



**Build your IT Skill**

# ណែនាំ C++ Programming



**ប្រើប្រាស់ដោយ៖**

**គ្រូអោយបីចំណុច**

**អនុបណ្ឌិត RUPP, TKU**

**(Software Engineering)**

**Assistands: SOUS SEYHA,**

**LOUN VIRAK , PHEARUM SIMMENG**

**២០២៣**



# ណែនាំអោយស្គាល់ពី C++ Programming

## I. អ្វីទៅជា C++ Programming?

C++ គឺជាភាសាបច្ចេកទេសដែលចាត់ទុកថា ជាភាសាគ្រឹះសំរាប់អ្នកចង់ចាប់ ផ្តើមសរសេរ កូដ លើកដំបូង។ C++ វាបានភ្ជាប់ជាមួយគ្រឹះនៃការសរសេរ កូដដូចជា Variable, Control Statement, Loop Statement, Function និង Array ជាដើម។ C++ វាក៏មានប្រភេទគ្រឹះដែល សំខាន់មួយបែបទៀត នោះគឺ OOP( Object Oriented Programming ) សម្រាប់ការបង្កើត Project ដែលប្រើប្រាស់នូវ Class និង Object ។

```

1  //Structure C++ Programming
2
3  #include<iostream>
4  using namespace std;
5  int main()
6  {
7      >>>> Block Statement(s);<<<<
8
9      return 0;
10 }
```

កម្មវិធីដែលអាចសរសេរកូដជាមួយនិង C++៖





## II. ការបង្ហាញទិន្នន័យរបស់ C++ Programming

ដើម្បីធ្វើការបង្ហាញទិន្នន័យលើ Monitor របស់ C++ យើងអាចប្រើប្រាស់នូវ Stream function ចំនួនពីររូបភាពដូចជា៖

ឧទាហរណ៍ ១៖

```

1  #include<iostream>
2  using namespace std;
3  int main()
4  {   cout<<">>>>>>>>ETEC i.T Professional Training Center<<<<<<<"<<endl;
5      cout<<"ID=1001"<<endl;
6      cout<<"Name=Sok Vanda"<<endl;
7      cout<<"Gender=Male"<<endl;
8      cout<<"Address: Phnom Penh"<<endl;
9      |
10     return 0;
11 }
```

លទ្ធផលទទួលបាន៖

```

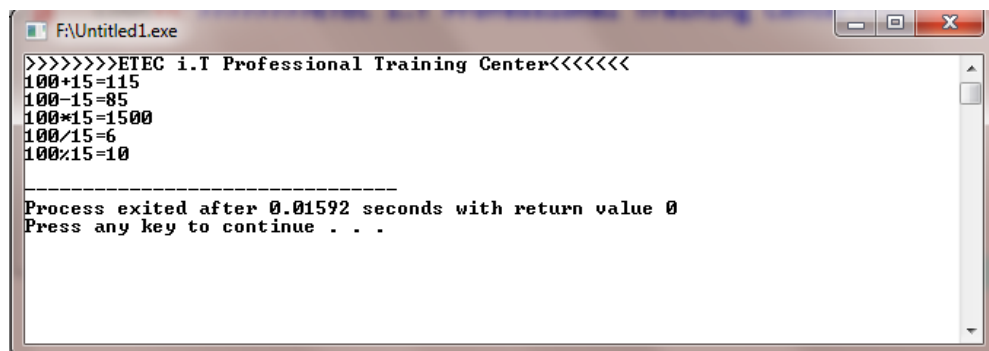
F:\Untitled1.exe
>>>>>>>>ETEC i.T Professional Training Center<<<<<<<
ID=1001
Name=Sok Vanda
Gender=Male
Address: Phnom Penh

Process exited after 0.04119 seconds with return value 0
Press any key to continue . . .
```

