



UNIVERSIDAD MARIANO GALVEZ DE GUATEMALA

Curso: Programación, Jornada Matutina

Sección: "A"

Catedrática: Ing. Carlos Alejandro Arias.



Carnet: __5090-23-1762

Nombre completo: Roberto Carlos de León Gramajo

Fecha entrega: 08/04/2024

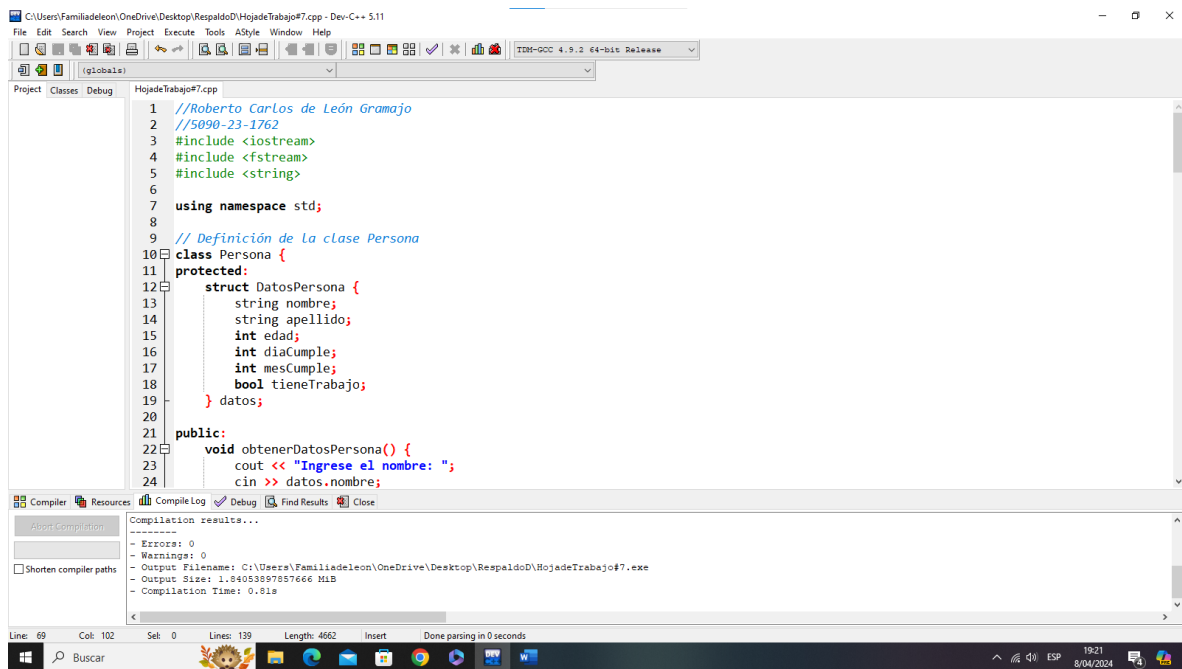
Tema tarea: Hoja de Trabajo#7

Introducción

Realizar un trabajo de programación donde se utilicen las herramientas vistas en clase, toda programación orientada en objetos, llevando clases privadas, publicas, protegidas, la clase central se llama Persona, la cual hereda una clase llamada Estudiante, todo registro de información debe de tener la biblioteca "fstream", nos ayudara a hacer archivos en la computadora, guardado todo registro de información.

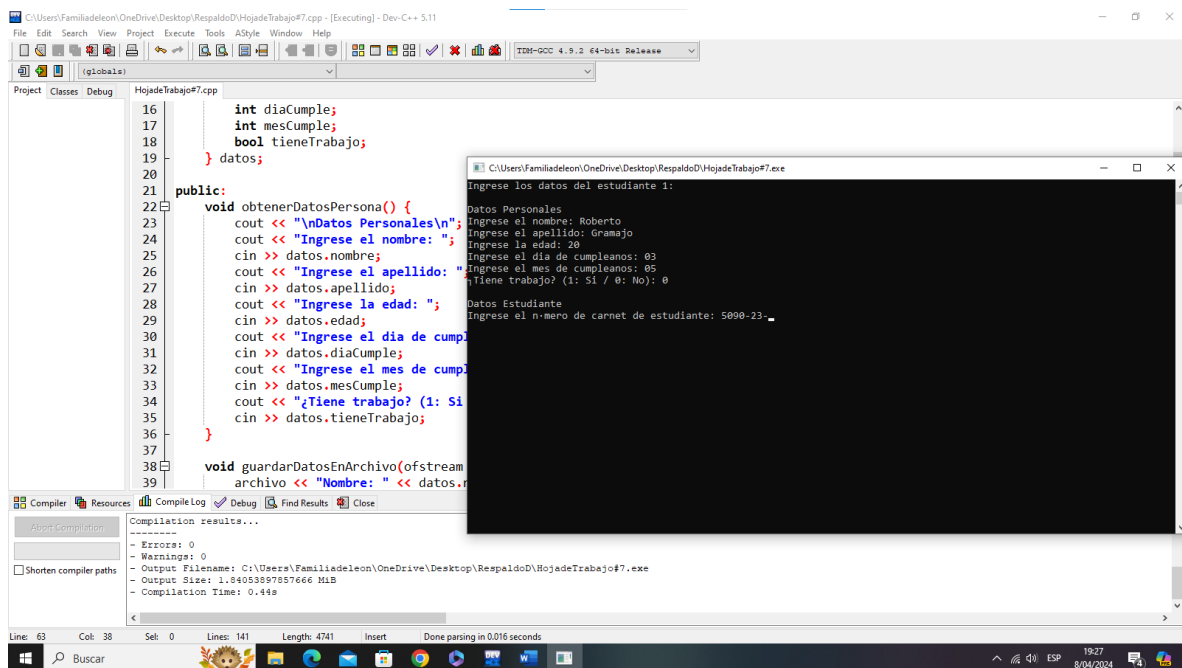
Debe de recordad en utilizar string, con su biblioteca respectiva llamada "string", esto nos ayuda a identificar una serie de cadenas por medio de símbolo o palabras utilizados que el usuario ingresa, ordenar todo atributo por un struct, para mayor facilidad de utilidad para string.

