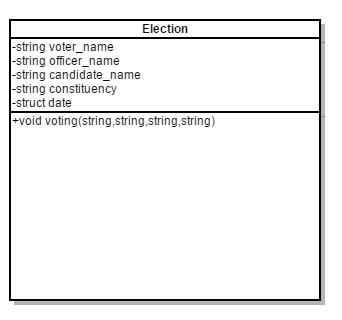
**AAKASH BANSAL (14103225)**

**SHAURYA RASTOGI (14103241)**

**GAURAV AGGARWAL (14103215)**

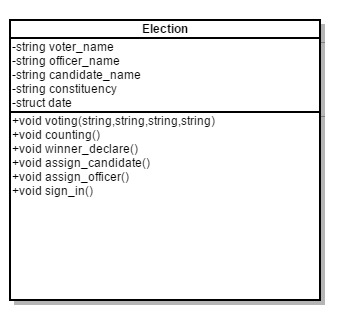
## INCREMENT 1

## SINGLE APPLICATION, SINGLE CLASS, SINGLE METHOD



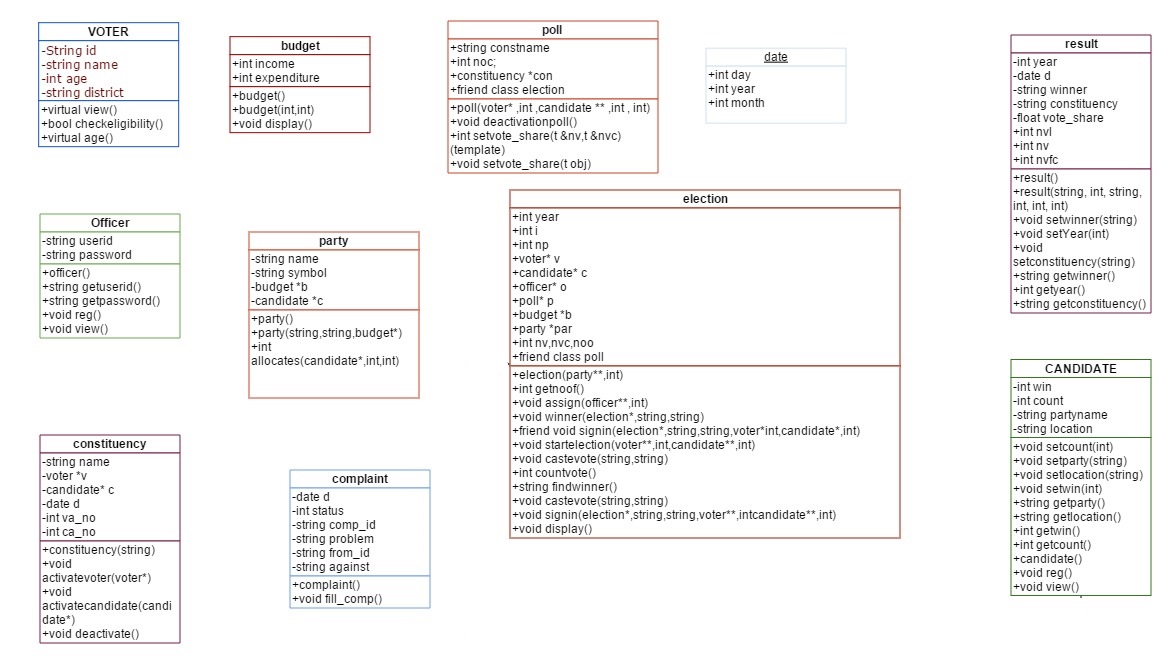
