

PROGRAMACIÓN ORIENTADA A OBJETOS

Estimados aprendices, ANTES de iniciar con el presente taller, es necesario leer respecto de los tipos de datos basicos presentes en C++. Realice una busqueda en la Red.

Tipo	Descripcion	Rango
int		
Long		
Float		
Double		
char		

En C++, para guardar por ejemplo un numero en la varibale llamada **a**, se declara int a;

```
cin>>a;
```

```
cout<< a;
```

Para guardar por ejemplo un nombre en la variable llamada **x**, se usa el tipo de char persona[10],

NOTA: si el nombre por ejemplo es *Sandra*, se puede guardar usando

```
cin>>nombrepersona;
```

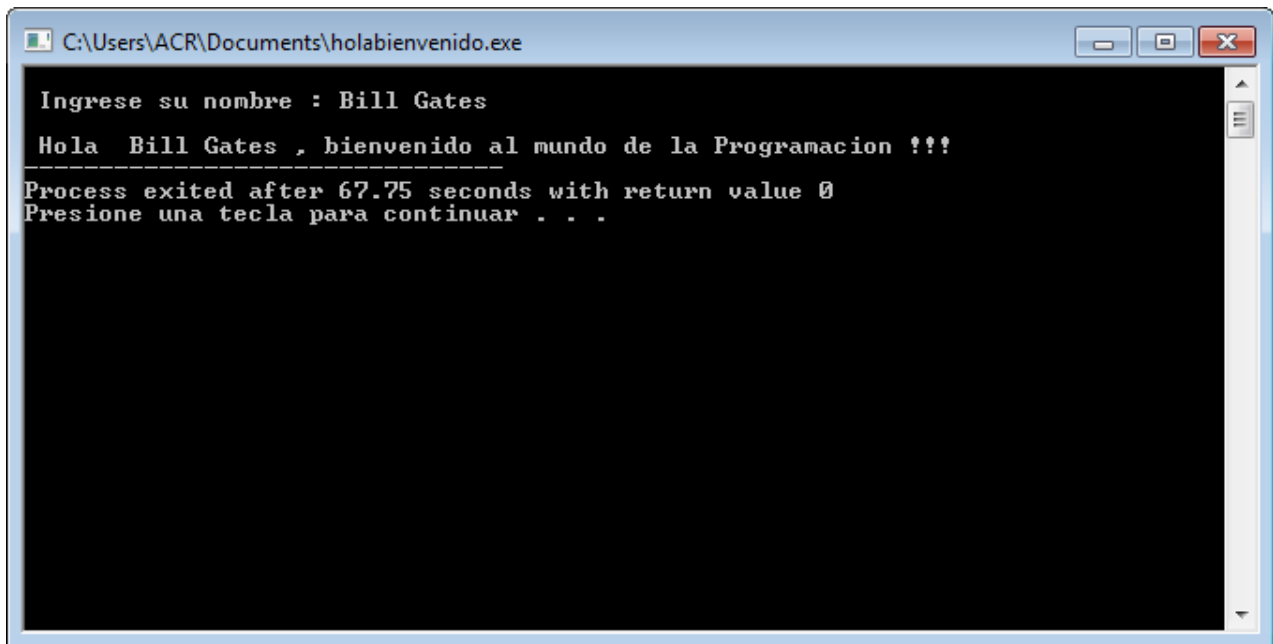
```
cout<< nombrepersona;
```

Pero si se desea guardar un nombre compuesto, por ejemplo *Sandra Milena*, la captura se realiza: `gets(x)`; si se usara `cin>>nombrepersona`; SOLO se guardaria *Sandra*

Ver codigo de ejemplo:

