More on classes

Syntex

```
class  ClassName
{
        members ;

};
```

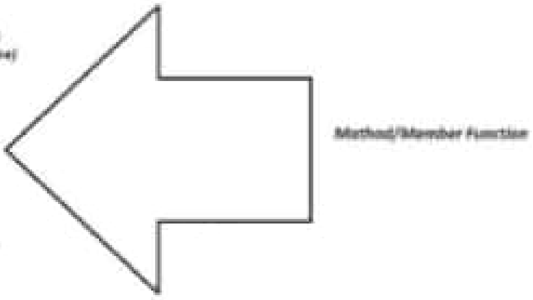
- C++ ***class always terminated with semicolon (;)***
- We can **declare *members inside the class i.e. properties, methods etc***
- ***Properties represent through variables / data structure*** and in C++ it is often known as data members.

```
class Book
{
        private:
                char name[20];
                int page;
                float price;
        public:
                .....
};
```

Properties /Attributes /
Data members

- ***Methods represent through function and in C++*** it is often known as behaviour/member function.

```
class Book
{
                char name[20];
                int page;
                float price;
        public:
                void getdata()
                {
                        cout << endl << "Enter book name";
                        //cin >> name ; // cin.operator(name)
                        cin.getline(name,20);
                        cout << endl << "Enter pages ";
                        cin >> page;
                        cout << endl << "Enter price ";
                        cin >> price;
                }
                void display()
                {
                        cout << endl << "Book #";
                        cout << endl << "Name : " << name;
                        cout << endl << "page : " << page;
                        cout << endl << "price: " << price;
                }
};
```



Encapsulation

- Encapsulation is the mechanism that binds together **code** (function) and the data it manipulates. Other way to think about encapsulation is, ***it is a protective shield that prevents the data from being accessed by the code outside this shield.***
- **class** ***is the keyword in C++ that allow us to implement the Encapsulation.***
- ***The size of the class is total depend*** **non-static members** ***of class.***

```
#include<iostream>
using namespace std;
class Book
{
```

```
            private:
                        char name[20];
                        int page;
                        float price;
            public:
                        void getdata()
                        {
                                    cout << endl << "Enter book name";
                                    //cin >> name ; // cin.operator(name)
                                    cin.getline(name,20);
                                    cout << endl << "Enter pages ";
                                    cin >> page;
                                    cout << endl << "Enter price ";
                                    cin >> price;
                        }
                        void display()
                        {
                                    cout << endl << "Book #";
                                    cout << endl << "Name : " << name;
                                    cout << endl << "page : " << page;
                                    cout << endl << "price: " << price;
                        }
};
main()
{
            Book b ;
            cout << "size of b object is " << sizeof(b);
}
```

- ***A class consists of 0 to N non-static data members. If no members inside the class then the object of its type takes 1 bytes of memory because that is minimum memory for object existence.***

e.g.

```
class Demo
{
};
main ()
{
            Demo a ;
            cout << "sizeof a " << sizeof (a)  ;
}
```

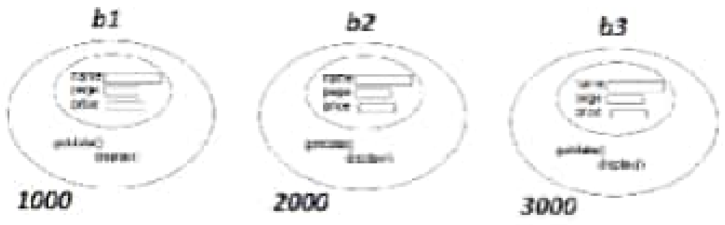
- ***Once a class is created then we can form any number of objects.***

Syntax

```
            className object;
```

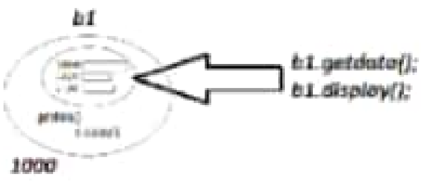
e.g.

```
main()
{
            Book b1, b2 , b3  ;
}
```



- ***Ones a object is created*** then ***object access their public members by . operator.***

```
main()
{
        book b1, b2 , b3;
        b1.getdata();
        b1.display();
        b2.getdata();
        b2.display();
        b3.getdata();
        b3.display();
}
```



- ***Any member function that defines inside the class becomes*** **inline function.**

```
class book
{
        private:
                char name[20];
                int page;
                float price;
        public:
                inline void getdata()
                {
                        cout << endl << "Enter book name";
                        cin >> name;
                        cout << endl << "Enter book page";
                        cin >> page;
                        cout << endl << "Enter book price";
                        cin >> price;
                }
                inline void display()
                {
                        cout << endl << "Book detail #1";
                        cout << endl << "book name : "<< name;
                        cout << endl << "book page : "  << page ;
                        cout << endl << "book price : " << price;

                }
};
```

- ***A member function can be define inside the class and outside the class***

```
class book
{
        private:
                char name[20];
                int page;
                float price;
        public:
                void getdata();
                voiddisplay();
};
```

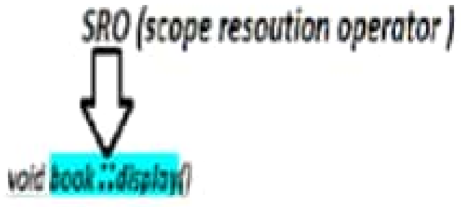
```
void book ::getdata()
{
        cout << endl << "Enter book name";
        cin >> name;
        cout << endl << "Enter book page";
        cin >> page;
        cout << endl << "Enter book price";
        cin >> price;
}
void book ::display()
{
        cout << endl << "Book detail ff1";
        cout << endl << "book name : "<< name;
        cout << endl << "book page : "  << page ;
        cout << endl << "book price : " << price;
}
```

