



**Build Your IT Skill**

2020

**ណែនាំស្តាប់ពី OOP(Object Oriented Programming)**

❖ **Abstraction    Inheritance    Polymorphsim**



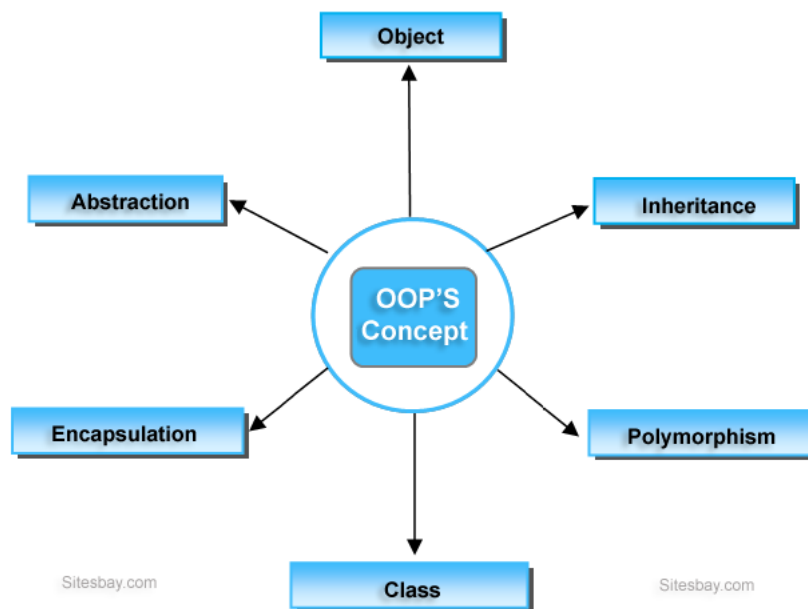
## វិទ្យាសាស្ត្រស្តង់ដារ OOP

### Object Oriented Programming

#### 1. ដូចម្តេចទៅដែលហៅថា OOP?

OOP ឈរលើពាក្យថា Object Oriented Programming សំដៅលើការបង្កើត និង Software ឬ Project ដោយពឹងផ្អែកលើ ការប្រើប្រាស់នូវ Class និង Object ដោយប្រើប្រាស់នូវ Concept ទាំង ៣ដូចជា:

- Encapsulation: សំដៅលើការប្រមូលផ្តុំនូវទិន្នន័យ(Data & Method) ដាក់ក្នុងការគ្រប់ដោយការប្រើប្រាស់ Class។
- Abstraction/Data hiding: សំដៅលើការបិទបាំងនូវទិន្នន័យមួយចំនួនពីផ្នែក ខាងក្រៅនៃ Class។



- Inheritance: សំដៅលើការបង្កើតនូវ Class ថ្មីមួយចេញពី Class ដែលមានស្រាប់ ពេលគឺការទទួលយកទិន្នន័យបន្តគ្នា ឬមរតបន្តគ្នា។
- Polymorphism : សំដៅលើទំរង់ច្រើននៃ Object ឬ Method របស់ Class។



### 1.1. ជូនម្តេចដែលហៅថា Class ?

Class: គឺជាការប្រមូលផ្តុំនូវ Data(Variable) និង Methods(Function) ដាក់ក្នុងការគ្រប់គ្រង ដោយការប្រើប្រាស់ដោយ Class។

```

1
2 Syntax:|
3
4 class Class_Name{
5     private:
6         .....
7         .....
8         Block Data Member
9     public:
10        .....
11        .....
12        Block Function Member
13 };

```

**នៅក្នុង Class អាចមាន Data member និង Method Member ៖**

- Data member: គឺជាប្រភេទ អញ្ញាតិដែលអាចផ្ទុកទិន្នន័យជា បណ្តុះ អាសន្ន។ ការបង្កើតនូវ Data memberគឺគេប្រើប្រាស់ត្រឹមតែ Accessing របស់ Class តែពីរគត់គឺ Private និង Protected ដោយសារតែ គោល ការណ៍របស់ Abstraction។
  - ✓ Private: គឺជាប្រភេទ Accessing ដែលអាច Accessing បានតែក្នុង Class តែប៉ុណ្ណោះ។
  - ✓ Protected: គឺជាប្រភេទ Accessing របស់ Class ដែលអាចប្រើប្រាស់បានក្នុង Class និង ក្រៅ Class ថែមទៀតផង តែ Class នោះត្រូវតែទំនង Inheritance។
  - ✓ Public: គឺជាប្រភេទ Accessing ដែលអាចដាក់ពីមុខ Data member ឬ Function member សំរាប់អាចអោយគេប្រើប្រាស់បានទាំងក្នុង Class និង ក្រៅ Class ផងដែរ។

