

# FUNCIONES AMIGAS



Por: Luis Humberto González Guerra  
Ma. Guadalupe Roque Díaz de León

# Funciones Libres

---

Son aquellas funciones que se generan para un fin específico y no pertenece a ninguna clase en particular.

*Ejemplo: función que encuentre el mayor de dos números enteros.*

```
int mayor(int a, int b){  
    if (a > b)  
        return a;  
    else  
        return b;  
}
```

```
int a, b;  
cin >> a >> b;  
int may = mayor(a,b);
```

# Función Miembro

---

- Es aquella función que pertenece a una clase en particular.

*Ejemplo: una función llamada **leeArchivo** que pertenece a la Clase **Series** y lee todas las series desde un archivo.*

```
void Series::leeArchivo(){  
:  
:  
}
```

```
Series ser;  
  
ser. leeArchivo();
```

# Funciones libres vs. Funciones miembro

---

- Los número Complejos son aquellos que tienen una parte real y una parte imaginaria.

```
private:  
    int iReal, ilmag;
```

# Funciones libres vs. Funciones miembro

---

- Los suma de dos números complejos

$$(8 + 5i) + (5 - 2i) = (13 + 3i)$$

- Los resta de dos números complejos da como resultado un número complejo.

$$(8 + 5i) - (5 - 2i) = (3 + 7i)$$

# Clase Complejo

---

```
class Complejo{  
    private:  
        int iReal, imag;  
    public:  
        Complejo();  
        Complejo(int , int );  
        int getReal();  
        void setReal(int);  
        int getImag();  
        void setImag(int i);  
        Complejo suma(Complejo c2);  
}
```

# Clase Complejo

```
Complejo Complejo::suma(Complejo c2){  
    int realN = real + c2.real;  
    int imagN = imag + c2.imag;  
    Complejo nuevo(realN, imagN);  
    return nuevo;  
}
```

La llamada sería:

```
Complejo a(8,5), b(5,2), c;  
c = a.suma(b);
```

```
Complejo resta(Complejo c1, Complejo c2){  
    int realN = c1.getReal() - c2.getReal();  
    int imagN = c1.getImag() - c2.getImag();  
    Complejo nuevo(realN, imagN);  
    return nuevo;  
}
```

La llamada sería:

```
Complejo a(8,5), b(5,2), c;  
c = resta(a,b);
```

# Funciones Amigas

---

- Es aquella función libre a la cual se le permite acceder 😬 a atributos privados.

Para que una función libre pueda ser **función amiga**, se debe poner el encabezado de la función dentro del prototipo de la clase, anteponiendo la palabra **friend**.

```
friend Complejo resta(Complejo c1, Complejo c2)
```



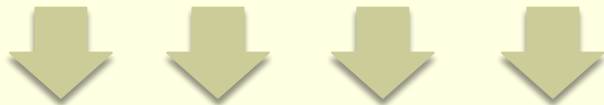
# Clase Complejo

---

```
class Complejo{
    private:
        int real, imag;
    public:
        Complejo( );
        Complejo(int, int );
        int getReal( );
        void setReal(int );
        int getImag( );
        void setImag(int );
        Complejo suma(Complejo );
        // función Amiga 🥳
        friend Complejo resta(Complejo , Complejo)
}
```

# La clase amiga en la Clase Complejo

```
Complejo resta(Complejo c1, Complejo c2){  
    int iRealN = c1.getReal()-c2.getReal();  
    int ilmagN = c1.getImag()-c2.getImag();  
    Complejo nuevo(iRealN, ilmagN);  
    return nuevo;  
}
```



```
Complejo resta(Complejo c1, Complejo c2){  
    int iRealN = c1.iReal - c2.iReal;  
    int ilmagN = c1.ilmag - c2.ilmag;  
    Complejo nuevo(iRealN, ilmagN);  
    return nuevo;  
}
```

La llamada sería:

Complejo a(8,5), b(5,2), c;  
c = resta(a,b);

# Actividad-

---

Añade a la clase Fracción una función amiga llamada suma - que sume dos fracciones y retorne la suma de las dos fracciones -  
Añade la llamada desde el main-

# Actividad-

## Solución

```
Fracciones suma(Fracciones f1, Fracciones f2){  
    int iNum, iDen;  
    if (f1.iDen == f2.iDen)  
    {  
        iNum = f1.iNum + f2.iNum;  
        iDen = f1.iDen ;  
    }  
    else  
    {  
        iNum = (f1.iNum * f1.iDen + f1.iDen * f2.iNum);  
        iDen = (f1.iDen * f2.iDen);  
    }  
  
    Fracciones nuevo(iNum, iDen);  
    return nuevo;  
}
```