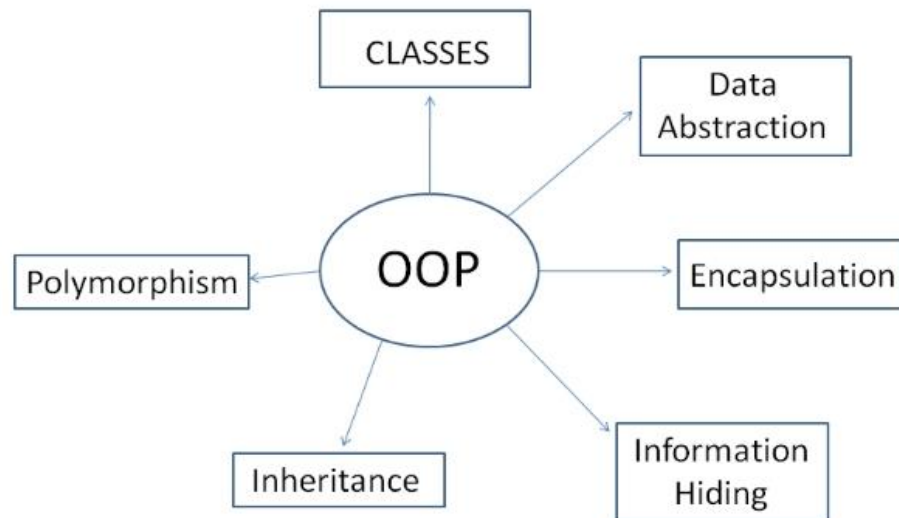


## ណែនាំអោយស្គាល់ OOP

### Object Oriented Programming in Java

#### I. ដូចម្តេចទៅដែលហៅថា OOP?

OOP ឈរលើពាក្យថា Object Oriented Programming សំដៅលើការបង្កើត Software ដោយពឹងលើ ការប្រើប្រាស់នូវ Class និង Object ។ OOP ត្រូវបានគេប្រើប្រាស់យ៉ាង ពេញនិយមលើការបង្កើតនូវ Software ឬ project ជាមួយនិង ក្រុមហ៊ុនធំៗ។ សព្វថ្ងៃក៏មានភាសាជាច្រើនដែលកំពុងតែប្រើប្រាស់នូវ Concept OOP ដូចជា៖ Java, C++, C#, Python, PHP, Ruby, Perl, Object Pascal, Objective-C, Dart, Swift, Scala, Common Lisp និង Smalltalk ។



Simula (ឆ្នាំ 1967) ត្រូវបានគេទទួលយកជាទូទៅថាជាភាសាទីមួយដែលមានលក្ខណៈ ចម្បងនៃ OOP (Object Oriented Programming) ។ វាត្រូវបានបង្កើតឡើងសម្រាប់បង្កើត Simulation ដែលអ្វីដែលត្រូវបានគេហៅថា Object គឺជាតំណាងការបង្ហាញព័ត៌មាន ។





OOP មាន Concept ៣ យ៉ាងសំខាន់ដូចជា៖

១) Abstraction: គឺសំដៅលើការរៀបចំ Data និង Method តាមលក្ខណៈមួយតាមលំដាប់លំដោយ ដែលត្រូវបិទបាំង ឬ ត្រូវបង្ហាញពីក្នុង Class។

Encapsulation: គឺសំដៅលើការប្រមូលផ្តុំ Data និង Methods ដោយក្នុងការគ្រប់គ្រងមួយត្រឹមត្រូវ។

២). Inheritance: សំដៅលើការបង្កើតនូវ Class ថ្មីមួយចេញពី Class ដែលមានស្រាប់ ឬ ការបង្កើត Class ដែលទទួលទិន្នន័យពីគ្នាបាន។

៣). Polymorphism: សំដៅលើការប្រើប្រាស់នូវទំរង់ច្រើនក្នុង Object តែមួយ។

II. ដូចម្តេចទៅដែលហៅថា Class?

Class សំដៅលើការប្រមូលផ្តុំនូវ Data & Method ដាក់ដោយការគ្រប់គ្រងដោយ Keyword Class និង ក្រោមឈ្មោះតែមួយរួមគ្នា។ ឬ Class គឺជាពុម្ពគំរូមួយដើម្បីបង្កើត object ។