## INCREMENT 2

## SINGLE APPLICATION, SINGLE CLASS, MULTIPLE METHOD



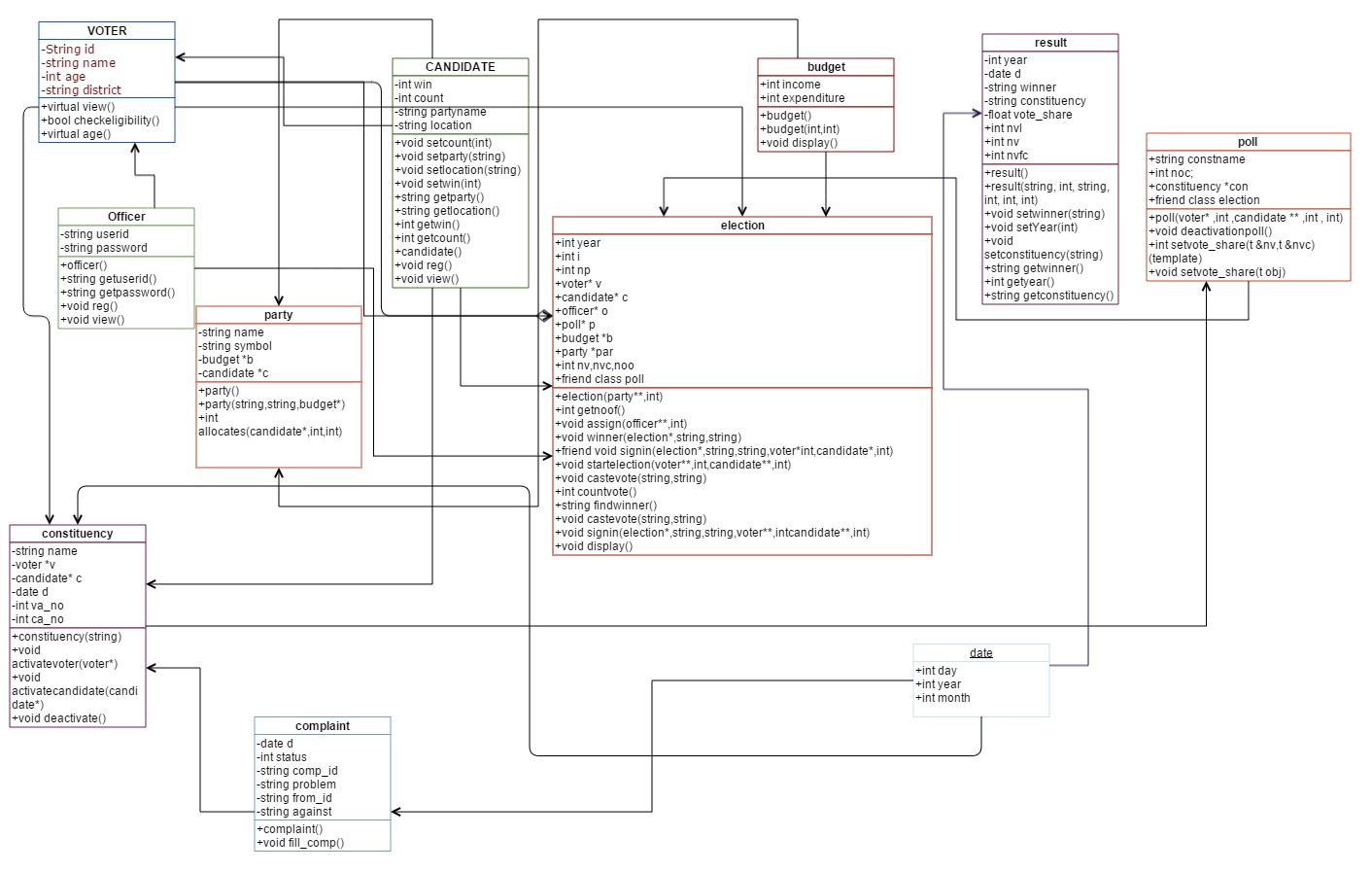
# INCREMENT 3

# SINGLE APPLICATION, MULTIPLE CLASS, MULTIPLE METHOD, SIMPLE RELATIONSHIP



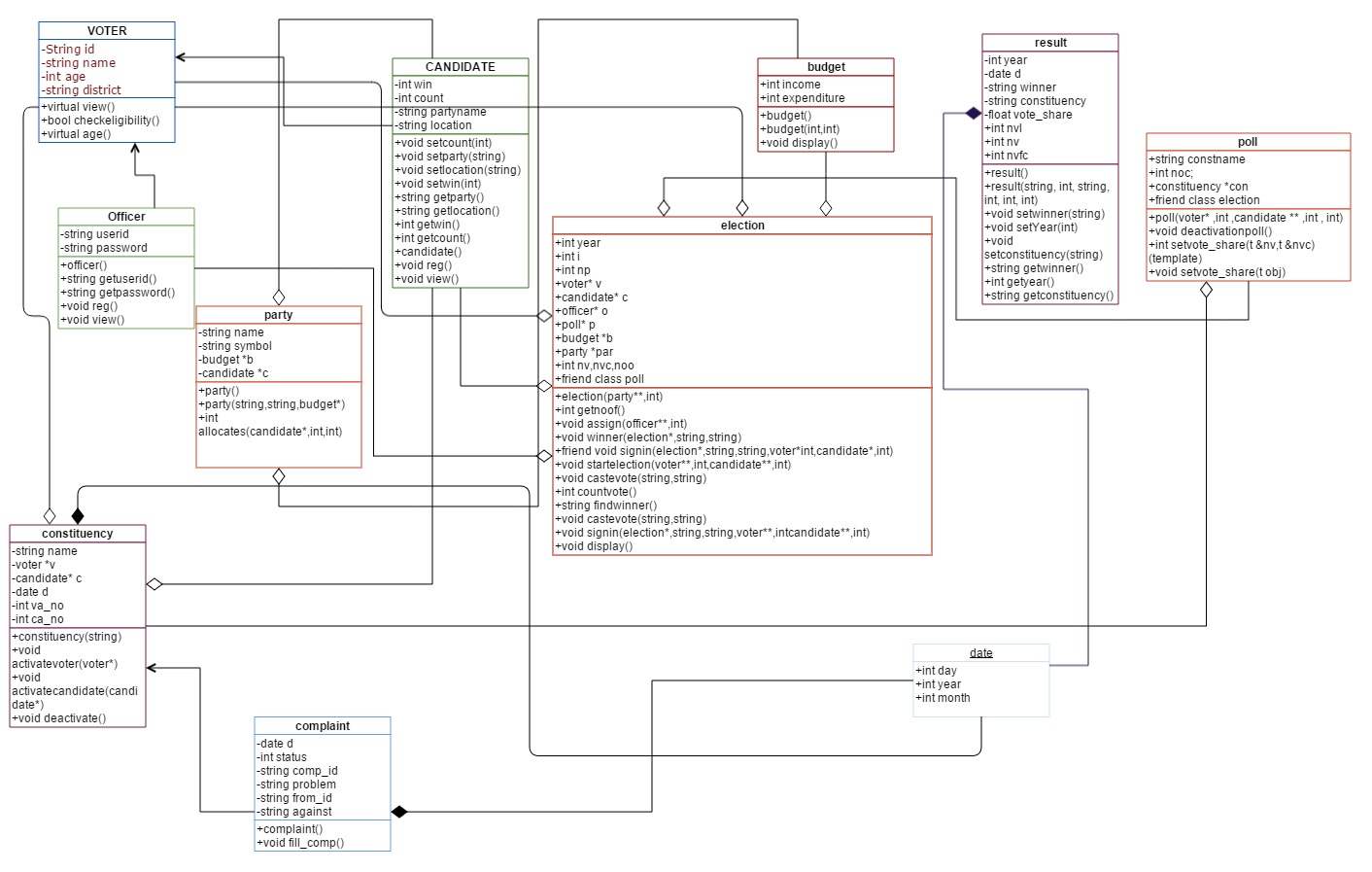
## INCREMENT 4

## SINGLE APPLICATION, MULTIPLE CLASS, MULTIPLE METHOD, COMPLEX