Algorithm: Mean, Variance and Standard deviation calculation

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

using namespace std;

Class declaration: declare a class “MVS”

Private member declaration:

Data variables:
declare float type data variables “i,n”
declare float type data variables “x[200],sum,var,mean,variance,SD”
where x[200] is declared as array and can contain upto 201 values.

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

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

“void MVS::getdata()”

Get input from user:

"enter the value of n="
The input is written in allocated memory space using
“cin>>n;”

Get input from user for parameter x[]
The input is written in allocated memory space using “for loop”

```
for (i=0;i<n;i++)
{
cin>>x[i];
}
```

Member function definition:
Define member functions calculate using scope resolution operator
“void MVS::caldata()”

Perform the calculations using for loop and display/output the results using **cout**

```
for(i=0;i<n;i++)
{
sum=sum+x[i]                                    //calculate sum
 }
mean=sum/n;                                     //calculate mean
cout<<"mean="<<mean<<endl;                      //display mean on screen

 for(i=0;i<n;i++)
{
 var=var+pow((x[i]-mean),2);                    //calculate variance
}
variance=var/n;
cout<<"variance="<<variance<<endl;              //display variance on screen
SD=sqrt(variance);                              //calculate standard deviation
cout<<"standard deviation="<<SD<<endl;          //display standard deviation
}
```

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

```
int main()
{
MVS a;                                          //create object
a.getdata();                                    //member function calling
a.caldata();
return 0;                                       //program termination
}
```