Data member: គឺជាប្រភេទ Variable ឬ Array ដែលសំរាប់ផ្ទុក ទិន្នន័យជាបណ្តោះអាសន្ន។

ឧទាហរណ៍ ៖

```
1 //Data Member in class
2 import java.util.*;
3 class Student{
4     public int id;
5     public String name;
6     public String sex;
7     public float score;
8 }
```

Method member: គឺជាប្រភេទ Function ដែលប្រើសំរាប់ដោះស្រាយការងារអ្វីមួយ ។

Method អាចជា Return function និង Non Return function អាស្រ័យលើការងារជាក់ស្តែង។

ឧទាហរណ៍ ៖

```
1 //Data Member in class
2 import java.util.*;
3 class Student{
4     public int id;
5     public String name;
6     public String sex;
7     public float score;
8     //Method Member
9     public void inputData()
10    {
11        Scanner objin=new Scanner(System.in);
12        System.out.print("Input ID=");
13        this.id=objin.nextInt();
14        System.out.print("Input Name=");
15        this.name=objin.nextInt();
16        System.out.print("Input Gender=");
17        this.sex=objin.next();
18        System.out.print("Input Score=");
19        this.score=objin.nextInt();
20    }
21 }
```

### III. ដូចម្តេចទៅដែលហៅថា Object?

Object គឺជារបស់ដែលកើតចេញពី Class និង ដែលមានលទ្ធភាពអាចប្រើប្រាស់នូវទិន្នន័យរបស់ Class តែបានតែជា ប្រភេទ Public តែប៉ុណ្ណោះ។ ដើម្បីបង្កើតនូវ object ត្រូវប្រើប្រាស់នូវ keyword new ជាការស្រេច។

```
22 class TestStudents{
23     public static void main(String args[])
24     {
25         Student obj1=new Student();
26         //Or
27         Student obj2;
28         obj2=new Student();
29         obj1.inputData();
30
31         obj2.inputData();
32     }
33 }
```

## ឧទាហរណ៍ ១៖

```
2 import java.util.*;
3 class Student{
4     public int id;
5     public String name;
6     public String sex;
7     public float score;
8     //Method Member
9     public void inputData()
10    {
11        Scanner objin=new Scanner(System.in);
12        System.out.print("Input ID=");
13        this.id=objin.nextInt();
14        System.out.print("Input Name=");
15        this.name=objin.next();
16        System.out.print("Input Gender=");
17        this.sex=objin.next();
18        System.out.print("Input Score=");
19        this.score=objin.nextFloat();
20    }
21    public void outputData()
22    {
23        System.out.print("\nID="+id);
24        System.out.print("\nName="+name);
25        System.out.print("\nGender="+sex);
26        System.out.print("\nScore="+score);
27    }
28 }
29 class TestStudents{
30     public static void main(String args[])
31     {
32         //Create Object Class students
33         Student obj1=new Student();
34         obj1.inputData();
35         obj1.outputData();
36     }
37 }
38 }
```

## ឧទាហរណ៍ ២៖

```
1 import java.util.*;
2 public class Test{
3     private int x;
4     private int y;
5     private int z;
6     public void input()
7     {
8         Scanner objin=new Scanner(System.in);
9         System.out.print("Input Value X=");
10        x=objin.nextInt();
11        System.out.print("Input Value Y=");
12        y=objin.nextInt();
13        System.out.print("Input Value Z=");
14        z=objin.nextInt();
15    }
16    private int sum()
17    {
18        return x+y+z;
19    }
20    private int sub()
21    {
22        return x-y/z;
23    }
24    public void output()
25    {
26        System.out.print("X=" + x + " Y=" + y + " Z=" + z + "\n");
27        System.out.print("Sum=" + sum());
28        System.out.print("Sub=" + sub());
29    }
30 }
31 public class DemoTest{
32     public static void main(String args[])
33     {
34         Test obj=new Test();
35         obj.input();
36         obj.output();
37     }
38 }
```

## IV. Constructor

Constructor គឺជាប្រភេទ function ពិសេសមួយប្រភេទដែលធ្វើការដោយស្វ័យប្រវត្តនៅពេល Object មួយបានកកើតឡើង។ Constructor មានពីរប្រភេទ ដូចជា៖