ឧទាហរណ៍ ១៖ ការបង្កើតស្ត្រូច Data member ជាប្រភេទ public (ឧបសគ្គប្លង់ Abstraction)

```

1 #include<iostream>
2 using namespace std;
3 class Test{
4     public:
5         int x;
6         int y;
7         int z;
8 };
9 int main()
10 { Test obj;
11     cout<<"Input X=";cin>>obj.x;
12     cout<<"Input Y=";cin>>obj.y;
13     cout<<"Input Z=";cin>>obj.z;
14 }
15 
```

C:\Users\Etec Center\Documents\003.exe  
 Input X=100  
 Input Y=200  
 Input Z=3300

ឧទាហរណ៍ ២៖ ការបង្កើតស្ត្រូច Data member ជាប្រភេទ private (ត្រូវបង្កើតប្លង់ Abstraction)

```

1 #include<iostream>
2 using namespace std;
3 class Test{
4     private:
5         int x;
6         int y;
7         int z;
8 };
9 int main()
10 { Test obj;
11     cout<<"Input X=";cin>>obj.x;
12     cout<<"Input Y=";cin>>obj.y;
13     cout<<"Input Z=";cin>>obj.z;
14 }
15 
```

| Line | Col | File                                   | Message                          |
|------|-----|--|----------------------------------|
|      |     | C:\Users\Etec Center\Documents\003.cpp | In function 'int main()':        |
| 5    | 8   | C:\Users\Etec Center\Documents\003.cpp | [Error] 'int Test::x' is private |
| 11   | 29  | C:\Users\Etec Center\Documents\003.cpp | [Error] within this context      |
| 6    | 8   | C:\Users\Etec Center\Documents\003.cpp | [Error] 'int Test::y' is private |
| 12   | 29  | C:\Users\Etec Center\Documents\003.cpp | [Error] within this context      |
| 7    | 8   | C:\Users\Etec Center\Documents\003.cpp | [Error] 'int Test::z' is private |
| 13   | 29  | C:\Users\Etec Center\Documents\003.cpp | [Error] within this context      |



- Function Member: គឺជាប្រភេទ function ដែលប្រកាសនៅក្នុង Class ហើយ  
ដែរវាសំរាប់ធ្វើអ្វីមួយ ដូចជា void input(), void Output(), void setID(),...។  
function អាចជា return function និង Non Return function ជាដើម។  
ឧទាហរណ៍ ១៖

```
1  #include<iostream>
2  using namespace std;
3  class Test{
4      private:
5          int x;
6          int y;
7          int z;
8      public:
9          //Function block
10     void Input()
11     {
12         cout<<"Input X=";cin>>x;
13         cout<<"Input Y=";cin>>y;
14         cout<<"Input Z=";cin>>z;
15     }
16     int Sum()
17     {
18         return x+y+z;
19     }
20 };
21 int main()
22 {
23     Test obj; //Create Object
24     obj.Input(); //Calling function
25     cout<<"\nSum="<<obj.Sum();
26
27 }
```

លទ្ធផលទទួលបាន៖

```
C:\Users\Etec Center\Documents\003.exe
Input X=7
Input Y=6
Input Z=52

Sum=65
```



## ឧទាហរណ៍ ២៖

```
1 #include<iostream>
2 using namespace std;
3 class Test{
4     private:
5         int x;
6         int y;
7         int z;
8     public:
9         //Function block
10        void Input()
11        {
12            cout<<"Input X=";<<cin>>x;
13            cout<<"Input Y=";<<cin>>y;
14            cout<<"Input Z=";<<cin>>z;
15        }
16        int findSum()
17        {
18            return x+y+z;
19        }
20        void output()
21        {
22            cout<<"X="<<x<<endl;
23            cout<<"Y="<<y<<endl;
24            cout<<"Z="<<z<<endl;
25            cout<<"Sum="<<findSum()<<endl; //calling function
26        }
27    };
28    int main()
29    {
30        Test obj; //Create Object
31        obj.Input(); //Calling function
32        obj.output();
33    }
34 }
```

