

Operator Overloading:-

```
class Complex {
```

```
    int real;
```

```
    int imag;
```

```
    public Complex (int r, int i)
```

```
    {
```

```
        this.real = r;
```

```
        this.imag = i;
```

```
    }
```

write default
constructor as well.
to avoid error
in complex

// Getter setter Concept in C#

```
public int Real {
```

```
    get { return this.real; }
```

```
    set { this.real = value; }
```

```
}
```

Property Syntax
taken from
visual
basic

```
public int Imag {
```

```
    get { return this.imag; }
```

```
    set { this.imag = value; }
```

```
}
```

// operator overloading

```
public static Complex operator +
```

```
    (Complex c1, Complex c2) {
```

Return type

operators
to add

be overloaded
Should
be
static

```
Complex temp = new Complex();  
temp.Real = C1.Real + C2.Real;  
temp.Imag = C1.Imag + C2.Imag;  
return temp;  
}
```

// operator overloading is required in scientific calculations.

// All arithmetic operators can be overloaded.

// in main

```
Complex C1 = new Complex(34, 56);  
Complex C2 = new Complex(11, 78);  
Complex C3 = C1 + C2;
```