១) Default Constructor: គឺជាប្រភេទ Constructor ដែលគេអាចបោះតំលៃ Default

អោយទៅ Data member សំរាប់ Object កកើតឡើង។

២) Constructor With Parameter: គឺជាប្រភេទ Constructor ដែលអាចបោះតំលៃពី ខាងក្រៅ តាមរយៈ Object របស់ Class។

```
1 import java.util.*;
2 public class Test{
3     private int x;
4     private int y;
5     private int c;
6     //Default Constructor
7     public Test()
8     { this.x=0;
9       this.y=0;
10      this.z=0;
11    }
12    //Constructor with Paramemter
13    public Test(int a,int b,int c)
14    {
15        this.x=a;
16        this.y=b;
17        this.z=c;
18    }
19    private int sum()
20    {
21        return x+y+z;
22    }
23    private int sub()
24    {
25        return x-y/z;
26    }
27    public void output()
28    {
29        System.out.print("X=" + x + " Y=" + y + " Z=" + z + "\n");
30        System.out.print("Sum=" + sum());
31        System.out.print("Sub=" + sub());
32    }
33 };
34 public class DemoTest{
35     public static void main(String args[])
36     { //Using Default Constructor
37         Test obj;
38         obj=new Test();
39         obj.output();
40         //Constructor with parameter
41         obj=new Test(12,30,3);
42         obj.output();
43     }
44 };
```

### លំហាត់អនុវត្តន៍

1). ចូរបង្កើត Class មួយដែលមានឈ្មោះ Employee ដែលផ្ទុក Data member ៣ គឺ id(String), Name(String), Sex(String) & Salary(double) និង មាន function ចំនួន ២ គឺ void Input() និង void Output() ដោយប្រើប្រាស់នូវ Console Application បន្ទាប់មកបង្កើតនូវ Object ចំនួនពីរ គឺ objemp1 and objemp2 ដែលប្រើប្រាស់នូវ Class នោះបាន?

## V. Object Array

```
public class DemoTest{
    public static void main(String args[])
    {
        Students [] stu=new Students[20];
        for(i=0;i<20;i++)
        {
            stu[i]=new Students();
        }
    }
};
```

ឧទាហរណ៍១ ៖ ចូរធ្វើការបង្កើតនូវClass មួយឈ្មោះ Books បន្ទាប់មកបង្កើតនូវ Data

member Code(int), title(String), subject(String) និង price(float) និង Constructor ពីរគឺ Books(), Books(\_\_\_\_\_) និង function ពីរទៀតគឺ void input(), void output() និង property set\_(), get\_()?

```
import java.util.*;
public class Books {
    private int code;
    private String title;
    private String subject;
    private float price;
    //Default Constructor
    public Books()
    {
        this.code=0;
        this.title="N/A";
        this.subject="N/A";
        this.price=0;
    }
    //Constructor with Parameter
    public Books(int i,String n, String s,float p)
    {
        this.code=i;
        this.title=n;
        this.subject=s;
        this.price=p;
    }
    //Property get
    public int getCode()
    {
        return code;
    }
    //property set
    public void setTitle(String n)
    {
        this.title=n;
    }
    public void setSubject(String s)
    {
        this.subject=s;
    }
    public void Price(float p)
    {
        this.price=p;
    }
    public void input ()
    {
        Scanner objin=new Scanner(System.in);
        System.out.println ("Input Code=");
        code=objin.nextInt();
        System.out.println ("Input Title=");
        title=objin.next();
        System.out.println ("Input Subject=");
        subject=objin.next();
        System.out.println ("Input price=");
        price=objin.nextFloat();
    }
    public void Output ()
    {
        System.out.println ("Code=" + code);
        System.out.println ("Title=" + title);
        System.out.println ("Subject=" + subject);
        System.out.println ("Price=" + price);
    }
}
```