## លទ្ធផលទទួលបាន៖

C:\Users\Etec Center\Documents\003.exe

```
Input X=12
Input Y=3
Input Z=56
X=12
Y=3
Z=56
Sum=71
```

- 1.2. Object: គឺជារបស់ដែលកើតចេញពី Class ដែលមានលទ្ធភាពអោយ ប្រើប្រាស់ នូវ Data និង Method របស់ Class បាន។

```
28 int main()
29 {
30     Test obj; //Create Object
31     obj.Input(); //Calling function
32     obj.output();
33
34     cout<<"Find Sum="<<obj.findSum();|
35
36 }
```

### លំហាត់អនុវត្ត

ចូរធ្វើការបង្កើតនូវ Class មួយឈ្មោះ Employee ដែលមាននូវ ៖

```
2 Data Member
3     int code;
4     string name;
5     char sex;
6     float salary;
7 Function Member
8     void Input()
9     {
10         .....
11
12     }
13     void Output()
14     {
15         .....
16     }
17
```

### 1.3. Property get\_() និង set\_()

ចំពោះការបង្កើតនូវ property set\_() និង get\_() សំរាប់ការប្រើប្រាស់នូវ Data member ដែលជានូវទំរង់ Private។

ដូចអ្នកបានដឹងហើយថា បើ Data របស់អ្នកមានកំរិតជា Private នោះអ្នកពិតជាមិនអាចប្រើប្រាស់នូវ private data member បាននោះទេ។ ដូច្នេះអ្នកអាចធ្វើការប្រើប្រាស់ជាមួយនិង Property ដែលគួរតែជាអ្នកទាញយក Data member ជា Private តាមទំរង់ មិនផ្ទាល់។

ឧទាហរណ៍ ១៖

```
1  #include<iostream>
2  using namespace std;
3  class Students{
4      private:
5          int id;
6          string name;
7          string sex;
8          float score;
9      public:
10     void setID(int i)
11     {
12         this->id=i;
13     }
14     void setName(string n)
15     {
16         this->name=n;
17     }
18     int getID()
19     {
20         return id;
21     }
22     string getName()
23     {
24         return name;
25     }
26 };
27 int main()
28 {
29     Students stu;
30     stu.setID(1001);
31     stu.setName("Sok Vanna");
32     cout<<"ID="<<stu.getID()<<endl;
33     cout<<"Name="<<stu.getName()<<endl;
34 }
35
```

លទ្ធផលទទួលបាន៖

```
1001
Name=Sok Vanna
-----
Process exited after 0.1173 seconds with return value 0
Press any key to continue . . .
```



#### 1.4. ការបង្កើតទូទៅ Constructor

Constructor គឺជាប្រភេទ Function ពិសេសមួយដែលដំណើរការដោយស្វ័យប្រវត្តិ ពេលគឺវាបានភ្ជាប់ជាមួយនិង object រួចតែម្តង លក្ខណៈសំគាល់របស់ Constructor មាន ឈ្មោះដូច Class គ្មាន return type សូម្បីតែ void ។ Constructor ត្រូវបានគេបែងចែកជា ២ប្រភេទ គឺ៖

១) Default Constructor: គឺជាប្រភេទ Function ពិសេសមួយដែលគេប្រើប្រាស់នៅពេលដែល Object មួយកើតឡើងវាទទួលបានតំលៃជាស្វ័យប្រវត្តិ។

ឧទាហរណ៍ ១៖

```
1  #include<iostream>
2  using namespace std;
3  class Students{
4      private:
5          int id;
6          string name;
7          string sex;
8          float score;
9      public:
10         Students()
11         {
12             id=0;
13             name="Null";
14             sex="Null";
15             score=0;
16         }
17         void Output()
18         {
19             cout<<"ID="<<id<<endl;
20             cout<<"Name="<<name<<endl;
21             cout<<"Gender="<<sex<<endl;
22             cout<<"Score="<<score<<endl;
23         }
24     };
25     int main()
26     {
27         Students stu;
28         stu.Output();
29     }
30
31 }
```

