Indexers

- An array elements can be accessed using [] operator.
- C# also provides a capability to use [] operators on any user-defined classes (or struct)
- For instance,
  - class FlowerVase{….}
  - FlowerVase f= new FlowerVase()
  - f[0]= 12
  - f[1]= 13
- But for this we need to create something called a indexer.
- the object will be act as an array when the class is having indexers
- Declaration:
- public return-type this[int index-position]
- {
- get { return some-value}
- set { some-value= value}
- }
- indexer name should be this keyword

Indexers

- Indexers in C# must have at least one parameter. Else the compiler will generate a compilation error.
- One parameter indexer are called one dimensional indexers
- Two parameter indexers are called two dimensional indexers
- Indexers can be overloading
- Indexers can't declared as static[static indexers are not supported by c#]
- indexers directly access with the class object

```
//to call setaccessor of indexers
object[index]=value;
//to call get accessor
some variable=object[index];
```

Indexers

- two dimensional indexers

```
public return-type this[int index, int index1]
{
get { return some-value}
set { some-value= value}
}
```