|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Ing. Claudia Rodríguez Espino. |
| *Asignatura:* | Fundamentos de Programación. |
| *Grupo:* | 03 |
| *No de Práctica(s):* | 12 |
| *Integrante(s):* | Flores Constantino Diego |
|  |  |
|  |  |
| *Semestre:* | 2018-2 |
| *Fecha de entrega:* | 17/05/18 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Guía Práctica de estudio 11: Arreglos unidimensionales y multidimensionales.**

**Objetivo:**

Elaborar programas en C donde la solución del problema se divida en funciones. Distinguir lo que es el prototipo o firma de una función y la implementación de ella, así como manipular parámetros tanto en la función principal como en otras.

**Desarrollo:**

1. *Codifique en lenguaje C una calculadora, con formato de menú, emplee funciones.*

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#include<windows.h>

void suma(double d,double c)

{

double e;

e=c+d;

printf("\nEl valor de la suma es: %.2lf\n",e);

}

void resta(double c,double d)

{

double e;

e=c-d;

printf("\nEl valor de la resta es %.2lf\n",e);

}

void mult(double d,double c)

{

double e;

e=c\*d;

printf("\nEl valor de la multiplicacion es: %.2lf\n",e);

}

void div(double c,double d)

{

double e;

e=c/d;

printf("\nEl valor de la division es %.2lf\n",e);

}

char r,a;

char user[20];

main()

{

printf("Bienvenido(a)\n");

printf("Por favor, ingresa tu nombre de usuario (maximo 20 caracteres)\n");

char user[20];

scanf(" %[^\n]",&user);

system("cls");

do{

printf("Hola %s\n\n",user);

Sleep(100);

puts("CALCULADORA\n");

puts("A. Suma\n");

puts("B. Resta\n");

puts("C. Multiplicacion\n");

puts("D. Division\n");

printf("Elige una opcion: ");

scanf("%s",&a);

a=toupper(a);

system("cls");

switch(a)

{

case 'A':

{

double x,y;

printf("Ingresa dos valores que quieras sumar: ");

scanf("%lf%lf",&x,&y);

suma(x,y);

}

break;

case 'B':

{

double z,w;

printf("Ingresa dos valores que quieras restar: ");

scanf("%lf%lf",&z,&w);

resta(z,w);

}

break;

case 'C':

{

double a,b;

printf("Ingresa dos valores que quieras multiplicar: ");

scanf("%lf%lf",&a,&b);

mult(a,b);

}

break;

case 'D':

{

double n,m;

printf("Ingresa dos valores que quieras dividir: ");

scanf("%lf%lf",&n,&m);

div(n,m);

}

break;

default:

printf("La opcion que seleccionaste no exise");

break;

}

printf("\n%s, deseas repetir el proceso? s/n)?",user);

scanf("%s",&r);

r=tolower(r);

system("cls");

} while(r=='s');

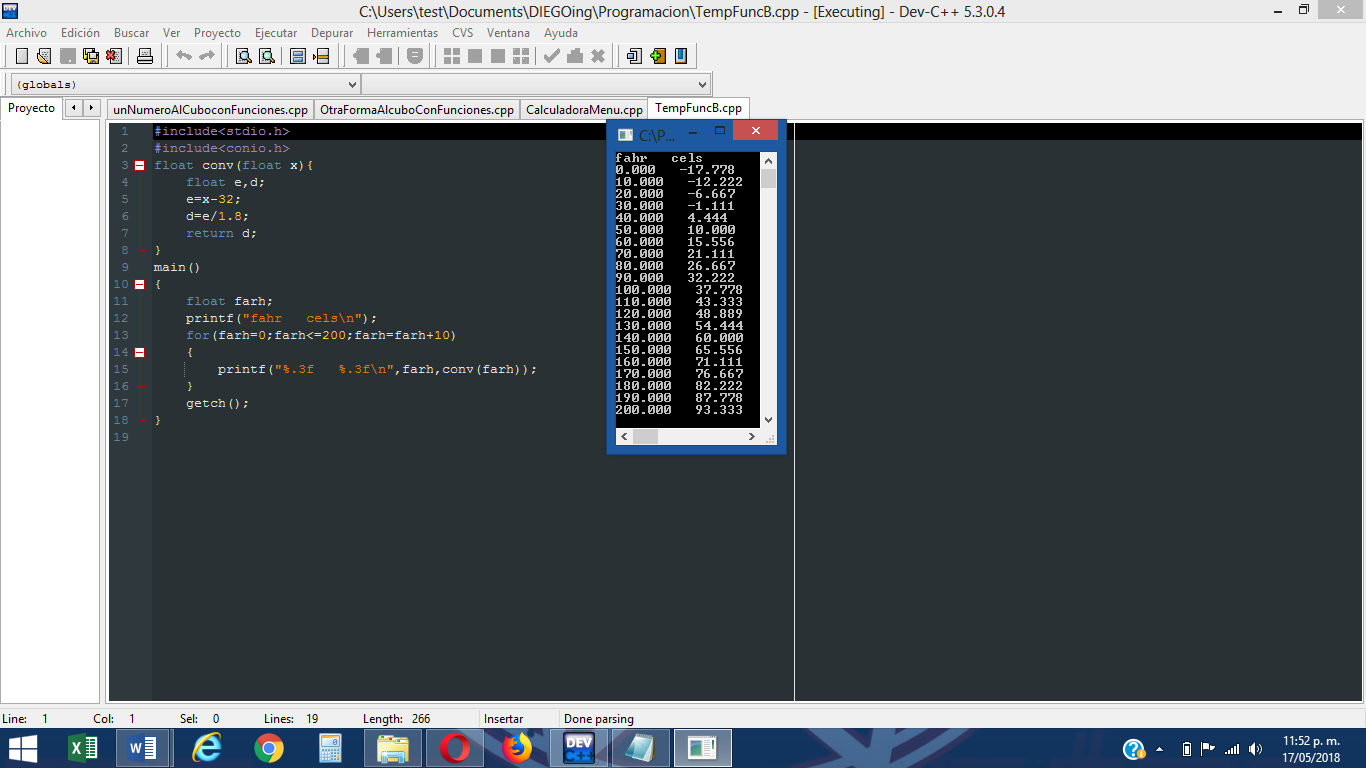
printf("\tHasta luego, %s.",user);