## ឧទាហរណ៍ ២៖

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 { cout<<">>>>>>>ETEC i.T Professional Training Center<<<<<<<"<<endl;
5
6     cout<<"100+15="<<100+15<<endl;
7     cout<<"100-15="<<100-15<<endl;
8     cout<<"100*15="<<100*15<<endl;
9     cout<<"100/15="<<100/15<<endl;
10    cout<<"100%15="<<100%15<<endl;
11    return 0;
12 }
```

## លទ្ធផលទទួលបាន៖



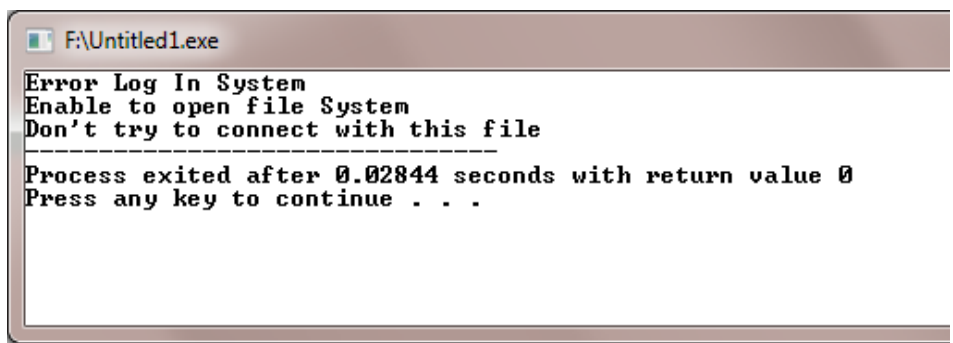
```
F:\Untitled1.exe
>>>>>>>ETEC i.T Professional Training Center<<<<<<<
100+15=115
100-15=85
100*15=1500
100/15=6
100%15=10

Process exited after 0.01592 seconds with return value 0
Press any key to continue . . .
```

## ឧទាហរណ៍ ៣៖ ការប្រើប្រាស់នូវ cerr សំរាប់បង្ហាញទិន្នន័យ Error។

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     cerr<<"Error Log In System"<<endl;
6     cerr<<"Enable to open file System"<<endl;
7     cerr<<"Don't try to connect with this file";
8     return 0;
9 }
```

## លទ្ធផលទទួលបាន៖



```
F:\Untitled1.exe
Error Log In System
Enable to open file System
Don't try to connect with this file

Process exited after 0.02844 seconds with return value 0
Press any key to continue . . .
```

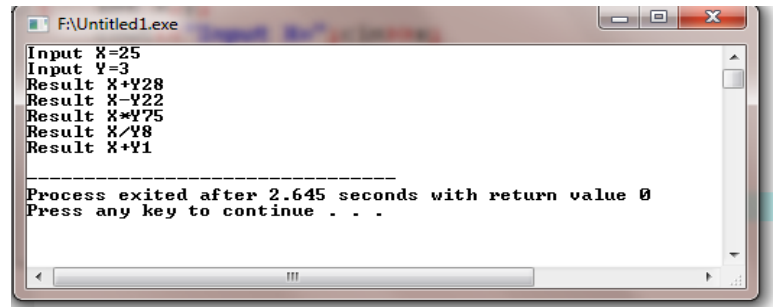
### III. ការបញ្ចូលទិន្នន័យរបស់ C++ Programming

ដើម្បីធ្វើបញ្ចូលទិន្នន័យរបស់ C++ គេអាចប្រើប្រាស់នូវ Stream Input ចំនួន ពីរគឺ ៖

ឧទាហរណ៍ ១៖

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {   int x,y;
5     cout<<"Input X=";<<cin>>x;
6     cout<<"Input Y=";<<cin>>y;
7     cout<<"Result X+Y"<<x+y<<endl;
8     cout<<"Result X-Y"<<x-y<<endl;
9     cout<<"Result X*Y"<<x*y<<endl;
10    cout<<"Result X/Y"<<x/y<<endl;
11    cout<<"Result X+Y"<<x%y<<endl;
12    return 0;
13 }
```

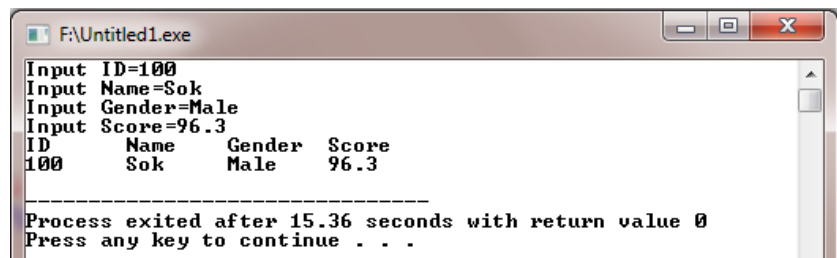
លទ្ធផលទទួលបាន៖



ឧទាហរណ៍ ២៖

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5     int id;
6     string name;
7     string sex;
8     float score;
9     cout<<"Input ID=";<<cin>>id;
10    cout<<"Input Name=";<<cin>>name;
11    cout<<"Input Gender=";<<cin>>sex;
12    cout<<"Input Score=";<<cin>>score;
13    cout<<"ID\tName\tGender\tScore\n";
14    cout<<id<<"\t"<<name<<"\t"<<sex<<"\t"<<score<<endl;
15    return 0;
16 }
```

