Algorithm: Roots of quadratic equation

Include predefined class libraries / header files in the program using preprocessor directive #include <iostream> , <cmath>

using namespace std;

Class declaration: declare a class “quadratic”

Private member declaration:

Data variables:
declare float type data variables “x,y,rpx,ipx,D,a,b,c”

Public member declaration:
Declare member function (1) getdata (2) calculate of type void

Member function definition:
Define member functions getdata using scope resolution operator

“void quadratic::getdata()”

Get input from user:

cout<<"enter the cofficients a,b,c"<<endl;

The input is written in allocated memory space using
“cin>>a>>b>>c; ”

Member function definition:

Define member functions calculate using scope resolution operator
“void quadratic::calculate()”

Perform the calculations and output the results using cout

```
D=b*b-4*a*c;                                          //disciminant calculation

if(D>0)                                               //if loop : condition 1
{
   cout<<"roots are real and different"<<endl;
   x=((-b)+sqrt(D))/(2*a);                            //calculate roots of equation
   y=((-b)-sqrt(D))/(2*a);
   cout<<"roots of given eqn are = "<<x<<","<<y<<endl;  //display results on screen
 }


else if(D==0)                                         //if loop : condition 2
{
   cout<<"roots are real and equal"<<endl;
   x=((-b)/(2*a));                                    //calculate roots of equation
   y=x;
```

```
    cout<<"roots of given eqn are="<<x<<","<<y<<endl;                    //display output on screen
  }

  else if(D<0)                                                           //if loop : condition 3
  {
    cout<<"roots are complex"<<endl;
    rpx=(-b)/(2*a);                                                      //calculate roots of equation
    ipx=(sqrt(-D))/(2*a);
    cout<<"first root of given eqn is ="<<rpx<<"+i"<<ipx<<endl;          //display result on screen
    cout<<"second root of given eqn is ="<<rpx<<"-i"<<ipx<<endl;
  }

}
```

Inside the main function of type integer, create an object of class quadratic and named q. Use this object to call public member functions and terminate the program using return 0 statement

```
int main()
  {
    quadratic q;                                                         //create object
    q.getdata();                                                         //member function calling
    q.calculate();
    return 0;                                                            //program termination
   }
```