លទ្ធផលទទួលបាន៖

```
name=Null
Gender=Null
Score=0
```

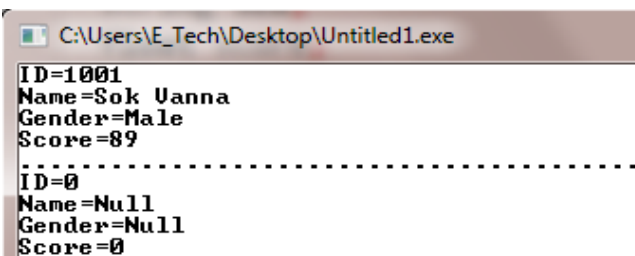
```
Process exited after 0.09539 seconds with return value 0
Press any key to continue . . .
```

២) Constructor with Parameter: គឺជាក្រុមហ៊ុននៃ Constructor ដែលយើងអាចផ្ទេរតំលៃ ពីខាងក្រៅនៃ Class តាម argument Object របស់ Class។

### ឧទាហរណ៍ ១

```
1  #include<iostream>
2  using namespace std;
3  class Students{
4      private:
5          int id;
6          string name;
7          string sex;
8          float score;
9      public:
10         Students()
11         {
12             id=0;
13             name="Null";
14             sex="Null";
15             score=0;
16         }
17         Students(int i,string n,string s, float sc)
18         {
19             id=i;
20             name=n;
21             sex=s;
22             score=sc;
23         }
24
25         void Output()
26         {
27             cout<<"ID="<<id<<endl;
28             cout<<"Name="<<name<<endl;
29             cout<<"Gender="<<sex<<endl;
30             cout<<"Score="<<score<<endl;
31         }
32     };
33
34     int main()
35     {
36         Students stu1;
37         Students stu(1001,"Sok Vanna","Male",89);
38         stu.Output();
39         cout<<"....."<<endl;
40         stu1.Output();
41     }
```

លទ្ធផលទទួលបាន៖



```
C:\Users\E_Tech\Desktop\Untitled1.exe
ID=1001
Name=Sok Vanna
Gender=Male
Score=89
.....
ID=0
Name=Null
Gender=Null
Score=0
```



## លំហាត់អនុវត្ត

ចូរធ្វើការបង្កើត Class មួយឈ្មោះ Employee ដែលមាន data member ដូចជា៖

```
1 class Employee
2   Data Member
3     int code;
4     string name;
5     char sex;
6     float salary;
7   Function Member
8   Employee(){
9
10    .....
11  }
12  Employee(_____,_____,_____,_____){
13    .....
14  }
15  }
16  void Input()
17  {
18    .....
19  }
20  void Output()
21  {
22    .....
23  }
24  }
25
```

### 1.5. Object Array

គឺសំដៅលើប្រភេទ Object ដែលអាចផ្ទុកទិន្នន័យច្រើនលើឈ្មោះតែមួយជាពិសេសទៀតនោះ រាល់ index នីមួយៗ មានទិន្នន័យច្រើនផ្ទុកលើវា។

|                 |                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|-----------------|
| 0               | 1               | 2               | 3               | 4               |
| x   y   methods | x   y   methods | x   y   methods | x   y   methods | x   y   methods |