លទ្ធផលទទួលបាន៖



### ឧទាហរណ៍ ៣៖ ការបញ្ចូលតួអក្សរអាចដកឃ្លានបាន

```
Untitled1.cpp
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int id;
6      string name;
7      string sex;
8      float score;
9      cout<<"Input ID=";cin>>id;
10     cin.ignore();
11     cout<<"Input Name=";getline(std::cin, name);
12     cout<<"Input Gender=";cin>>sex;
13     cout<<"Input Score=";cin>>score;
14     cout<<"ID\tName\tGender\tScore\n";
15     cout<<id<<"\t"<<name<<"\t"<<sex<<"\t"<<score<<endl;
16     return 0;
17 }
```

#### IV. ការប្រើប្រាស់នូវអញ្ញាតិ C++ Programming

អញ្ញាតិគឺជាឈ្មោះមួយប្រភេទដែលបង្កើតចេញពីប្រភេទទិន្នន័យណាមួយច្បាស់លាស់ ហើយប្រើប្រាស់សំរាប់ផ្ទុកទិន្នន័យជាបណ្តុះអាសន្នលើ RAM។

```
3  #include<iostream>
4  using namespace std;
5  //Global Variable
6  Datatype Variable_Name1;
7
8  OR
9
10     Datatype Variable_Name1,Variable_Name2,....;
11
12  int main()
13  {
14      //Local Variable
15      Datatype Variable_Name1;
16
17      OR
18
19      Datatype Variable_Name1,Variable_Name2,....;
20
21      return 0;
22  }
```

ឧទាហរណ៍ ៖

```

3  #include<iostream>
4  using namespace std;
5  int main()
6  {  int x,y;
7     long a,b;
8     float c,d;
9     double e,f;
10    string str1,str2,str3; |
11
12
13    return 0;
14 }
```

### ប្រភេទនៃ Datatype

```

1  Datatype String
2  char: store value with single character
3  string: store value with character but allow many character
4  Datatype Numeric
5  int: store value with Integer
6  long: store value with integer but allow many
7  float: store value with floating value
8  double: store value with floating value but allow many
9  Boolean:
10 bool: store value with true and false
11
12
```

### ប្រភេទនៃ Operator

1. Arithmetic Operator

operator	description
+	addition
-	subtraction
*	multiplication
/	division
%	modulo

2. Comparison Operator

operator	description
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to



## 3. Logical Operator

<b>&amp;&amp; OPERATOR (and)</b>		
a	b	a && b
true	true	true
true	false	false
false	true	false
false	false	false

<b>   OPERATOR (or)</b>		
a	b	a    b
true	true	true
true	false	true
false	true	true
false	false	false

**លំហាត់អនុវត្ត (ផ្នែកទី ១)**

១) ចូរសរសេរកូដដោយគេអាចបញ្ចូលព័ត៌មាន ដូចជា id(int), name(string), Sex(string), Salary(float) និង address(String) បន្ទាប់មកបង្ហាញទិន្នន័យទាំងនោះ ចេញមកលើអេក្រង់វិញ។

២) ចូរសរសេរកូដសំរាប់អោយគេបញ្ចូលនូវពិន្ទុដូចជា Score1(float), Score2(float), Score3(float), score4(float) និង score5(float) បន្ទាប់មកបង្ហាញនូវពិន្ទុ សរុប និង មធ្យមភាគជាការស្រេច។

**លំហាត់អនុវត្ត (ផ្នែកទី ២)**

១. ចូរសរសេរកូដដោយគេបញ្ចូលផលិតផល ដូចជា code(int), name(string), qty(int), price(double) បន្ទាប់មកបង្ហាញព័ត៌មាន ទាំងនោះ មកក្រៅវិញ រួមទាំង total(double) និង payment(double) ?

តំលៃសរុប (Total)	បញ្ចុះ តំលៃ %
1 ដល់ 10\$	10%
10 ដល់ 20\$	20%
20 ដល់ 30\$	30%
30 ដល់ 40\$	40%
40 ដល់ 50\$	50%
50 ដល់ 60\$	60%
60\$-	70%





២) ចូរសរសេរកូដអោយគេបញ្ចូលពិន្ទុ ៥មុខដូចជា score1(float), score2(float), score3(float), score4(float) និង score5(float) បន្ទាប់មកបង្ហាញទិន្នន័យនោះ ចេញមកក្រៅវិញរួមមាន score ទាំង ៥មុខ និងពិន្ទុសរុប(total), មធ្យមភាគ(average) និង Grade ?

