

# Temas

- Arreglos con objetos
- Ejemplo



- Arreglos con objetos

## Arreglos con objetos

### Arreglos con objetos

- Un arreglo de objetos de clase es útil cuando se requieren instancias múltiples de la misma clase. Así por ejemplo, si se define un arreglo de objetos Punto llamado figura, el constructor por omisión Punto llama para cada miembro del arreglo.  
Punto figura[3];
- Hay dos formas en las que se pueden relacionar los arreglos con los objetos:
  - Un arreglo puede contener elementos de tipo objetos.
  - Una clase puede contener como atributos variables de tipo arreglo.

# Arreglos con objetos

## Arreglos con objetos

- En esta sección veremos el caso en el que tenemos un arreglo con elementos de tipo Objeto.

Supón que tenemos la siguiente declaración:

```
class Articulo
{
    public:
        Articulo();
        void inicializa(int, int, double);
        double getPrecio();
        int getClave();
        int getExistencia();
        void registraEntrada(int cant);
        int registraSalida(int cant);
    private:
        int clave;
        int existencia;
        double precio;
};
```

## Arreglos con objetos



## Arreglos con objetos

Articulo Lista[20];

Para asignar valor al primer artículo del arreglo, usamos:

```
Lista[0].inicializa(12398, 200, 47.50);
```

- Nota que aquí se combina la sintaxis del uso de arreglos y del uso de objetos. Es importante que notes que lo que viene después del punto debe ser algún miembro público del objeto; es decir, no puedes accesar directamente los datos del objeto que son privados.

# Arreglos con objetos



## Ejemplo

### Ejemplo de programa

```
// Esta clase se encuentra en el archivo Articulo.h
// Clase articulo.
// Cada articulo esta definido por su clave, precio y la cantidad de articulos en existencia.
#include <iostream.h>
```

```
class Articulo
{
public:
    Articulo();
    void inicializa(int, int, double);
    double getPrecio();
    int getClave();
    int getExistencia();
    void registraEntrada(int cant);
    int registraSalida(int cant);
private:
    int clave;
    int existencia;
    double precio;
```

# Arreglos con objetos

## Ejemplo

```
Articulo::Articulo()
{
    clave = 0;
    existencia = 0;
    precio = 0;
}

// pone al objeto los valores recibidos como parámetro
void Articulo::inicializa(int cve, int exis, double pre)
{
    clave = cve;
    existencia = exis;
    precio = pre;
}

// Regresa el precio
double Articulo::getPrecio()
{
    return precio;
```

# Arreglos con objetos



## Ejemplo

```
// Regresa la clave
int Articulo::getClave()
{
    return clave;
}

// Regresa la cantidad en existencia
int Articulo::getExistencia()
{
    return existencia;
}

// Agrega cant a la cantidad de articulos en existencia
void Articulo::registraEntrada(int cant)
{
    existencia += cant;
}

// Si no hay suficientes articulos regresa 0, si hay suficientes
```

# Arreglos con objetos



## Ejemplo

```
int Articulo::registraSalida(int cant)
{
    if (cant > existencia)
        return 0; // false indica que no hubo suficientes artículos
    else
    {
        existencia -= cant;
        return 1; // true indica que si registro la salida
    }
}
```

# Arreglos con objetos

## Ejemplo

```
// Este programa se encuentra en el archivo UsaArt.cpp
#include <iostream.h>
#include "Articulo.h"

Articulo lista[30];
int tam;

void verInventario()
{
    int i;

    cout<<"La lista de artículos en el almacén es la siguiente"<<endl;
    cout<<"\t" <<"Clave" <<"\t" <<"Cantidad en Existencia" <<"\t" <<"Precio" <<endl;
    for (i = 0; i < tam; i++)
        cout<<"\t" <<lista[i].getClave() << "
            <<"\t \t" <<lista[i].getExistencia() << "
            <<"\t $" <<lista[i].getPrecio() << endl;
    cout<<endl << endl;
}
```

# Arreglos con objetos

## Ejemplo

```
int Busca(int cve)
{
    int i;
    int posicion = -1; // -1 indica que no se encontró el artículo
    for (i = 0; i<tam && posicion == -1; i++)
        if (lista[i].getClave() == cve)
            posicion = i; // posición en la que encuentra el artículo
    return posicion;
}

void registraLlegada()
{
    int posicion, cve, cantidad;

    cout<<endl;
    cout<<"cuál es la clave del artículo"<<endl;
    cin>>cve;
//CONTINUA
```

# Arreglos con objetos

## Ejemplo

```
//busca la clave en el almacen
if ((posicion = Busca(cve)) != -1)
{
    cout<<"cuantos artículos llegaron"<<endl;
    cin>>cantidad;
    lista[posicion].registraEntrada(cantidad);
}
else
    cout<<"Esa clave no existe "<<endl;
cout<<endl;
}    //CONTINUA
```

# Arreglos con objetos

## Ejemplo

```
void registraSalida()
{
    int posicion, cve, cantidad;

    cout<<endl;
    cout<<"cuál es la clave del artículo"<<endl;
    cin>>cve;
    //busca la clave en el almacen
    if ((posicion = Busca(cve)) != -1)
    {
        cout<<"cuantos artículos salieron"<<endl;
        cin>>cantidad;
        if (!lista[posicion].registraSalida(cantidad))
            cout<<"No hay suficientes artículos "<<endl;
    }
    else
        cout<<"Esa clave no existe "<<endl;
    cout<<endl;
}
```

# Arreglos con objetos

## Ejemplo

```
char menu()
{
    char opc;
    cout<<endl<<endl;
    cout<<" Menu de Opciones"<<endl;
    cout<<"a. Ver el inventario"<<endl;
    cout<<"b. Registrar una llegada al almacén"<<endl;
    cout<<"c. Registrar una salida de almacén"<<endl;
    cout<<"d. Salir del programa"<<endl;
    do
    {
        cout<<"Opcion ->";
        cin>>opc;
    }
// CONTINUA
```

# Arreglos con objetos



## Ejemplo

---

```
while ((opc < 'a') || (opc > 'd'));  
    return opc;  
}
```

```
int main()  
{  
    int i, cve, exis;  
    double pre;  
    char opc;
```

```
// CONTINUA
```

# Arreglos con objetos

## Ejemplo

```
cout<<"Cuantos artículos ";
cin>>tam;
for (i = 0; i < tam; i++)
{
    cout<<"Teclea la clave para el artículo "<<i+1<<endl;
    cin>>cve;
    cout<<"Teclea la existencia para el artículo "<<i+1<<endl;
    cin>>exis;
    cout<<"Teclea el precio para el artículo "<<i+1<<endl;
    cin>>pre;
    lista[i].inicializa(cve, exis, pre);
}
do
{
// CONTINUA
```

# Arreglos con objetos



## Ejemplo

```
opc = menu();
switch (opc)
{
    case 'a': verInventario();
                break;
    case 'b': registraLlegada();
                break;
    case 'c': registraSalida();
                break;
}
} while (opc != 'd');
return 0;
}
```