ឧទាហរណ៍២ ៖ ចូរបង្កើតនូវ Object ជាលក្ខណៈ Array បន្ទាប់មកបង្កើត function ជំនួស ៤ គឺ Search(), Update() និង Sort()?

```
import java.util.*;
public class TestBook {
    public int Search(Books book[],int n)
    {
        Scanner objin=new Scanner(System.in);
        int scode,i;
        int index=-1;
        int b=0;
        System.out.println ("Input Book Code to search=");
        scode=objin.nextInt();
        for(i=0;i<n;i++)
        {
            if(scode==book[i].getCode())
            {
                book[i].Output();
                index=i;
                b=1;
                break;
            }
        }
        if(b==0) System.out.println ("Search not found");
        return index;
    }
    public void Update(Books book[],int n)
    {
        Scanner objin=new Scanner(System.in);
        int index=Search(book,n);
        String ntitle,nsup;
        float nprice;
        if(index!=-1)
        {
            System.out.println ("Input New Title=");
            ntitle=objin.next();
            System.out.println ("Input New Subject=");
            nsup=objin.next();
            System.out.println ("Input New price=");
            nprice=objin.nextFloat();
            book[index].setTitle(ntitle);
            book[index].setSubject(nsup);
            book[index].setPrice(nprice);
            System.out.println ("Update Completed.....!");
        }
        else{
            System.out.println ("Invalid Updated.....!");
        }
    }
}
```





```

public TestBook() {
    Scanner objin=new Scanner(System.in);
    int op,n=0;
    String st;
    Books []book=new Books[20];
    do{

        System.out.println ("1. Input");
        System.out.println ("2. Output");
        System.out.println ("3. Search");
        System.out.println ("4. Update");
        System.out.print ("Choose One=");
        op=objin.nextInt();
        switch(op)
        {
            case 1:{
                System.out.print ("Input Number Array=");
                n=objin.nextInt();
                for(int i=0;i<n;i++)
                {
                    book[i]=new Books();
                    book[i].input();
                }
            }break;
            case 2:{
                for(int i=0;i<n;i++)
                {
                    book[i].Output();
                }
            }break;
            case 3:{
                Search(book,n);
            }break;
            case 4:{
                Update(book,n);
            }break;
        }

        System.out.println ("Press yes to Continue....!");
        st=objin.next();
    }while(st.equals("yes"));
}

public static void main (String[] args) {
    new TestBook();
}
}

```

### លំហាត់អនុវត្តន៍

១). ចូរបង្កើត Class មួយដែលមានឈ្មោះ Employee ដែលផ្ទុក Data member ៣ គឺ id(String), Name(String), Sex(String) & Salary(double) និង មាន function ចំនួន ២ គឺ void Input() និង void Output() និង Constructor ចំនួនពីរ ទៀតគឺ Employee(), Employee(\_\_\_\_), Employee(\_\_\_\_,\_\_\_\_,\_\_\_\_) ដោយប្រើប្រាស់នូវ Console Application បន្ទាប់មកបង្កើតនូវ Object ចំនួនពីរ គឺ objemp1 and objemp2 ដែលប្រើប្រាស់នូវ Class នោះបានបន្ទាប់មកបង្កើតនូវ Object ArrayList?

## ណែនាំអោយស្គាល់ Inheritance

### Object Oriented Programming in Java

#### I. ដូចម្តេចទៅដែលហៅថា Inheritance ?

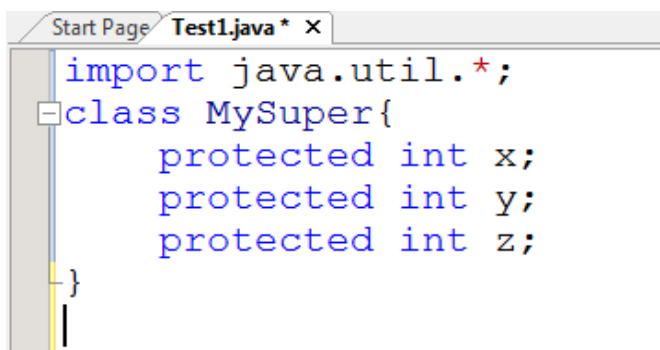
Inheritance គឺជាដំណើរនៃការកកើត Class ថ្មីមួយចេញពី Class ដែលមានស្រាប់ ឬ សំដៅលើការបង្កើតនូវ Relationship Class (ទំនាក់ទំនង Class) ។ Class ដែលមានស្រាប់គេហៅថា Base Class Or Super Class និង class ដែលមានស្រាប់គេហៅថា Derived Class ឬ Sub Class។

លក្ខណៈសំខាន់របស់ Inheritance គឺកាត់បន្ថយនូវការសរសេរកូដ ឬ ប្រកាសអញ្ញាតិ ដដែលៗ ច្រើនដង ឬកំណត់នូវការទំនាក់ទំនង នៃ Class អោយភាពច្បាស់លាស់។

ទំរង់ Inheritance ៖

```
class Subclass-name extends Superclass-name
{
    //methods and fields
}
```