មធ្យមភាគ/Average	និទេស/Grade
90-100	A
80-90	B
70-80	C
60-70	D
50-60	E
0-50	F

### លំហាត់អនុវត្តន៍

1) ចូរបង្កើត function ជា Return ឬ Non Return Function សំរាប់ការប្រើប្រាស់នូវ

Loop ដូចខាងក្រោម៖

1.  $1+2+3+....+N$
2.  $2+4+6+....+N$
3.  $3+5+7+....+N$
4.  $1!+2!+3!+...+N!$

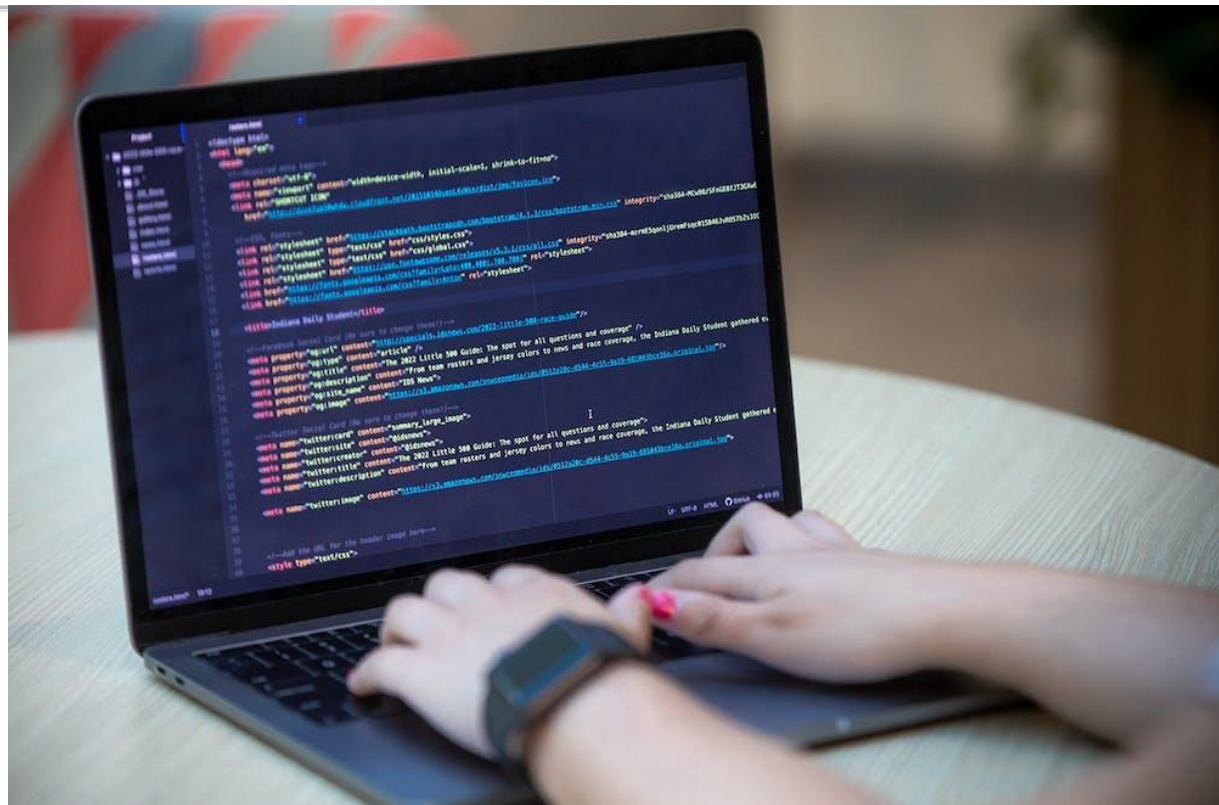
ដោយប្រើប្រាស់នូវ For Loop or While Loop or Do While Loop ?



**Build your IT Skill**

# ណែនាំ OOP (Object Oriented Programming)

Encapsulation    Abstaction    Class /Object



## វិទ្យាសាស្ត្រស្តង់ដារ OOP

### Object Oriented Programming

#### 1. ជួបម្តេចទៅដែលហៅថា OOP?

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

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



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

### 1.1. ដូចម្តេចដែលហៅថា 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 មាន Data member និង Method Member ៖**

- Data member: គឺជាប្រភេទ អញ្ញាតិដែលអាចផ្ទុកទិន្នន័យជា បណ្តុះ អាសន្ន។ ការបង្កើតនូវ Data member គឺគេប្រើប្រាស់ត្រឹមតែ Accessing របស់ Class តែពីរគត់គឺ Private និង Protected ដោយសារជា គោល ការណ៍របស់ Abstraction ។
  - ✓ Private: គឺជាប្រភេទ Accessing ដែលអាច Accessing បានតែក្នុង Class តែប៉ុណ្ណោះ។
  - ✓ Protected: គឺជាប្រភេទ Accessing របស់ Class ដែលអាច ប្រើប្រាស់បានក្នុង Class និង ក្រៅ Class ថែមទៀតផង តែ Class នោះត្រូវតែទំនង Inheritance ។
  - ✓ Public: គឺជាប្រភេទ Accessing ដែលអាចដាក់ពីមុខ Data member ឬ Function member សំរាប់អាចអោយគេប្រើ ប្រាស់បានទាំងក្នុង Class និង ក្រៅ Class ផងដែរ។

