¿Qué es void?

     El tipo void especifica un conjunto vacío de valores. En realidad void no es un tipo como lo son int, char, float, etc., aunque por la forma de utilizarlo si lo comparamos con la forma de funcionamiento a los otros tipos fundamentales, se considera como tal. Por esta razón, no se puede declarar una variable de tipo void.

     void a;    //<---- ESTO NO SE PUEDE HACER.

     Void permite contruir mis propias funciones, para luego utilizarlas en un programa.

     Por ejemplo:

     void calcular(); //<---- ESTA ES LA FORMA CORRECTA DE NOMBRARLO.

     El tipo void se utiliza para indicar que una función no acepta argumentos.

     En el siguiente ejemplo, void indica que la función fx no tiene argumentos.

     double fx(void);

     Para declarar funciones que no retornan un valor. En el siguiente ejemplo, void indica que la función fy no retorna un valor.

     void fy (int, int);

     Para declarar un puntero genérico, como veremos más adelante; esto es, un puntero a un objeto de tipo aún desconocido.

     void \*p;

     Los ejemplos anteriores declaran la función fx, como una función sin argumentos que devuelve un valor de tipo real de doble precisión; por otro lado la función fy es una función con dos argumentos de tipo int que no devuelve valor alguno; y por ultimo un puntero genérico p.

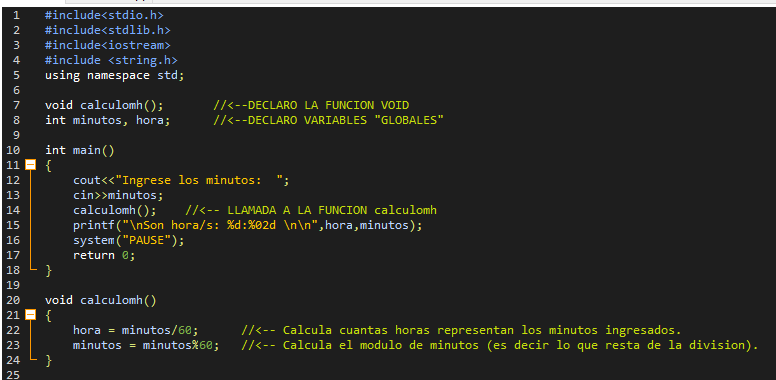
    Void se utiliza para agrupar un conjunto de sentencias en una sola unidad la cual puede ser llamada desde cualquier parte del programa o desde otra función. La estructura de trabajo será la siguiente:

# include <bibliotecas.h>  
  
void funcion(); //<-- Debemos nombrar la "funcion".  
  
   
  
main()  
  
{  
  
instrucciones del programa ;  
  
línea de código…  
  
funcion() ;  
  
línea de código…  
  
línea de código…  
  
funcion() ;  
  
línea de código…  
  
línea de código…  
  
}  
  
   
  
void funcion ()  
  
{  
  
instrucciones de la funcion ;  
  
instrucciones de la funcion ;  
  
.......  
}  
  
En esta clase desarrollaremos ejercitaciones con métodos de tipo void con y sin parámetros.

**SIN PARAMETROS:**

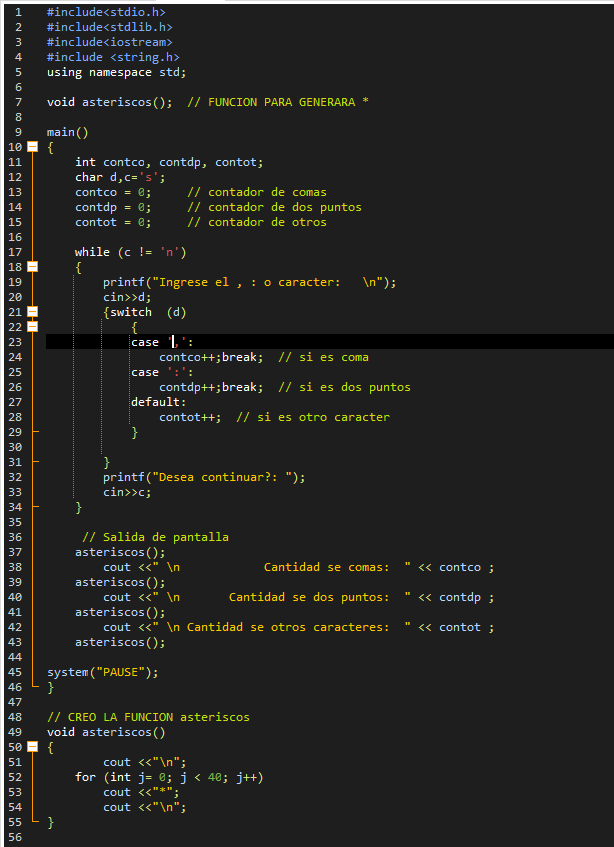
     Primero realizaremos un programa que dada una cantidad de minutos nos permita calcular su equivalente en horas. En este ejercicio no se incluirá parámetros para la función y lo que vamos a realizar es una llamada a VOID calculomh la cual estará desarrollada fuera del programa principal main y que resolverá nuestra ecuación para convertir minutos a horas.