holabienvenido.cpp

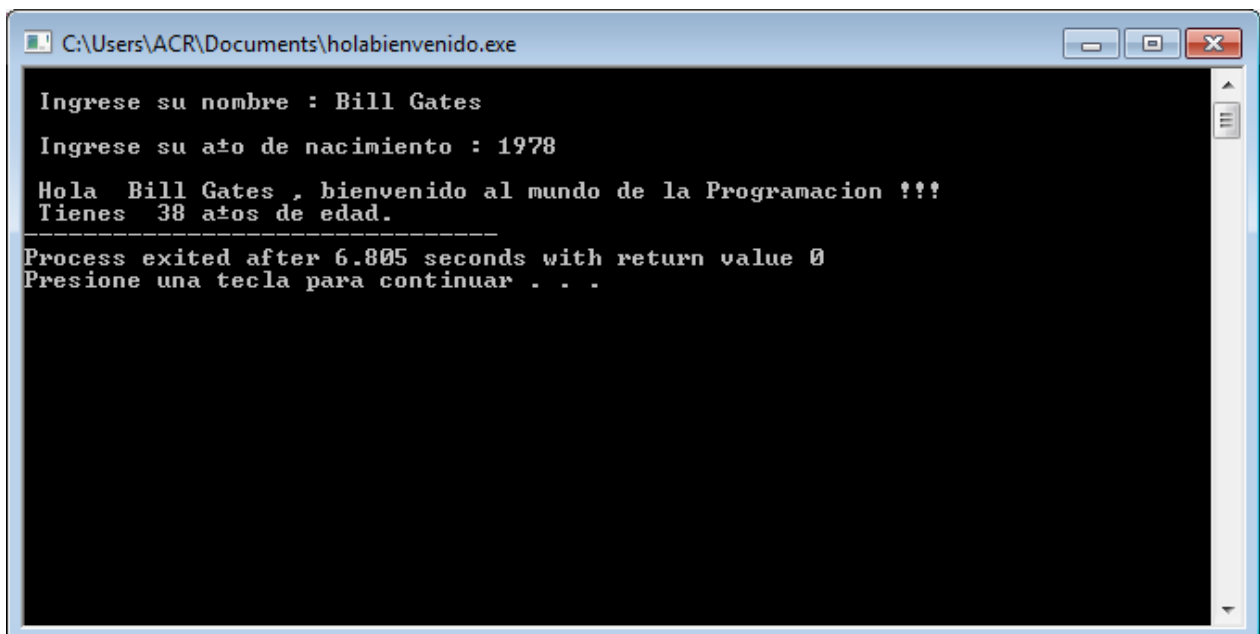
```
1  /*
2  EJEMPLO CAPTURAR NOMBRE
3  09 MAYO 2016
4  ALVARO CÉSPEDES ROMERO
5  */
6  #include<iostream>
7  #include<conio.h>
8  using namespace std;
9
10 int main()
11 {   char nombre[20];
12     cout<<"\n Ingrese su nombre : ";
13     gets(nombre);
14     cout<<"\n Hola  "<<nombre<<" , bienvenido al mundo de la Programacion !!! ";
15
16
17 }
18
19
```



```
C:\Users\ACR\Documents\holabienvenido.exe

Ingrese su nombre : Bill Gates
Hola Bill Gates , bienvenido al mundo de la Programacion !!!
-----
Process exited after 67.75 seconds with return value 0
Presione una tecla para continuar . . .
```

Digite el código de ejemplo. Ahora modifique el código de ejemplo, para que además de pedir el nombre, pida el año de nacimiento. Y elabore un código fuente que muestre algo similar a esta salida:



```
C:\Users\ACR\Documents\holabienvenido.exe

Ingrese su nombre : Bill Gates
Ingrese su año de nacimiento : 1978
Hola Bill Gates , bienvenido al mundo de la Programacion !!!
Tienes 38 años de edad.
-----
Process exited after 6.805 seconds with return value 0
Presione una tecla para continuar . . .
```

// Posterior al mapa conceptual de objetos y la inducción por parte del Instructor.

Hasta el momento, habíamos realizado lo que se denominaba programación estructurada donde indicaba que todo programa puede escribirse utilizando únicamente las tres instrucciones de control siguientes:

- Secuencia
- Instrucción condicional.
- Iteración (bucle de instrucciones) con condición al principio.