ឧទាហរណ៍ ១៖ ការបង្កើតស្ត្រូង Data member ជាប្រភេទ public ( ខុសឆ្ងាប់ 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៖ ការបង្កើតស្ត្រូង Data member ជាប្រភេទ private (ត្រូវឆ្ងាប់ Abstraction )

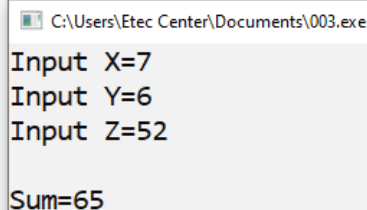
```
1 #include<iostream>
2 using namespace std;
3 class Test{
4     private:
5         int x;
6         int y;
7         int z;
8 };
9 int main()
10 { Test obj;
11     cout<<"Input X=";cin>>obj.x;
12     cout<<"Input Y=";cin>>obj.y;
13     cout<<"Input Z=";cin>>obj.z;
14 }
```

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 ហើយដើរតាមរូបមន្តដូចជា 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();
```

### 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
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 ដែលមានលទ្ធភាពអាច 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     void Output()
13     {
14         .....
15     }
16
17
```





ចូរធ្វើការបង្កើតស្ត្រូច Class មួយឈ្មោះ Student ដែលមានស្ត្រូច ៖

```
class Student{
    private:
        int id;
        char name[20],gender;
        float score1,score2,score3;
    public:
        void Input(){
            .....
        }
        double Total(){
            .....
        }
        double Average(){
            .....
        }
        char Grade(){
            .....
        }
        void Output(){
            .....
        }
};
```

### 1.3. Property get\_( ) និង set\_( )

ចំពោះការបង្កើតស្ត្រូច property set\_( ) និង get\_( ) សំរាប់ការប្រើប្រាស់ស្ត្រូច Data member ដែលជានូវទំរង់ Private។

ដូចអ្នកបានដឹងហើយថា បើ Data របស់អ្នកមានកំរិតជា Private នោះអ្នកពិតជាមិនអាចប្រើប្រាស់ស្ត្រូច private data member បាននោះទេ។ ដូច្នេះអ្នកអាចធ្វើការប្រើប្រាស់ជាមួយនិង Property ដែលគួរតែទីជាអ្នកទាញយក 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៖

```
C:\Users\E_Tech\Desktop\Untitled1.exe
ID=1001
Name=Sok Vanna

Process exited after 0.1173 seconds with return value 0
Press any key to continue . . .
```

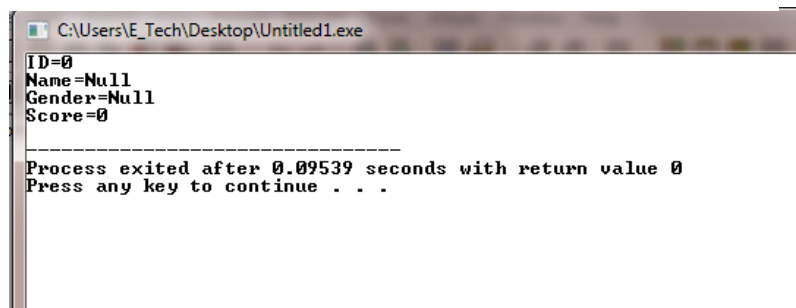
#### 1.4. ការបង្កើតទូទៅ Constructor

Constructor គឺជាប្រភេទ Function ពិសេសមួយដែលដំណើរការដោយស្វ័យប្រវត្តិ ពេល គឺវាបានភ្ជាប់ជាមួយនិង object រួចតែម្តង លក្ខណៈ សំគាល់របស់ 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 }
```



```
C:\Users\E_Tech\Desktop\Untitled1.exe
ID=0
Name=Null
Gender=Null
Score=0

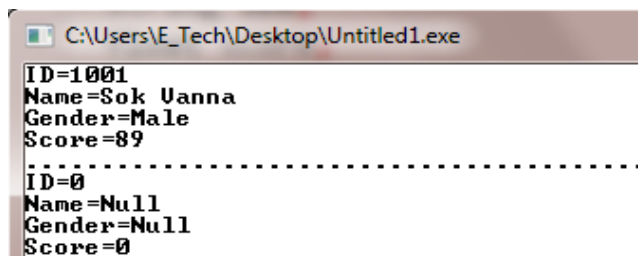
Process exited after 0.09539 seconds with return value 0
Press any key to continue . . .
```

២) Constructor with Parameter: គឺជាប្រភេទនៃ 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 }
```