ឧទាហរណ៍ 1៖ គេមាន Super Class មួយដែលផ្ទុកនូវ Protected Data member 3 គឺ x,y,z



```
Start Page Test1.java * x
import java.util.*;
class MySuper{
    protected int x;
    protected int y;
    protected int z;
}
```

ឧទាហរណ៍ ២៖ ចូរបង្កើតនូវ Sub Class ដែលអាចទាញទិន្នន័យពី Super Class ដែលមានលក្ខណៈ ជា Protected ឬ Public ព្រមទាំងបង្កើតនូវ object ដែលអាចប្រើប្រាស់នូវ Sub Class នោះបាន?

```
Start Page Test1.java * x
import java.util.*;
class MySuper{
    protected int x;
    protected int y;
    protected int z;
}
class MySub extends MySuper{
    private int a;
    private int sum()
    {
        return x+y+z+a;
    }
    private int sub()
    {
        return x-y+z;
    }
    public void Input()
    {
        Scanner objin=new Scanner(System.in);
        System.out.print("Input X=");
        x=objin.nextInt();
        System.out.print("Input Y=");
        y=objin.nextInt();
        System.out.print("Input Z=");
        z=objin.nextInt();
        System.out.print("Input A=");
        a=objin.nextInt();
    }
    public void Output()
    {
        System.out.print("X=" + x + "\n");
        System.out.print("Y=" + y + "\n");
        System.out.print("Z=" + z + "\n");
        System.out.print("A=" + a + "\n");
        System.out.print("Sum=" + sum() + "\n");
        System.out.print("Sub=" + sub() + "\n");
    }
}
class Test1{
    public static void main (String[] args) {
        MySub obj=new MySub();
        obj.Input();
        obj.Output();
    }
}
```

## លំហាត់អនុវត្តន៍ ១៖

ចូរបង្កើតនូវ Super Class មួយឈ្មោះ Person ដែលមាននូវ Data member ៤ គឺ៖ Id(String), Name(String), Sex(String), Address(String) និង DOB(String) បន្ទាប់មកបង្កើតនូវ Function ចំនួនពីរ គឺ void Input() និង void Output()។

ចូរបង្កើតនូវ Class Employee ដែលមាន data member 3 គឺ Salary(float), Rate(float) និង hour(int) បន្ទាប់មកបង្កើតនូវ Function ចំនួនពីរគឺ void Input(), void output() ដែល Function ទាំងពីរនេះ ត្រូវទាញទិន្នន័យចេញពី Class person មកប្រើប្រាស់បន្ថែម?

II. ដើម្បីហៅ Function ពី Super Class ត្រូវប្រើប្រាស់នូវ keyword super.Input( ) ឬ super.Output( ) .

```
class MySub extends MySuper{
    private int a;
    public void Input()
    {
        Scanner objin=new Scanner(System.in);
        //Calling Input from Super Class
        super.Input();
        System.out.print("Input A=");
        a=objin.nextInt();
    }
    public void Output()
    {
        //Calling Output from Super Class
        super.Output();
        System.out.print("A=" + a + "\n");
        System.out.print("Sum=" + sum() + "\n");
        System.out.print("Sub=" + sub() + "\n");
    }
}
```

### III. ដើម្បី ហៅ Constructor ពី Super Class

```
Start Page Test1.java * x
import java.util.*;
class MySuper{
    protected int x;
    protected int y;
    protected int z;
    //Default Constructor
    public MySuper()
    {
        x=0;
        y=1;
        z=2;
    }
    //Constructor with Parameter
    public MySuper(int x,int y,int z)
    {
        this.x=x;
        this.y=y;
        this.z=z;
    }
}
class MySub extends MySuper{
    private int a;
    public MySub()
    { //calling Default Constructor
        super();
        a=4;
    }
    public MySub(int a,int b,int c,int d)
    { //calling Constructor with parameter
        super(a,b,c);
        this.a=d;
    }
}
```

#### លំហាត់អនុវត្តន៍ ២៖

តពីលំហាត់អនុវត្តន៍ទី ១ ចូរបន្ថែមនូវ Default Constructor និង Constructor With Parameter ក្នុង Super Class បន្ទាប់មកបង្កើតនូវ Sub class ហៅ Constructor ទាំងនោះចូលទៅប្រើប្រាស់បន្ថែម ?

