CLASS AND OBJECT

C	C++
1. VARIABLE	1. OBJECT
2. DATA TYPE	2. ABSTRACT DATA TYPE (USER DEFINED DATA TYPE)
3. FUNCTION	3. MEMBER FUNCTION OR OPERATION OR METHOD
4. FUNCTION CALL	4. MESSAGE PASSING

DRAWBACK OF FUNCTIONS

```
int  fact ( int  n )
{
        .
        .
        return(f);
}
int   main()
{
      int   a ;  // OBJECT  OR VARIBLE
      float  b ;
```

```
        char   c ;
        a = fact(3);
        b = fact(3); X
        c = fact(3);  X
}
```

2.

```
        struct    shape
        {
                int a ;
                int b ;  // data
        };
        int     main()
        {
        struct shape ract , square ; // object
              ract . a = 4 ; ract . b = 5;
                square . a = 7, square. b  = 9;
                area ( ract.a , ract.b );
                area ( square.a , square.b);

                area ( ract.a , square.b );  X
                area ( 6 , 7); X
        }
void   area( int  p , int  q )  // function
        {
        cout << p * q << endl;
}
```

C++ :- DATA + FUNCTION

C	C++
structure	structure
OOPS X	OOPS ✓

C++

structure	class	
OOPS ✓	OOPS ✓	(SIMILARITY)
public	private	(DIFFERENCE)

(by default)
public : int a , b ;

---

public	private
main() ✓	main() X
normal fun() ✓	normal fun() X

USE public KEYWORD

```
//    INPUT AND    PRINT NO.
#include<iostream>
using namespace std;
class  test
{
       public : int  a ;
};

int  main()
{
    test  p ;   // VARIABLE OR OBJECT
    p . a = 4;
    cout <<"  "<<  p . a << endl ;  // 4
 }

//  USE  PRIVATE KEYWORD

#include<iostream>
using namespace std;
class  test
{
       private :  int a ;
};
int  main()
{
     test  p ;   // VARIABLE OR OBJECT
```

```
    p . a = 4;
    cout <<"   "<<  p . a << endl ;  // 4
 }
```

ERROR :- a IS NOT ACCESSIBLE
a IS NOT ACCESSIBLE

---

private
---
main() X
normal function X

same class --> public ✓

CLASS :- COLLECTION OF DIFFERENT DATA TYPES.

WITH THE HELP OF CLASS WE CAN CREATE USER - DEFINED-DATA TYPE.

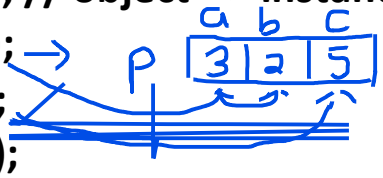
syntax

```
class   tagname
{
        private :       data   member ✓
                        member function

        public :        data   member
                        member function ✓
};
```

```
// INPUT  AND PRINT  NO. USING  CLASS  AND  OBJECT

#include<iostream>
using namespace std;

 class  test       // PROGRAMMER
 {
            private :    int   a ;    // DATA MEMBER

            public  :
                        void   get()  // MEMBER FUNCTION
                        {
                              cout << " ENTER NO " << endl;
                              cin >> a ;
                        }

                        void out()
                        {
                              cout << " a = " << a << endl;
                        }
 };
 int   main()   // USER
 {
       test  p ;    //  OBJECT OR VARIABLE

       p . get();  //   INPUT a
       p . out();  //    PRINT a
 }
```

// ADDITION OF TWO NOS USING CLASS AND OBJECT

```
#include<iostream>
using namespace std;

class   sum // programmar
{
        private :   int   a , b , c ; // data member

        public  :
                    void   get() // member function
                    {
                            cout << " enter two no " << endl;
                            cin >> a >> b;
                    }
                    void   cal()
                    {       c = a + b;
                    }
                    void   out()
                    {       cout<< " sum = " << c << endl;
                    }
  };
  int     main() // user
  {       sum   p ; // object --> instance of class
          p . get();
          p . cal();
          p . out();
  }
```

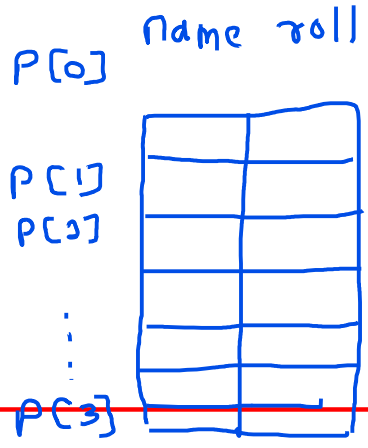


```
// INPUT AND PRINT ONE STUDENT RECORD

#include<iostream>
using namespace std;

  class     student
  {

            private :    char   name[10];
                         int    roll;
            public :
                          void   get()
                         {
                         cout<< " enter name and roll " << endl;
                         cin >> name >> roll;
                         }
                    void   out()
                    {
                          cout << " name = " << name << endl;
                         cout << " roll = " << roll << endl;
                    }
  };
  int    main()

  {      student   p ; // object
         p . get();
         p . out();
        cout<< " size of object = " << sizeof(student) << endl;  // 14
  }
```