លទ្ធផលអនុលោម៖



```
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 ដែលអាចផ្ទុកទិន្នន័យច្រើនលើឈ្មោះតែមួយជាពិសេស ទៀតនោះ រាល់ index នីមួយៗ មានទិន្នន័យច្រើនផ្ទុកលើវា។

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

#### ឧទាហរណ៍ទី១៖

```

1  #include<iostream>
2  #include<conio.h>
3  #include<iomanip>
4  #include<string.h>
5  using namespace std;
6  class Employee{
7  private:
8      int id;
9      char name[20],gender[10],position[30];
10     float salary;
11 public:
12     int getID(){
13         return id;
14     }
15     char *getName(){
16         return name;
17     }
18     void Input(){
19         cout<<" Input ID      =";cin>>id;
20         cout<<" Input Name    =";cin.ignore();cin.getline(name,20);
21         cout<<" Input Gender  =";cin>>gender;
22         cout<<" Input Position =";cin.ignore();cin.getline(position,30);
23         cout<<" Input Salary  =";cin>>salary;
24     }
25     void Output(){
26         cout<<setw(12)<<id
27             <<setw(14)<<name
28             <<setw(16)<<gender
29             <<setw(18)<<position
30             <<setw(15)<<salary<<"$"
31             <<endl;
32     }
33 };
34 void Header(){
35     cout<<setw(12)<<"ID"
36         <<setw(14)<<"Name"
37         <<setw(16)<<"Gender"
38         <<setw(18)<<"Position"
39         <<setw(16)<<"Salary"
40         <<endl;
41 }
42 
```

```
43 int main(){
44     Employee em[50];
45     int i,n,op;
46     do{
47         cout<<endl<<"----- M E N U -----"<<endl;
48         cout<<" 1.Input"<<endl;
49         cout<<" 2.Output"<<endl;
50         cout<<" 3.Search"<<endl;
51         cout<<" 4.Update"<<endl;
52         cout<<" 5.Delete"<<endl;
53         cout<<" 6.Insert"<<endl;
54         cout<<" 7.Sort"<<endl;
55         cout<<" 8.Add"<<endl;
56         cout<<"-----"<<endl;
57         cout<<" Please choose one option =";cin>>op;
58         switch(op){
59             case 1:{
60                 do{
61                     cout<<" Input number of employee =";cin>>n;
62                 }while(n>50);
63                 for(i=0;i<n;i++){
64                     cout<<"\n----- Employee #"<<i+1<<"-----"<<endl;
65                     em[i].Input();
66                 }
67             }break;
68             case 2:{
69                 Header();
70                 for(i=0;i<n;i++){
71                     em[i].Output();
72                 }
73             }break;
74             case 3:{
75                 int sid;
76                 bool b=false;
77                 cout<<" Input employee's id for search =";cin>>sid;
78                 for(i=0;i<n;i++){
79                     if(sid==em[i].getID()){
80                         Header();
81                         em[i].Output();
82                         b=true;
83                     }
84                 }
85             }
```

```
85 □ if(b==false){
86 □     cout<<"\n Search id not found..."<<endl;
87 □ }
88 □ }break;
89 □ case 4:{
90 □     char sname[20],uposition[30];
91 □     bool b=false;
92 □     cout<<" Input employee's name for search to update =";
93 □     cin.ignore();cin.getline(sname,20);
94 □     for(i=0;i<n;i++){
95 □         if(stricmp(sname,em[i].getName()==0){ //strcmp,strcmp
96 □             em[i].Input();
97 □             b=true;
98 □             cout<<"\n Update successful..."<<endl;
99 □         }
100 □     }
101 □     if(b==false){
102 □         cout<<"\n Search name not found..."<<endl;
103 □     }
104 □ }break;
105 □ case 5:{
106 □     int sid;
107 □     bool b=false;
108 □     cout<<" Input employee's id for search to delete =";cin>>sid;
109 □     for(i=0;i<n;i++){
110 □         if(sid==em[i].getID()){
111 □             for(int j=i;j<n-1;j++){
112 □                 em[j]=em[j+1];
113 □             }
114 □             n--;
115 □             b=true;
116 □             cout<<"\n Delete successful..."<<endl;
117 □         }
118 □     }
119 □     if(b==false){
120 □         cout<<"\n Search id not found...\n"<<endl;
121 □     }
122 □ }break;
123 □ case 6:{
124 □     int sid;
125 □     bool b=false;
126 □     cout<<" Input employee's id for search to insert =";cin>>sid;
```