RELATIONSHIP



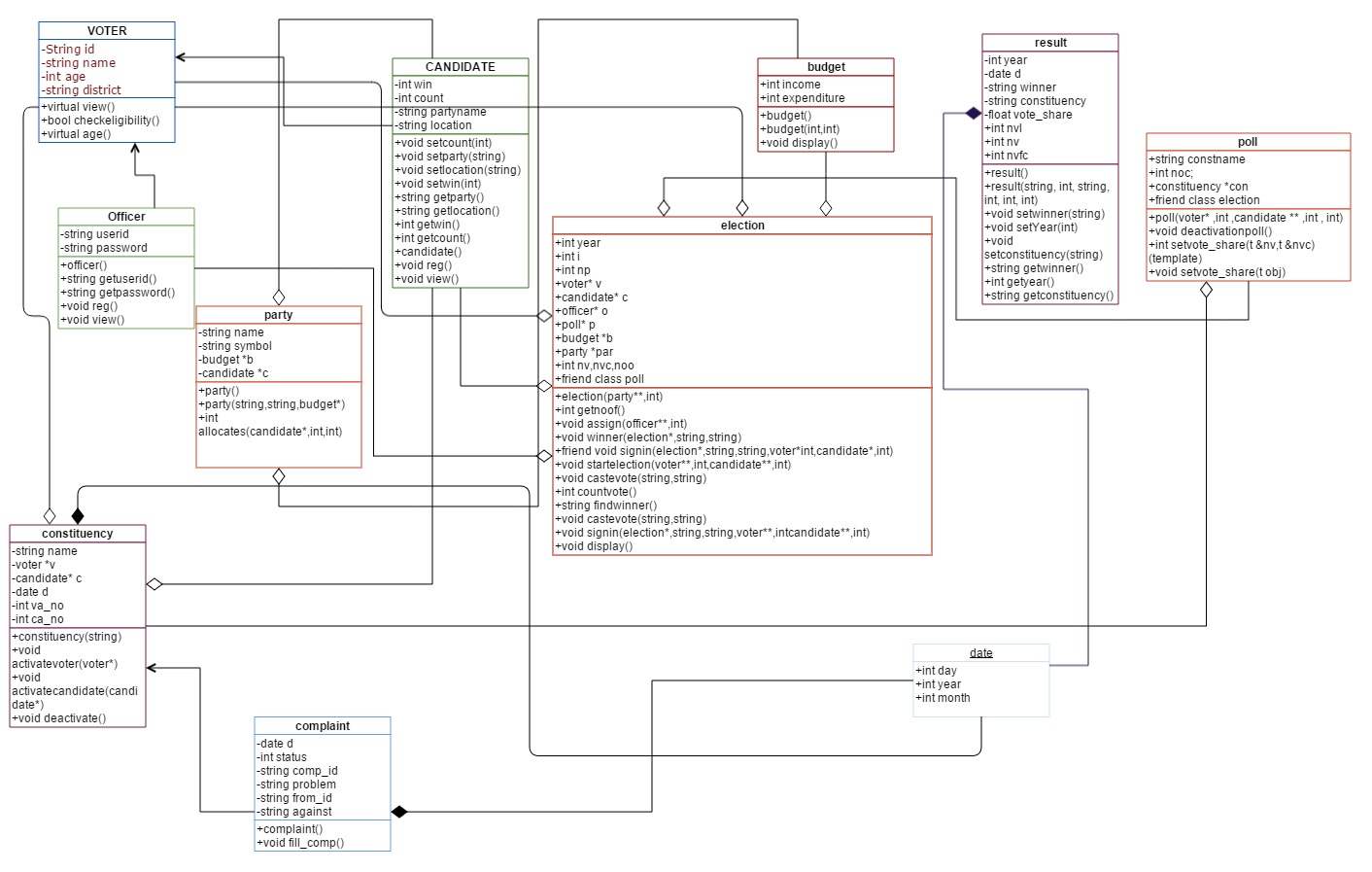
## INCREMENT 5

## SINGLE APPLICATION, MULTIPLE CLASS, MULTIPLE METHOD, AGGREGIATION/COMPOSITE ASSOCIATION



## INCREMENT 6

## Single Application, Multiple Class, Multiple Method, Inheritance (including Friend Functions)



## INCREMENT 7

## Single Application, Multiple Class, Multiple Method, Polymorphism (including Overloading and Overriding)

 virtual void view()

    {

        cout<<"\nID\t:"<<id;

        cout<<"\nName\t:"<<name;

        cout<<"\nAge\t"<<age;

        cout<<"\nDistrict\t"<<district;

    }

`

   virtual void reg()