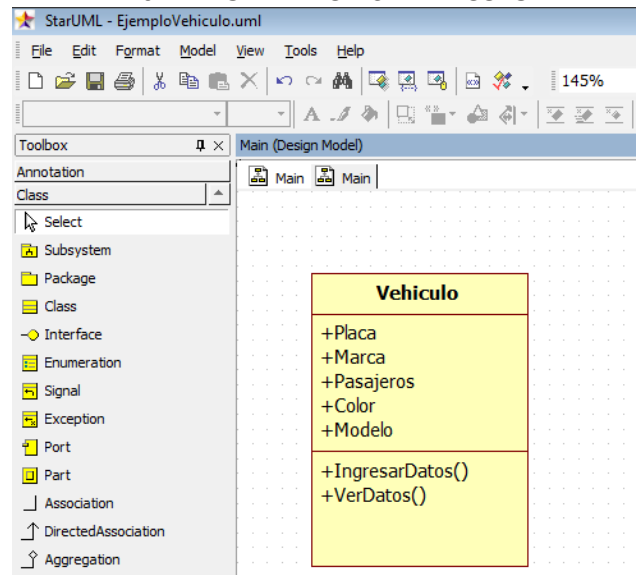
Durante nuestra primera parte de trabajo en un lenguaje de programación, trabajamos solamente en este enfoque, pues lo hacíamos para acostumbrarnos al uso de un lenguaje, a observar las reglas de una sintaxis.

Sin embargo, los programas y aplicaciones actuales han evolucionado desde la década pasada al uso de la programación orientada a objetos. Este tipo de programación, nos permite identificar lo que llamaremos objetos. Que son básicamente unidades lógicas, el ver las cosas en el estilo natural.

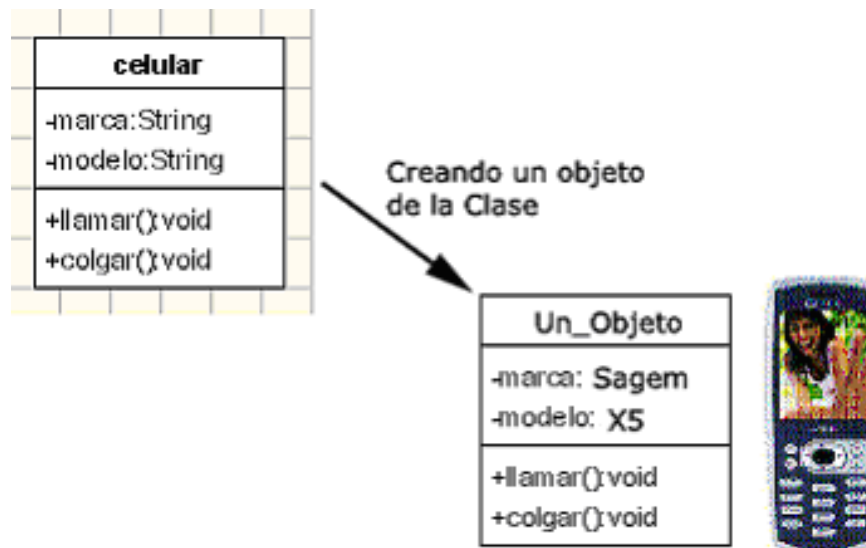
Esto nos trae ventajas, como la reutilización de código, la optimización y el mayor orden dentro de nuestro código fuente. Cada objeto podemos llevarlo a una Clase. (**class en inglés**). Aquí podemos determinar a los objetos, como unidades que se relacionan entre sí para realizar acciones. Cada objeto tiene unas características que llamamos propiedades (en algunos entornos de programación las llaman atributos).

Lenguajes de programación que soportan POO : C++, Java, Visual Basic .Net. Así pues, la POO, es el estándar actual para desarrollo, por esta razón, **debemos pensar ya en POO**.

EJEMPLO DE LA CLASE VEHICULO



Imaginemos que necesitamos realizar un programa para gestionar información de un vehículo, entonces identificamos las propiedades del vehículo. Como son por ejemplo placa, marca, pasajeros, color, modelo (Aquellas que sean necesarias para el problema que estemos tratando). Las propiedades también son llamadas atributos de la clase.



Del mismo modo que cuando antes declarábamos una variable como “int a;”, ello implicaba que la variable a, guardaría un número. Pues ahora podemos crear nuestras propias clases, donde podemos almacenar toda la información referente a un vehículo, **enlazando/agrupando** todos esos datos dentro **una sola variable** de tipo vehículo.

Ejemplo Vehículo x;

Donde x, es una variable de la clase Vehículo, por lo tanto x, tendra campos para almacenar Placa, marca, pasajeros, color, modelo.