```
127   for(i=0;i<n;i++){
128       if(sid==em[i].getID()){
129           for(int j=n;j>i;j--){
130               em[j]=em[j-1];
131           }
132           n++;
133           em[i].Input();
134           b=true;
135           cout<<"\n Insert completed..."<<endl;
136           break;
137       }
138   }
139   if(b==false){
140       cout<<"\n Search id not found..."<<endl;
141   }
142   }break;
143   case 7:{
144       Employee temp;
145       for(i=0;i<n-1;i++){
146           for(int j=i+1;j<n;j++){
147               if(strcmp(em[i].getName(),em[j].getName())>0){
148                   temp=em[i];
149                   em[i]=em[j];
150                   em[j]=temp;
151               }
152           }
153       }
154       cout<<"\n Sort successful..."<<endl;
155   }break;
156   case 8:{
157       int m;
158       cout<<" Input number of employee to add =";cin>>m;
159       for(i=n;i<n+m;i++){
160           em[i].Input();
161       }
162       n+=m;
163       cout<<"\n Add successful..."<<endl;
164   }break;
165   }
166   cout<<"\n Press {enter} to continue...";
167   }while(getch()==13);
168   return 0;
169 }
```

## ឧទាហរណ៍ទី២៖

```
1  #include<iostream>
2  #include<iomanip>
3  #include<conio.h>
4  using namespace std;
5  int i;
6  class Lecture{
7      private:
8          int id;
9          string name,gender,subject;
10     public:
11         int getID();
12         string getName();
13         void setName(string n);
14         void Input();
15         void Output();
16 };
17 string Lecture::getName(){
18     return name;
19 }
20 int Lecture::getID(){
21     return id;
22 }
23 void Lecture::setName(string n){
24     name=n;
25 }
26 void Lecture::Input(){
27     cout<<" Input ID      =";cin>>id;
28     cout<<" Input Name    =";cin.ignore();getline(cin,name);
29     cout<<" Input Gender  =";cin>>gender;
30     cout<<" Input Subject =";cin.ignore();getline(cin,subject);
31 }
32 void Lecture::Output(){
33     cout<<setw(12)<<id
34         <<setw(14)<<name
35         <<setw(16)<<gender
36         <<setw(17)<<subject
37         <<endl;
38 }
```



```
39 void Header(){
40     cout<<setw(12)<<"ID"
41         <<setw(14)<<"Name"
42         <<setw(16)<<"Gender"
43         <<setw(17)<<"Subject"
44         <<endl;
45 }
46 int Search(Lecture lec[],int n){
47     int sid,index=-1;
48     cout<<" Input lecture's id for search =";cin>>sid;
49     for(i=0;i<n;i++){
50         if(sid==lec[i].getID()){
51             index=i;
52         }
53     }
54     return index;
55 }
56 void Update(Lecture lec[],int n){
57     int index=Search(lec,n);
58     string uname;
59     if(index==-1){
60         cout<<"\n Search id not found..."<<endl;
61     }else{
62         cout<<" Input lecture's name for update =";cin.ignore();getline(cin,uname);
63         lec[index].setName(uname);
64         cout<<"\n Update successful..."<<endl;
65     }
66 }
67 void Delete(Lecture lec[],int *n){
68     int index=Search(lec,*n);
69     if(index==-1){
70         cout<<"\n Search id not found..."<<endl;
71     }else{
72         for(int j=index;j<*n;j++){
73             lec[j]=lec[j+1];
74         }
75         *n=*n-1;
76         cout<<"\n Delete completed..."<<endl;
77     }
```

```
78 }
79 void Insert(Lecture lec[],int *n){
80     int index=Search(lec,*n);
81     if(index==-1){
82         cout<<"\n Search id not found..."<<endl;
83     }else{
84         for(int j=*n;j>index;j--){
85             lec[j]=lec[j-1];
86         }
87         *n=*n+1;
88         lec[index].Input();
89         cout<<"\n Insert succesful"<<endl;
90     }
91 }
92 void Sort(Lecture lec[],int n){
93     Lecture temp;
94     for(i=0;i<n-1;i++){
95         for(int j=i+1;j<n;j++){
96             if(lec[i].getID() > lec[j].getID()){
97                 temp=lec[i];
98                 lec[i]=lec[j];
99                 lec[j]=temp;
100             }
101         }
102     }
103     cout<<"\n Sort successful..."<<endl;
104 }
105 void Add(Lecture lec[],int *n){
106     int m;
107     cout<<" Input number of lecturer =";cin>>m;
108     for(i=*n;i<*n+m;i++){
109         lec[i].Input();
110     }
111     *n=*n+m;
112     cout<<"\n Add successful..."<<endl;
113 }
114 int main(){
115     Lecture lec[50];
116     int n;
```