    {

    cout<<"\n Enter the Id of the voter\t";

    cin>>id;

    try

    {

    int l,i;

    l=id.size();

    for(i=0;i<l;i++)

        if(id[i]>'a'&&id[i]<'z')

        continue;

        else if(id[i]>'A'&&id[i]<'Z')

            continue;

        else if(id[i]>42&&id[i]<=50)

            continue;

        else

            throw l;

}

catch(int l)

{

    cout<<"special Characters are not allowed in userid\n";

}

    cout<<"\n Enter the Name of the Voter\t";

    cin>>name;

    cout<<"\n Enter the age of the voter\t";

    cin>>age;

    if(checkeligibility(age))

        {

             cout<<"\n\tYou are eligible\n";

             cout<<"\n\n Enter the district of the voter\t";

             cin>>district;

        }

    else

        cout<<"\n\t you are not eligible\n";

    }

## INCREMENT 8

Templates

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Templates\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//

template<typename t>

int setvote\_share(t &nv,t &nvc)

{

int vote\_share;

vote\_share=nvc\*100/nv;

return vote\_share;

}

template<typename t>

void setvote\_share(t obj)

{

obj.vote\_share=obj.nvc\*100/obj.nv;

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Template -2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_//

template<typename t>

void sortedlist(t \*a,int n)

{

    cout<<"\n Sorted voter List\t";

    for(int i=0;i<n;i++)

    {

        for(int j=i+1;j<n;j++)

        {

            voter \*s;

            if(a->v[i]->getname()>a->v[j]->getname())

                {

                   s=a->v[i];

                   a->v[i]=a->v[j];

                   a->v[j]=s;

                }

        }

        for(i=0;i<n;i++)

        {

        a->v[i]->view();

        cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

        }

    }

}

## INCREMENT 9

Exception Handling to Check Username is not Invalid ( Username containing Special Characters)

try

{

int l,i;

l=userid.size();

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

if(userid[i]>'a'&&userid[i]<'z')

continue;

else if(userid[i]>'A'&&userid[i]<'Z')

continue;

else if(userid[i]>42&&userid[i]<=50)

continue;

else

throw l;

}

catch(int l)

{

cout<<"special Characters are not allowed in userid\n";

}

Exception Handler to check System Error due to improper Allocation of Memory

try

{

cout<<"\n enter constituency where elections are to be held\t";

cin>>constname;

con[i]=new constituency(constname);

cout<<"\n\_\_\_\_\_\_Activated Constituency\_\_\_\n";

//to associate voters of a constituency

for(int j=0;j<nv;j++)

if(con[i]->name==v[j]->getdistrict())

con[i]->activatevoter(v[j]);

//to associate candidates of a constituency

cout<<"\n\_\_\_\_\_\_\_\_\_\_Activated Voter List\_\_\_\_\_\_\_\_\_\_\n";

for(int j=0;j<nc;j++)

if(con[i]->name==c[j]->getlocation())

con[i]->activatecandidate(c[j]);

cout<<"\n\_\_\_\_\_\_\_\_\_\_\_Activated Candidature\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Poll Activated \_\_\_\_\_\_\_\_\_\_\_\_\_\n";

throw e;

}

catch(exception &e)

{

cout<<"\n Exception is \t"<<e.what();

}

Exception Handler to scrutinize the validity of the officer to set winner for a particular election

try

{ for(j=0;j<e->getnoof();j++)

{ if(o[j]->getuserid()==userid && o[j]->getpassword()==password)

for(i=0;i<10;i++)

{ if(ca->getname()==c[i]->getname())

{ ca->setwin(1);

n=1;

}

else

{

ca->setwin(0);

n=1;s

}

}

}

if(n==1)

cout<<"\nstatus is set\t";

else

throw n;

}

catch(int n)

{

cout<<"\nYou cannot set status\t";

}

Exception Handler to handle the legitimacy of the voter casting his vote (To vote in the district to

which he belongs and nowhere else.)

try

{

if(v[i]->district!=constituency)

throw 0;

}

catch(int x)

{

cout<<"\nYou are not the legitimate voter Sir!\n You Belong to a different constituency...\n";

}