ARRAY OF OBJECT

INPUT AND PRINT N RECORDS

```
#include<iostream>
using namespace std;

  class student
  {
            private :    char   name[10];
                         int    roll;
            public  :
                        void   get()
                       {
                        cout<< " enter name and roll " << endl;
                       cin >> name >> roll;
                       }
                      void   out()
                       {
                            cout << " name = " << name << endl;
                            cout << " roll = " << roll << endl;
                       }
  };
  int     main()

  {       student p[10] ; // object
          int  i , n ;
          cout<< " ENTER SIZE " << endl;
          cin >> n;
```



```
            for( i = 0 ; i < n ; i++ )
            {
                  p[i] . get();  // input records
            }
            for( i = 0 ; i < n ; i++ )
            {
                  p[i] . out();  // print records
            }

  }
```

WHY USE PRIVATE DATA MEMBER

```
      class  prime
      {
            public : int  n , t ;

                  void get()
                  {     cin >> n;
                  }
                  void cal()
                  {       ----
                          ----
                  }
                  void out()
                  {
                          ----
                  }
      };
```

```
int    main() // user
{
            prime p;          → object
            p . get();     p . n = 5
            p . cal();     p . t = 1    prime no
            p . t = 0;
            p . out();   not prime  no
}
```

```
2     class   prime
      {
            private : int n , t;
            public:
                  void get()
                  .
                  .
      };
```

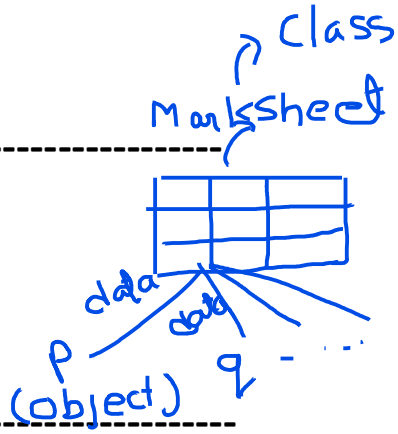
```
3.    class   test1      car
        {
            private :   // data

            public  :
                  void get1() // function
                  {
                        -----
                  }
        };
```

```
class  test2        Home
{
      private :   // data

      public :
            void   get2() // function
            {
                  ------
            }
};

int    main()
{
      test1 p ;    Can  p ;
      test2  q ;   Home q ;

      p . get1() ;  ✓      p get(), ✓

      q . get2() ;  ✓      q get(), ✓

      p . get2() ; X

      q . get1(); X

};
```

OOPS :- C++ , JAVA , VB , PYTHON ,.....

OOPS	C++
DATA HIDING	PRIVATE
ENCAPSULATION	DATA + FUNCTION
ABSTRACTION	INCLUDE ONLY ESSENTIAL ELEMENTS
ABSTRACT DATA TYPE	'CLASS' KEYWORD

e.g. STUDENT , PERSON , ...



---

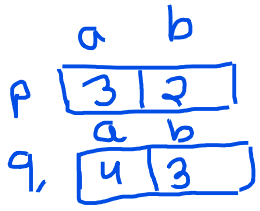
object :- noun , place,
int a ;
class object
student p ;

OBJECT AS FUNCTION AGRUMENT

ADDition of two complex nos (a +i b)

```
#include<iostream>
using  namespace std;

 class    complex
 {
```

```
        private : int a , b;
public:
        void    get()
        {
            cout<<" enter real and imag value "<<endl;
            cin>>a>>b;
        }
        void    out()
        {
                cout<< a << "+i " << b <<endl;
        }
        void    sum ( complex x , complex  y )
        {
                a = x . a + y . a;  // real
                b = x . b + y . b;  // imag.
        }
  };
  int   main()
  {
        complex  p , q , t ; // object

        p. get();   // 3   2
        q. get();   // 4   3

        t. sum( p,q );  // object as function agrument
        p. out();   //  3 +i 2
        q. out();   //  4 +i 3
        t. out();  //   7 +i 5
```



```
    }
```

ADDITION OF TWO TIMES

```
#include<iostream>
using namespace std;
class   time
{
        private:  int  h , m ;

        public  :
                  void   get()
                  {
                        cout<< " ENTER h AND m " << endl;
                        cin >> h >> m;
                  }
                  void   out()
                  {
                       cout << h << ":" << m << endl;
                  }

     void  sum ( time  x , time  y )
     {

m = (x.m + y.m) % 60 ; // m = (40+30) = 70 % 60 = 10  // remainder

h = x.h + y.h + (x.m + y.m)/ 60 ; // h = 2+3+ (70/60) = 2 + 3 + 1 = 6

     }
 };
```

```
int   main()
{
        time p , q , t ;  // object
                          h   m
        p . get();    // 2 : 40

        q . get();   //  3 : 30

        t . sum ( p, q ); // object as fun. agrument

        p . out();   // 2 : 40

        q . out();   // 3 : 30

        t . out();   // 6 : 10

}

//   class   hight-------->
f , in
```