getchar();

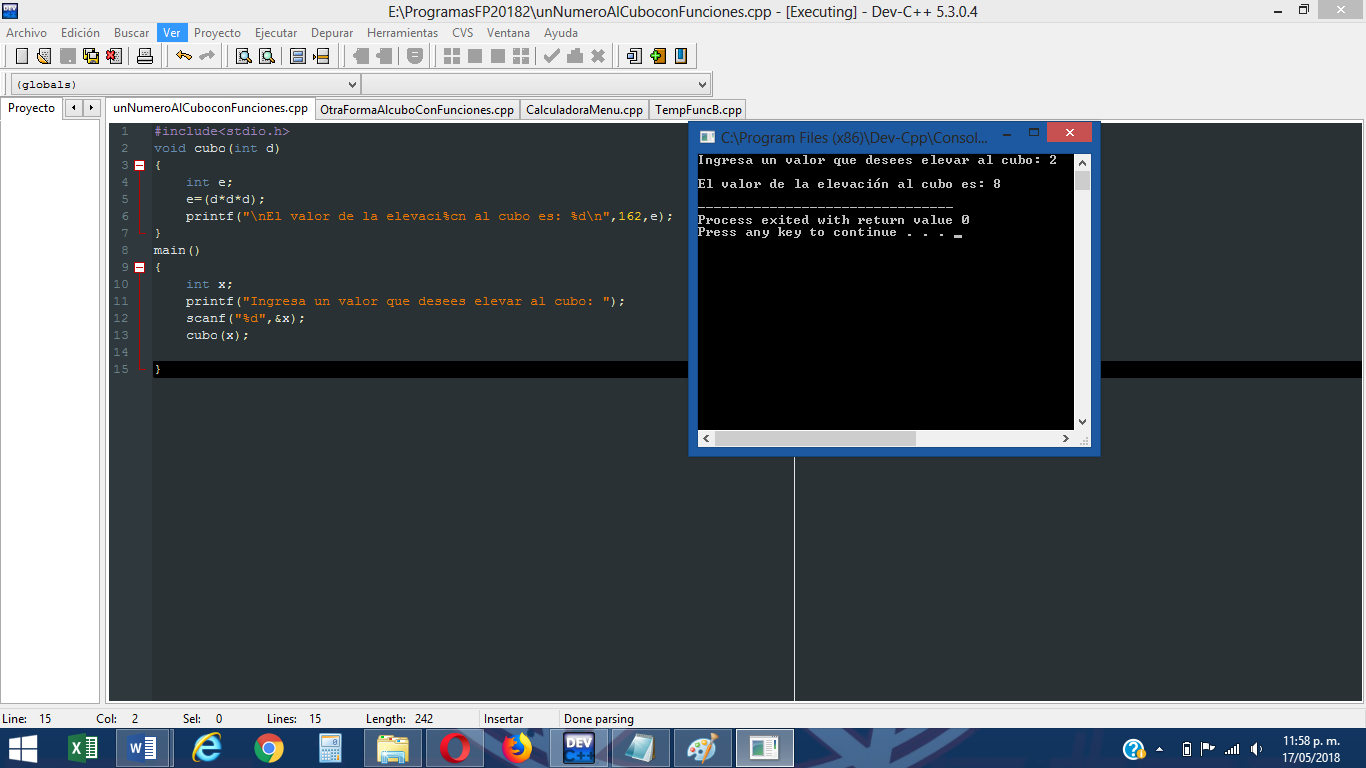
getchar();