Veamos otro ejemplo de POO en C++. Imaginemos que necesitamos gestionar informacion sobre madres de familia. Entonces, lo que hacemos es definir una clase llamada Mama, y definimos el numhijos, como el numero de hijos que cada mamá tiene. Veamoslo con la siguiente captura de pantalla:

```
/******  
DESARROLLADOR: ING ALVARO CÉSPEDES ROMERO  
15 DE NOVIEMBRE DE 2014  
CODIGO FUENTE DE EJEMPLO PARA ILUSTRAR CLASES EN C++  
*****/  
  
// ESTIMADO APRENDIZ, RECUERDE COMENTAR SUS CODIGOS FUENTES  
// PRACTICA MUY RECOMENDABLE PARA LLEVAR UN CONTROL EN EL DESARROLLO  
  
#include<iostream.h>  
#include<conio.h>  
  
class Mama{ // SUGERENCIA : LA PRIMERA LETRA EN MAYUSCULA  
public:  
    int numhijos; // numhijos ES MIEMBRO DE LA CLASE Mama  
};  
  
int main()  
{Mama x; // X ES UNA VARIABLE DE TIPO CLASE  
  
    cout<<"\n Ingrese Numero de Hijos ";  
    cin>>x.numhijos;  
  
    cout<<"\n El numero de hijos es : ";  
    cout<<x.numhijos;  
    getch();  
return 0;  
}
```

De manera individual digite el codigo fuente mostrado en la anterior imagen, guardelo con el nombre mama.cpp.

Ejecutelo varias veces, para que observe su funcionamiento, verifique que no haya errores de sintaxis. Acostumbrese a comentar sus codigos fuentes, de no hacerlo es probable que pasado meses despues de escribirlos no recuerde para que sirve y necesite ejecutarlos o leerlos varias veces, el comentar nos ayuda a ser mas rapidos a la hora de encontrar codigo fuente y poder "reutilizarlo".

Realice las modificaciones necesarias al código anterior, para que ahora, no maneje una sola variable tipo Mama, sino 2, en el ejemplo está la variable **x**, ahora también la **y**.

```

/*****
DESARROLLADOR: ING ALVARO CÉSPEDES ROMERO
15 DE NOVIEMBRE DE 2014
CODIGO FUENTE DE EJEMPLO PARA ILUSTRAR CLASES EN C++
*****/

#include<iostream.h>
#include<conio.h>

class Mama{
public:
    int numhijos;
};

int main()
{Mama x,y;
    // Mama y; TAMBIEN ES VALIDO DECLARLA LA VARIABLE CLASE MAMA EN OTRA LINEA

    cout<<"\n Ingrese Numero de Hijos de la Mamá x :";
    cin>>x.numhijos;
    cout<<"\n Ingrese Numero de Hijos de la Mamá y : ";
    cin>>y.numhijos;

    cout<<"\n El numero de hijos de la Mamá x es : ";
    cout<<x.numhijos;
    cout<<"\n El numero de hijos de la Mamá y es : ";
    cout<<y.numhijos;

    getch();
return 0;
}

```

De manera individual digite el código fuente mostrado en la anterior imagen, guárdelo con el nombre dosmadres.cpp.

Verifique que el programa funcione adecuadamente.

EJERCICIO 1:

Ubique un compañero/a de trabajo, e identifique 7 objetos diferentes, piense en ellos identifique las propiedades de cada uno. Escriba cada una de ellas en un código fuente diferente. Para cada clase defina sus propiedades, realice el código para capturar y el código para mostrar la información de CADA CLASE.

No puede copiar ejercicios de otros grupos de trabajo. El pensar hace parte indispensable de nuestra labor como desarrolladores.

Sugerencia: tome los objetos de la cotidianidad. La programación POO se basa en la representación de los mismos.

EJERCICIO 2:

Estimado/a aprendiz, ahora para mediante un trabajo grupal propuesto y orientado por el Instructor participa activamente para dar solución a las operaciones básicas de multiplicar y dividir 2 números fraccionarios. Esta actividad se realizará en conjunto con todos los aprendices. Imagina números a, b, c, que sean fraccionarios.

Tras finalizar el planteamiento y tu participación, si tienes dudas plantealas al Instructor. Luego ubica tu compañero de trabajo y procede a realizar el programa que resuelva esas operaciones.

CONSULTAR Y RESOLVER:

Ahora bien como se dejó para consultar que es un Método en POO??

