

# Tema

- Clases y objetos



- Clases y objetos

## Clases y objetos

### Cómo se define una clase en C++

- Una clase puede contener partes públicas y privadas. Por default, los elementos que son definidos dentro de una clase son privados, esto quiere decir que no pueden ser accesados desde afuera de la clase, también se puede especificar cuales elementos serán privados con la palabra `private`.
- Para hacer algunos elementos de la clase públicos se utiliza la palabra `public`, los elementos que se definen después de esta palabra son públicos, lo que significa que pueden ser utilizados desde afuera de la clase.

## Clases y objetos

- Solo los métodos de la clase pueden accesar los atributos privados de la misma.

**El formato para declarar una clase es el siguiente:**

```
class NombreClase
{
    public:
        // declaración de atributos y métodos públicos
    private:
        // declaración de atributos y métodos privados
};
```

## Clases y objetos

Ejemplo 1 Supongamos que requerimos declarar la clase círculo para un programa en el que se requiere calcular el área del círculo, podemos declarar la clase como se muestra en el siguiente ejemplo:

```
class Circulo
{
    public:
        double CalcArea( );
        void setRadio(double radio);
        double getRadio();
    private:
        double radio;
};
```

## Clases y objetos

- Nota que en esta declaración las funciones no se declaran completas, solamente se declara su prototipo.
- Las funciones se definen después, utilizando un formato en el que se tiene que especificar de que clase es miembro este método, como se muestra en el siguiente formato:

tipo NombreClase :: NombreMétodo ( lista de parámetros)

```
{  
    estatutos;  
}
```

# Clases y objetos



## Clases y objetos

continuando con el ejemplo:

```
double Circulo::CalcArea( )
```

```
{  
    return 3.1416*radio*radio;  
}
```

```
void Circulo::setRadio( double ra)
```

```
{  
    radio = ra;  
}
```

```
double Circulo::getRadio()
```

```
{  
    return radio;  
}
```

## Clases y objetos

- Dentro de un programa de C++ una instancia u objeto de la clase se representa como una variable, por lo que para crear una instancia declara una variable cuyo tipo de dato es el nombre del objeto, se usa el siguiente formato:

NombreClase NombreVariable;

para nuestro ejemplo:

Circulo A;

- en este caso estamos declarando la variable A que es un objeto de tipo Circulo; nota que el nombre de la clase es el tipo de dato.

## Clases y objetos

- Un mensaje pide a un objeto que ejecute alguno de sus métodos. Para enviar un mensaje a un objeto se usa el siguiente formato:

NombreObjeto.NombreMétodo (parámetros);

para nuestro ejemplo:

A.setRadio(10);

en este caso estamos llamando al método setRadio de la clase Circulo.

## Clases y objetos

- Ejemplo 2 El siguiente ejemplo muestra como se puede crear y usar un objeto dentro de una aplicación:

```
int main( )
{
    Circulo A;    // se crea el objeto A de la clase Circulo
    double area;
    A.setRadio(10); // se envía el mensaje al objeto A para que
                    // ejecute el método setRadio
    area = A.CalcArea( ); // se envía el mensaje al objeto A para que
                        // ejecute el método CalcArea
    return 0;
}
```

## Clases y objetos

- Es común colocar la declaración y definición de una clase en un archivo y poner a ese archivo el nombre de la clase con extensión .h. Para nuestro ejemplo podemos utilizar:

Circulo.h

Y para usar la clase Circulo en un programa cliente, se incluye la clase con un #include. Por ejemplo:

```
#include "Circulo.h"
```

## Clases y objetos

**El ejemplo anterior completo queda como sigue:**

// Esto se encuentra en el archivo Circulo.h

**class Circulo**

{ **public:**

**double CalcArea( );**

**void setRadio(double ra);**

**double getRadio();**

**private:**

**double radio;**

**}**

**double Circulo::CalcArea( )**

**{**

**return 3.1416\*radio\*radio;**

**}**

**void Circulo::setRadio( double ra)**

**{**

**radio= ra;**

**} // continua**

# Clases y objetos



## Clases y objetos

```
double Circulo::getRadio()
{
    return radio;
}

// Esto se encuentra en la aplicación UsaCirculo.cpp
#include <iostream.h>
#include "Circulo.h"

int main( )
{
    Circulo A;
    double area;

    A.setRadio(10);
    area = A.CalcArea( );
    cout<<area;

    return 0;
}
```