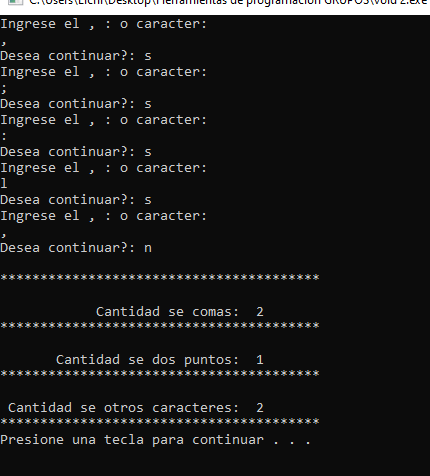
**\*Ejemplo VOID 1:**



     Nótese que realizo la llamada a la función en la línea 14 del programa y que luego de haberse realizado continua normalmente con las siguientes líneas de código. En el printf de la línea 15 vuelvo a utilizar %02d para los minutos asi en el caso de ingresar por ej. 308 min me va a mostrar 5:08 de otra forma si yo utilizara %d me colocaría 5:8 obviando el 0 antes del 8.

**\*Ejemplo VOID 2:**





     En este ejemplo utilizo la función void varias veces en las líneas de código 37/39/41/43 para crear una línea delimitante de asteriscos, dentro de la función void asteriscos con un for hago que se repita el símbolo 40 veces logrando este efecto.

**UTILIZANDO PARAMETROS:**

     En el caso de querer incluir parámetros para la función, es decir utilizar valores de algún tipo para resolver ecuaciones construiremos la estructura de la función void de la siguiente forma:

void metodo (variables)

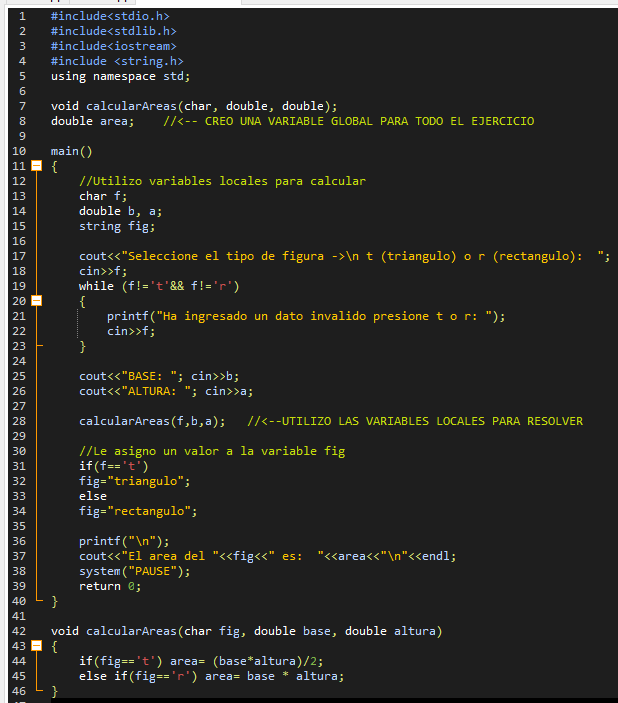
{

líneas de código….

}

Veamos un ejemplo en el que codificaremos un programa que permita calcular áreas de figuras geométricas, como un triángulo y un rectángulo. Utilizaremos los caracteres "t" para triangulo y "r" para rectángulo, según la figura es al tipo de calculo que realizara nuestra función calcularAreas.