- If we define member function outside the class then
  - :- Prototype of the member function must be declare inside the class.

    void display();

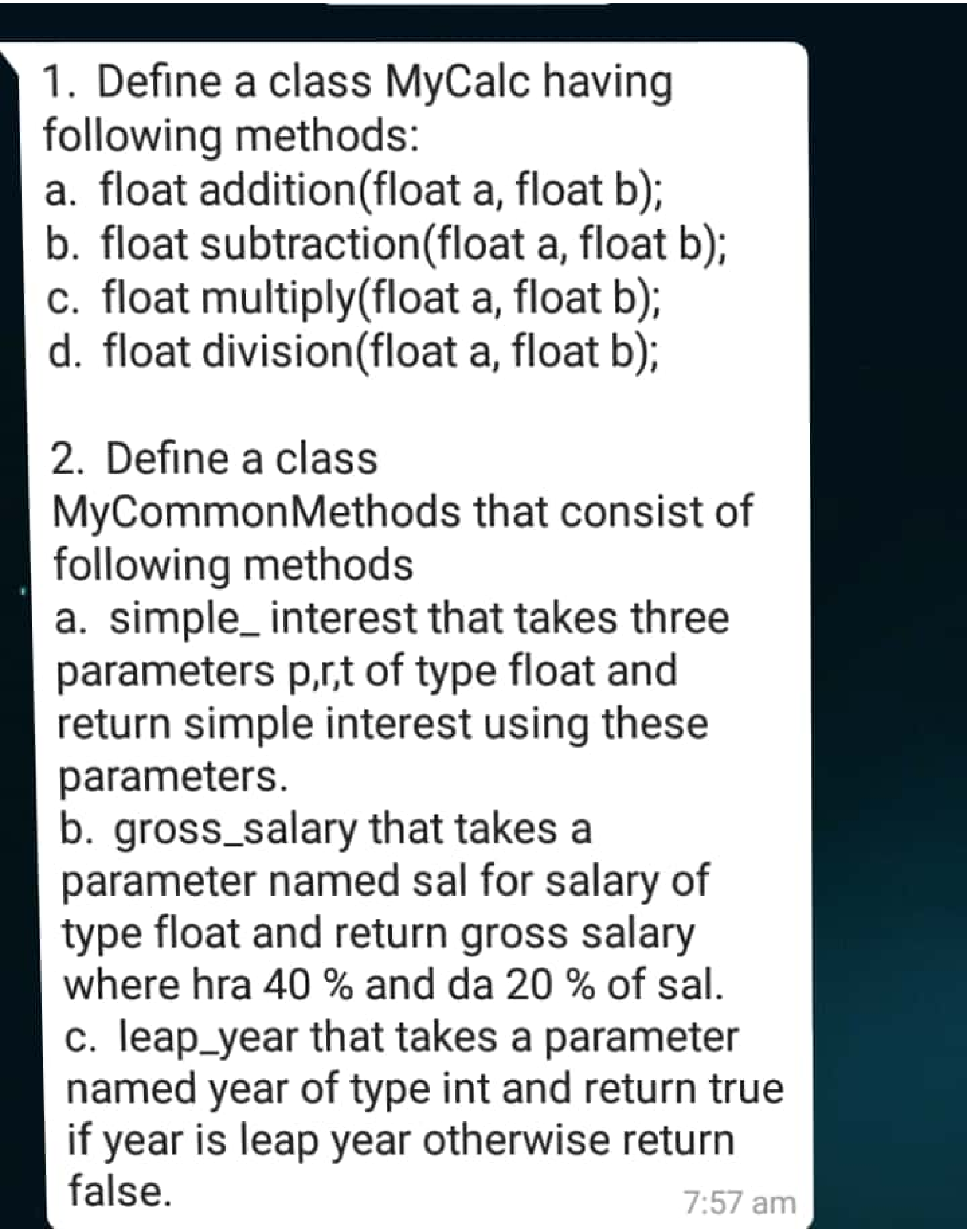
- In function definition which define outside the class must use *full qualifier name*

```
void book ::display()
{
        cout << endl << "Book detail ff1";
        cout << endl << "book name : "<< name;
        cout << endl << "book page : "  << page ;
        cout << endl << "book price : " << price;
}
```



- ***Every C++ class has these members function/methods.***
  - ∴ ***Default constructor***
  - ∴ ***Copy constructor***
  - ∵ ***Address of operator***
  - ∴ ***new operator***
  - ∵ ***assignment operator***
  - ∵ ***delete operator***

```
Class Demo
{
        Default constructor
        Copy constructor
        Address of operator
        new operator
        assignment operator
        delete operator
};
```



1. Define a class MyCalc having following methods:
a. float addition(float a, float b);
b. float subtraction(float a, float b);
c. float multiply(float a, float b);
d. float division(float a, float b);

2. Define a class MyCommonMethods that consist of following methods
a. simple_ interest that takes three parameters p,r,t of type float and return simple interest using these parameters.
b. gross_salary that takes a parameter named sal for salary of type float and return gross salary where hra 40 % and da 20 % of sal.
c. leap_year that takes a parameter named year of type int and return true if year is leap year otherwise return false.

7:57 am