#### IV. ប្រភេទនៃ Inheritance

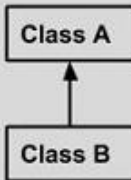
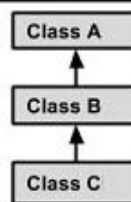
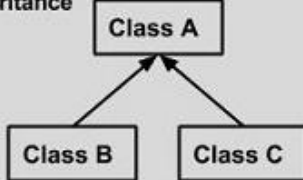
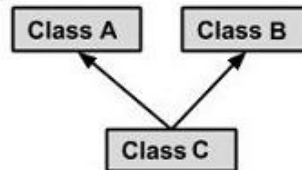
នៅក្នុង Inheritance ត្រូវបានគេបែងចែកជា ៤ ទំរង់ដូចជា៖

១) Single Inheritance: គឺជាប្រភេទ Inheritance ដែលមាន Super Class មួយអាច មាន នូវ Sub Class ១ ទាញយកទិន្នន័យបន្តគ្នា។

២) Multiple Level inheritance : គឺជាប្រភេទ inheritance ដែលអាចមាន Super class មួយ និង Sub class 2 ឡើងទៅដែលត្រូវទាញទិន្នន័យបន្ត Class គ្នា។

៣) Hierarchical Inheritance: គឺជាប្រភេទ Inheritance ដែលមាន Super Class មួយ ហើយមាន Sub Class ច្រើនទាញទិន្នន័យដោយផ្ទាល់ពី Super Class ទាំងនោះ។

៤) Multiple Inheritance: គឺជាប្រភេទ Inheritance ដែលមាននូវ Super Class ពីរ ឬ ច្រើនឡើងទៅ។

<b>Single Inheritance</b> 	<pre>public class A {     ..... } public class B extends A {     ..... }</pre>
<b>Multi Level Inheritance</b> 	<pre>public class A { .....} public class B extends A {.....} public class C extends B {.....}</pre>
<b>Hierarchical Inheritance</b> 	<pre>public class A { .....} public class B extends A {.....} public class C extends A {.....}</pre>
<b>Multiple Inheritance</b> 	<pre>public class A { .....} public class B {.....} public class C extends A,B {     ..... } // Java does not support mutiple Inheritance</pre>

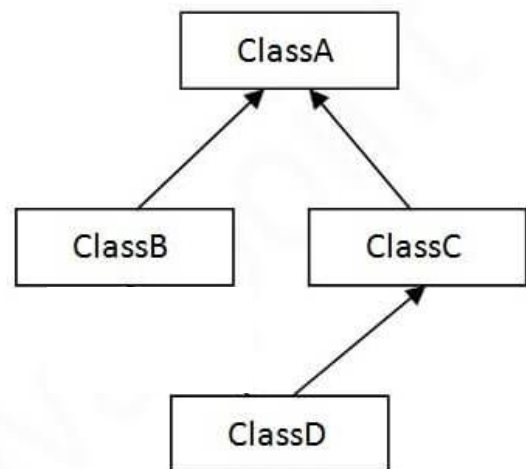
### លំហាត់អនុវត្តន៍ ៣៖

ចូរបង្កើតនូវ Super Class មួយឈ្មោះ Person ដែលមាននូវ Data member ៤ គឺ៖ Id(String), Name(String), Sex(String), Address(String) និង DOB(String) និង បង្កើតនូវ Constructor ចំនួនពីរទៀតគឺ Person(), Person(\_\_\_\_\_,\_\_\_\_\_,\_\_\_\_\_,\_\_\_\_\_) បន្ទាប់មកបង្កើតនូវ Function ចំនួនពីរ គឺ void Input() និង void Output()។

ចូរបង្កើតនូវ Sub Class មួយគឺ Students ដែលមាន score 5 មុខវិជ្ជា, Constructor ចំនួនពីរ និង Function Member ចំនួនពីរ ថែមទៀត?

ចូរបង្កើតនូវ Sub Class មួយ ទៀតគឺ Employee ដែលមាន Data member ជា Salary(float) និង Constructor ចំនួនពីរ និង Function Member ចំនួនពីរ ថែមទៀត?

ចុងក្រោយចូលអោយ Sub Class ទាំងពីរទាញយកទិន្នន័យពី Super Class មួយនោះ?  
និងចូលបង្កើត Class Teacher ដែលមាន Data member ពីរ គឺ Subject និង RoomNo និង Constructor ចំនួនពីរ និង Function Member ចំនួនពីរ ថែមទៀត  
ទៅទាញយកទិន្នន័យពី Class Students



5) Hybrid