**\*Ejercicio VOID 3:**

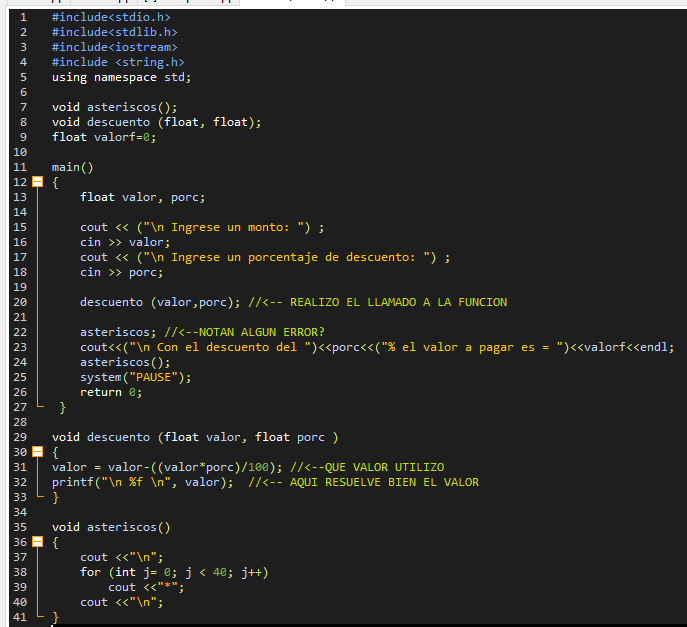


        En este ejercicio se puede ver que utilizamos variables globales (double area) y variables locales (figura/f, base/b, altura/a) para la resolución de la función void calcularAreas.

     Observamos la necesidad de transportar un resultado (área) por todo el programa es por ello que la generamos globalmente sin embargo las ecuaciones se llevan a cabo teniendo en cuenta las variables que se encuentran declaradas localmente. Esto es realmente importante ya que podríamos declarar variables que cumplan con los parámetros que necesitamos es decir que sean del tipo char, int, doublé, etc. y que realicen el cálculo correspondiente gracias a que declaramos parámetros dentro de la función.

     Por último, se crea la función: void calcularAreas (char fig, double base, double altura) en este punto le indicamos que los valores locales “f, b, a” corresponden al char para fig / double para base / double para altura. La función interpreta y traduce los valores locales a valores de la función y resuelve área que es un valor global y que se muestra al finalizar el cálculo. Tengamos en cuenta que en este caso tome la decisión de llamar al char que utiliza calcularAreas fig para no confundirlos, pero podría haber utilizado también “f” ya que las mismas son variables locales para resolver.

**\*Ejercicio VOID 4:**



     En este último ejercicio vamos a mezclar todos los conocimientos adquiridos con la funcion VOID, la idea va a ser calcular un descuento y para ello voy a crear void descuento que utilizara 2 variables del tipo float una para asignarle el valor a pagar y otra con el porcentaje de descuento a aplicar. Además, al ejemplo le voy a agregar otra función asteriscos para que me separe el resultado final y lo muestre de forma destacada en pantalla.

     Para completar el ejercicio voy a crear una variable global valorf que me permitirá mostrar el resultado de los cálculos en el programa main. Observemos que en este caso en particular decidí llamar a las variables locales que utiliza el main (valor, porc) igual que las variables locales que utiliza descuento (valor, porc) de esta forma espero que puedan comprender la pluralidad de las variables, ya si bien se llaman igual no son las mismas variables. Finalmente y para comprobar lo antedicho, dentro del void he decidido incluir un printf que me muestre que resuelve bien la ecuación del descuento sin embargo en el ejercicio no resuelve el resultado, se entiende porque?. Copien los ejercicios y vayan comprobando los resultados.

**\*\*Para finalizar la clase quiero recordarles** a tod@s Uds. mis querid@s alumnos que deben revisar el area **"AVISOS"** donde se encuentran todas las novedades semanales de la materia, para cualquier duda o consulta que pueda surgir tenemos disponible el FORO de **"CONSULTAS**" así como también mi mail personal eshimoyama@esbabarrionorte.edu.ar.