```
117 | char op;
118 | do{
119 |     cout<<"\n=====>> M e n u <<===== "<<endl;
120 |     cout<<" A.Input"<<endl;
121 |     cout<<" B.Output"<<endl;
122 |     cout<<" C.Search"<<endl;
123 |     cout<<" D.Update"<<endl;
124 |     cout<<" E.Delete"<<endl;
125 |     cout<<" F.Insert"<<endl;
126 |     cout<<" G.Sort"<<endl;
127 |     cout<<" H.Add"<<endl;
128 |     cout<<"===== "<<endl;
129 |     cout<<" Please choose one option =";cin>>op;
130 |     switch(op){
131 |         case 'a':
132 |         case 'A':{
133 |             cout<<" Input number of lecture=";cin>>n;
134 |             for(i=0;i<n;i++){
135 |                 cout<<"\n===== Lecture #"<<i+1<<" ===== "<<endl;
136 |                 lec[i].Input();
137 |             }
138 |             }break;
139 |         case 'b':
140 |         case 'B':{
141 |             Header();
142 |             for(i=0;i<n;i++){
143 |                 lec[i].Output();
144 |             }
145 |             }break;
146 |         case 'c':
147 |         case 'C':{
148 |             int index=Search(lec,n);
149 |             if(index==-1){
150 |                 cout<<"\n Search id not found...!"<<endl;
151 |             }else{
152 |                 Header();
153 |                 lec[index].Output();
154 |             }
155 |             }break;
```



```
156 | case 'd':
157 | case 'D':{
158 |     Update(lec,n);
159 | }break;
160 | case 'e':
161 | case 'E':{
162 |     Delete(lec,&n);
163 | }break;
164 | case 'f':
165 | case 'F':{
166 |     Insert(lec,&n);
167 | }break;
168 | case 'g':
169 | case 'G':{
170 |     Sort(lec,n);
171 | }break;
172 | case 'h':
173 | case 'H':{
174 |     Add(lec,&n);
175 | }break;
176 | }
177 | cout<<"\n Press {enter} to continue...";
178 | }while(getch()==13);
179 | return 0;
180 | }
```

## 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=">>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=">>id;  
23            cout<<"Input Name=">>name;  
24            cout<<"Input Sex=">>sex;  
25            cout<<"Input Salary=">>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=">>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 }
```



## លំហាត់អនុវត្តទី១

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 ?





## លំហាត់អនុវត្តទី២

គេមានclass មួយឈ្មោះ **Worker** ដែលមាននូវData member អត្តលេខ(id), ឈ្មោះ(name), ភេទ(gender), ប្រាក់ខែ(salary), ប្រាក់កម្រៃក្នុងមួយម៉ោង(rate), និងចំនួនម៉ោងធ្វើការ(hour)។

1). **Input:** បញ្ចូលព័ត៌មានរបស់កម្មករចំនួន n ធាតុពី keyboard ។

2). **Income:** គណនាកម្រៃប្រាក់ចំណូលសរុបរបស់កម្មករម្នាក់ៗ ដោយយកប្រាក់ខែ បូកនឹងចំនួនម៉ោងធ្វើការ គុណនឹងប្រាក់កម្រៃក្នុងមួយម៉ោង ( $\text{Income} = \text{salary} + (\text{rate} * \text{hour})$ ) ។

3). **Output:** បង្ហាញព័ត៌មានរបស់កម្មករទាំងអស់មកលើអេក្រង់។

4). **Search** ស្វែងរកព័ត៌មានរបស់កម្មករណាម្នាក់។

5). **Maximum Income:** បង្ហាញព័ត៌មានរបស់កម្មករដែលមានប្រាក់ចំណូលខ្ពស់ជាងគេ។

6). **Minimum Income:** បង្ហាញព័ត៌មានរបស់កម្មករដែលមានប្រាក់ចំណូលទាបជាងគេ។

7). **Equal Income:** បង្ហាញព័ត៌មានរបស់កម្មករដែលមានប្រាក់ចំណូលស្មើគ្នា។

8). **Sort:** តំរៀបព័ត៌មានរបស់កម្មករតាមលំដាប់កើនជាបន្តបន្ទាប់។

9). **Sum:** បូកសរុបប្រាក់ចំណូលដែលត្រូវទៅកម្មករទាំងអស់។

10). **Update:** បន្ថែម 15% ទៅលើប្រាក់ចំណូលរបស់កម្មករដែលទទួលបានប្រាក់ចំណូលតិចជាង 150\$ ។

11). **Insert:** បន្ថែមនូវចំនួនកម្មករទំនាក់ ត្រង់Record ណាមួយ។

12). **Delete:** លុបព័ត៌មានរបស់កម្មករណាម្នាក់ចេញ។