#### ឧទាហរណ៍៖

```
1 #include<iostream>
2 #include<conio.h>
3 using namespace std;
4 class Employee{
5     private:
6         int id;
7         string name;
8         string sex;
9         double salary;
10    public:
11        Employee()
12        {
13            id=0;
14            name="N/A";
15            sex="N/A";
16            salary=0;
17        }
18        int getID()
19        {
20            return id;
21        }
22        void setName(string n)
23        {
24            name=n;
25        }
26        void setGender(string s)
27        {
28            sex=s;
29        }
30        void setSalary(double s)
31        {
32            salary=s;
33        }
```

```
34 void input()
35 {
36     cout<<"Input ID=";cin>>id;
37     cout<<"Input Name=";cin>>name;
38     cout<<"Input Sex=";cin>>sex;
39     cout<<"Input Salary=";cin>>salary;
40 }
41 void output()
42 {
43     cout<<id<<" "<<name<<" "<<sex<<" "<<salary<<endl;
44 }
45 }
46 };
47 int search(Employee emp[20],int n)
48 { int scode,b=0;
49   int index=-1;
50   cout<<" Input Code to Search=";cin>>scode;
51   for(int i=0;i<n;i++)
52   {
53       if(emp[i].getID()==scode)
54       { b=1;
55         emp[i].output();
56         index=i;
57         break;
58       }
59   }
60   if(b==0) cout<<"Search not found\n";
61   return index;
62 }
63
64 int main()
65 {
66     Employee emp[20];
67     int n,op,i;
68     do{
69         cout<<"1. Input"<<endl;
70         cout<<"2. Output"<<endl;
71         cout<<"3. Search"<<endl;
72         cout<<"4. Update"<<endl;
73         cout<<"5. Sort"<<endl;
74         cout<<"Choose One(1-5)=";cin>>op;
```

```
75      switch(op)
76      {
77      case 1:{
78          cout<<"Input Number of Array=";cin>>n;
79          for(i=0;i<n;i++)
80          {
81              emp[i].input();
82          }
83          }break;
84      case 2:{
85          for(i=0;i<n;i++)
86          {
87              emp[i].output();
88          }
89          }break;
90      case 3:{
91          int index=search(emp,n);
92          if(index==-1)
93              cout<<"Error Search\n";
94          else
95              cout<<" ";
96          }break;
97      case 4:{
98          string nname,nsex;
99          double nsalary;
100         int index=search(emp,n);
101         if(index==-1)
102             cout<<"Error Search\n";
103         else
104             cout<<"Input New Name=";cin>>nname;
105             cout<<"Input New Gender=";cin>>nsex;
106             cout<<"Input New Salary=";cin>>nsalary;
107             emp[index].setName(nname);
108             emp[index].setGender(nsex);
109             emp[index].setSalary(nsalary);
110             cout<<"Update Completed....!";
111         }break;
112     case 5:{
113         Employee temp;
114         for(i=0;i<n;i++)
115         {
116             for(j=i+1;j<n;j++)
117             {
118                 if(emp[i].getID()>emp[j].getID())
119                 {
120                     temp=emp[i];
121                     emp[i]=emp[j];
122                     emp[j]=temp;
123                 }
124             }
125         }
126         }break;
127     }
128 }
129 cout<<"\nPress Enter to Continue....!";
130 }while(getch()!='\n');
131 return 0;
132 }
```

## 1.6. Object Pointer

នៅក្នុង C++ អ្នកក៏អាចយក pointer របស់ C មកបង្កើតជា Dynamic Memory

Allocation តាមរយៈការប្រើប្រាស់នូវ Malloc(), Calloc() និង realloc() បានផងដែរ។

```
36 Employee *pemp;  
37 cout<<"Input Number of Array=";cin>>n;  
38 pemp=(Employee*)calloc(n,sizeof(Employee));
```

ចូរធ្វើការបង្កើតនូវ Object លក្ខណៈ ជា Dynamic Memory Allocation?

```
1 #include<iostream>  
2 #include<conio.h>  
3 #include<stdlib.h>  
4 #include<string.h>  
5 using namespace std;  
6 class Employee{  
7     private:  
8         int id;  
9         char name[20];  
10        char sex[10];  
11        double salary;  
12    public:  
13        Employee()  
14    {  
15        id=0;  
16        strcpy(name,"N/A");  
17        strcpy(sex,"N/A");  
18        salary=0;  
19    }  
20    void input()  
21    {  
22        cout<<"Input ID=";cin>>id;  
23        cout<<"Input Name=";cin>>name;  
24        cout<<"Input Sex=";cin>>sex;  
25        cout<<"Input Salary=";cin>>salary;  
26    }  
27    void output()  
28    { |  
29        cout<<id<<" "<<name<<" "<<sex<<" "<<salary<<endl;  
30    }  
31 }  
32 };  
33 int main()  
34 {  
35     int n,i;  
36     Employee *pemp;  
37     cout<<"Input Number of Array=";cin>>n;  
38     pemp=(Employee*)calloc(n,sizeof(Employee));  
39     for(i=0;i<n;i++)  
40     {  
41         (pemp+i)->input();  
42     }  
43     for(i=0;i<n;i++)  
44     {  
45         (pemp+i)->output();  
46     }  
47     return 0;  
48 }
```



## លំហាត់អនុវត្ត

តាមរយៈឧទាហរណ៍ខាងលើ ចូរធ្វើការបង្កើតចេញជា Menu ដូចខាងក្រោម៖

```
1
2      >>>>>>> Menu <<<<<<<<<<<
3      | 1. Input
4      | 2. Output
5      | 3. Search
6      | 4. Update
7      | 5. Sort
8      | 6. Exit
9      Choose One(1-6)=_____
10
11
```

ដោយប្រើប្រាស់នូវ Pointer ?

Good Luck!