}

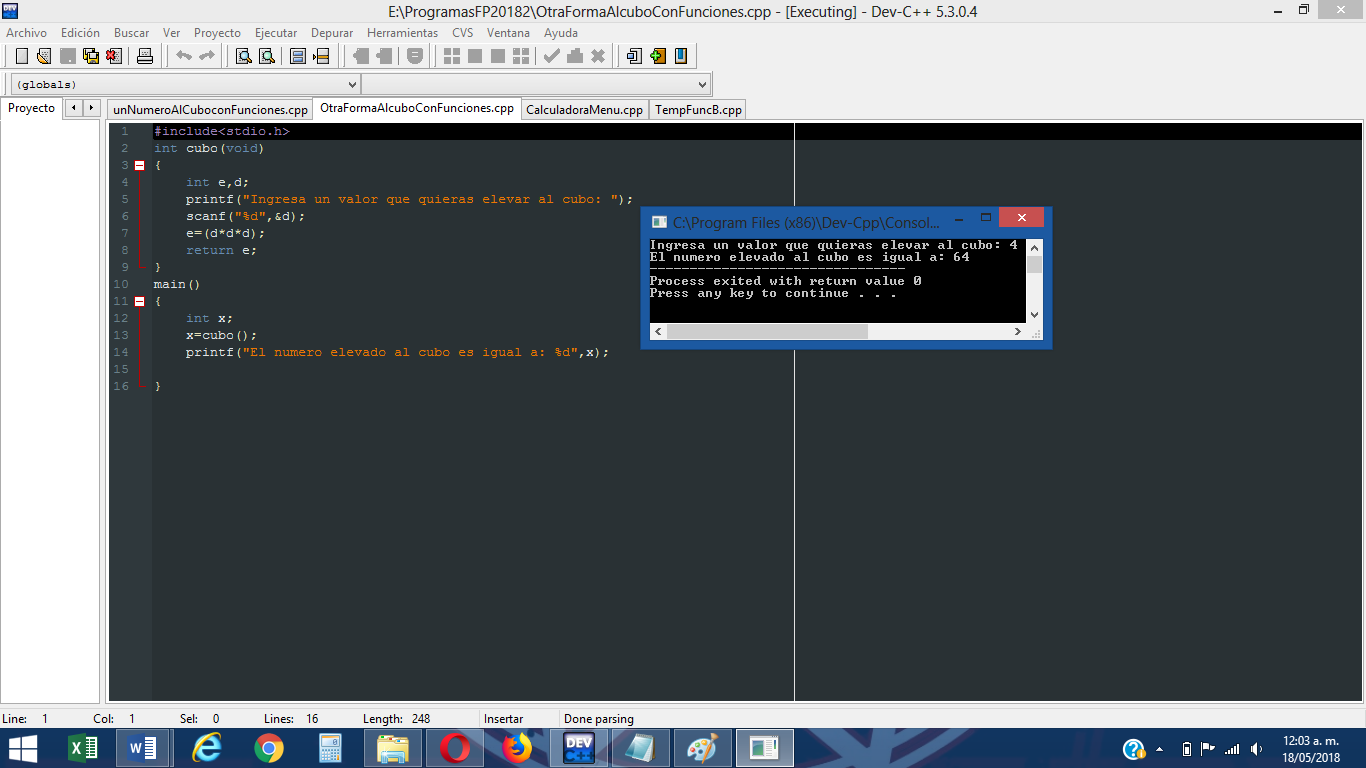
1. *Realice un programa que realice la conversión de grados centígrados a grados Fahrenheit, utilice funciones.*

******

1. *Realice un programa que eleve un número al cubo. Utilice una función con argumentos que NO devuelva un valor, que en la función main se pida el número que se elevará al cubo.*

**

1. *Realice un programa que eleve un número al cubo. Utilice una función sin argumentos que devuelva un valor, que en la función main se imprima el valor.*

******

**Conclusiones.**

* Trabajar con funciones al programar, resulta una de las maneras más formales, pues resulta más fácil detectar errores, pues parte en diferentes módulos al programa.
* Existen diferentes tipos de funciones, las cuales deberemos identificar de acuerdo al problema a resolver para tener presente cual es la más apropiada o cómoda de utilizar, sin perder de vista el objetivo.
* Las funciones y las clases tienen utilidades similares, sin embargo, las clases (en la programación orientada a objetos) resultan más poderosas, pues poseen la capacidad de heredar ciertos parámetros en diferentes partes del programa, así como también cuentan con polimorfismo, modularidad, etc., características que dejan a las funciones carentes de capacidad aunque, a pesar de esto, no pierden su gran utilidad, pues el programador posee una gran herramienta de trabajo si las funciones son empleadas en la codificación.