

Property of Array:

All arrays inherit implicitly from Array class

Length: return number of element in array

ex: `int []x = new int[7];`

`int y = x.Length;`                      `//y = 7`

`int [,]a = new int[3, 4];`

`int b = a.Length;`                      `//b = 12`

`int row = a.GetLength(0);`                      `//3`

`int col = a.GetLength(1);`                      `//4`


c# allows you to iterate over **all items** within a collection (ex: array, ...)

using *foreach* statement

Ex:

class Test

{

    public static void Main()

    {

        int Sum = 0;

        int []ar = new int[5]{1, 2, 3, 4, 5};

        int Max, Min;

        bool isFirst = true;

        foreach(int i in ar)

        {

            if(isFirst)

            {           Max = Min = i;

                      isFirst = false;    }

            else

            {           if(i > Max)        {Max = i;}

                      if(i < Min)        {Min = i;}

            }

            Console.WriteLine(i);

            Sum = Sum + i;

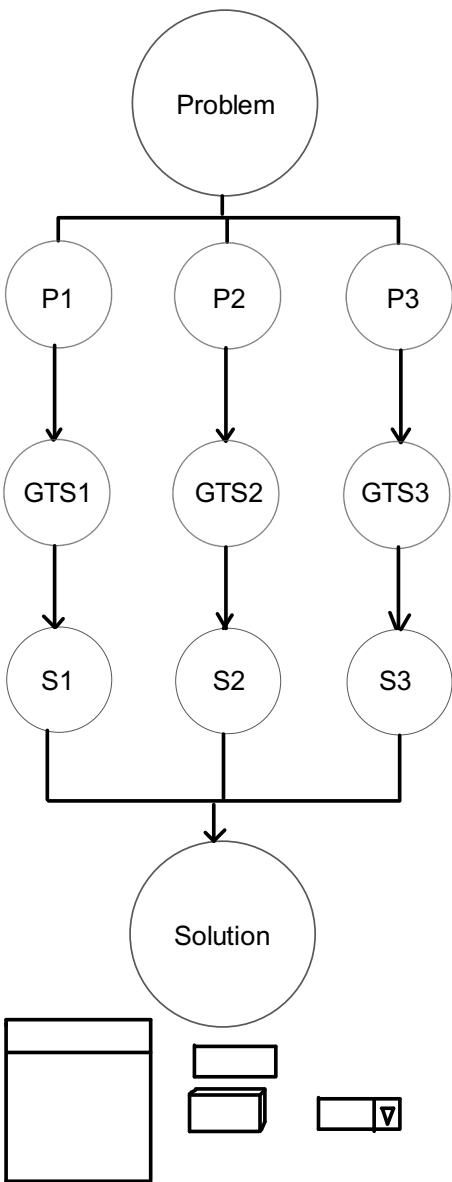
            i++;                      //Compile Time Error

        }

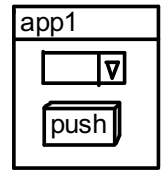
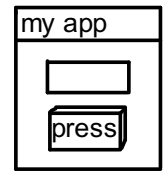
    }

}

You Can't Modify the element in a collection using foreach, because the iteration variable is readonly



# Object Oriented Programming

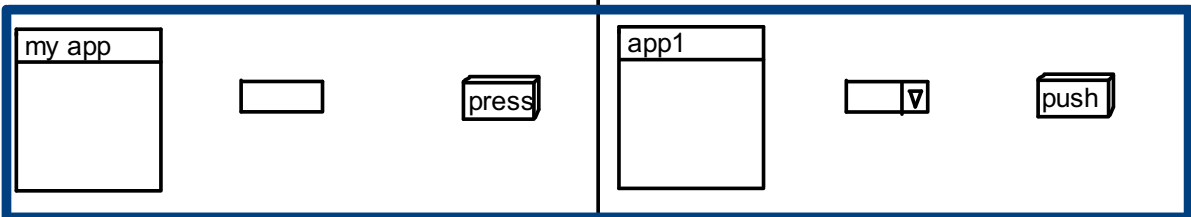


Object:  
property  
behavior

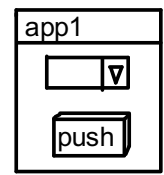
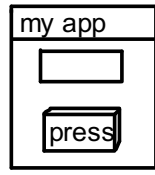
Re-Usability

Form	textbox	Button	Form	DDList	Button
width height caption ... OnOpen() OnClose() ...	width noline text ... OnSelect() ... ...	size text ... OnClick() ... ...	✓	width noitem visibility ... OnChange() ... ...	✓

*class*



*object*



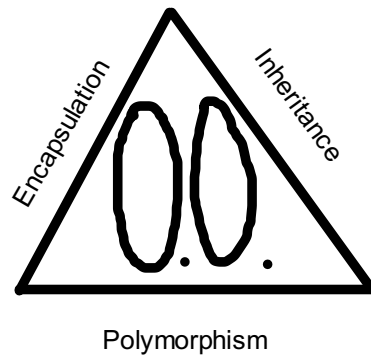
Object Oriented is based on:

Encapsulation:

Inheritance:

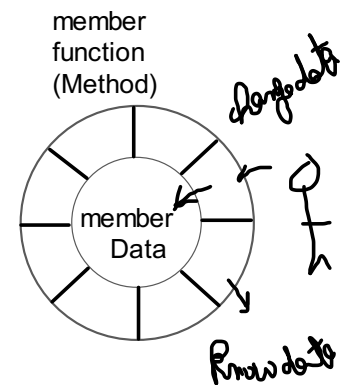
Polymorphism:

Overloading  
Overriding



Object:

Property  
Behavior



```

class MyClass
{
    private int x;           //member data
    public int GetX()        //Method
    {
        return x;
    }
    public void SetX(int a)   //Method
    {
        this.x = a;
    }
}

```

Creating Object: (classes are reference type)

- 1) Declaration MyClass obj;
- 2) Creation obj = new MyClass();

```

class Test
{
    public static void Main()
    {
        MyClass obj;
        MyClass obj1;
        obj = new MyClass();
        obj1 = new MyClass();
        obj.SetX(20);
        obj1.SetX(70);
    }
}

```

Access Modifier:

private: no one can access private member except member of the same class

public: anyone can access public member