Programa



```
1 //Roberto Carlos de León Gramajo
2 //5090-23-1762
3 #include <iostream>
4 #include <fstream>
5 #include <string>
6
7 using namespace std;
8
9 // Definición de la clase Persona
10 class Persona {
11 protected:
12     struct DatosPersona {
13         string nombre;
14         string apellido;
15         int edad;
16         int diaCumple;
17         int mesCumple;
18         bool tieneTrabajo;
19     } datos;
20
21 public:
22     void obtenerDatosPersona() {
23         cout << "Ingrese el nombre: ";
24         cin >> datos.nombre;
```

Iniciamos nuestro código definiendo datos personales, como también definiendo las bibliotecas principales para el programa como son, iostream, fstream, y string.

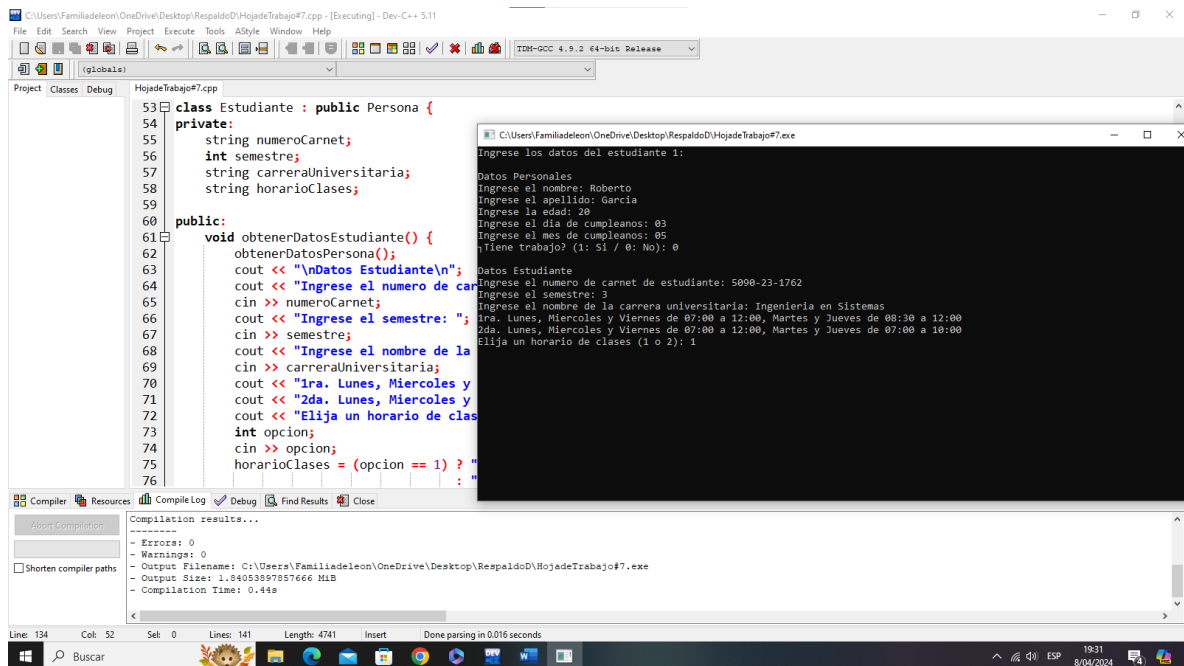


```
16     int diaCumple;
17     int mesCumple;
18     bool tieneTrabajo;
19 } datos;
20
21 public:
22     void obtenerDatosPersona() {
23         cout << "\nDatos Personales\n";
24         cout << "Ingrese el nombre: ";
25         cin >> datos.nombre;
26         cout << "Ingrese el apellido: ";
27         cin >> datos.apellido;
28         cout << "Ingrese la edad: ";
29         cin >> datos.edad;
30         cout << "Ingrese el día de cumpleaños: ";
31         cin >> datos.diaCumple;
32         cout << "Ingrese el mes de cumpleaños: ";
33         cin >> datos.mesCumple;
34         cout << "¿Tiene trabajo? (1: Si / 0: No): ";
35         cin >> datos.tieneTrabajo;
36     }
37
38     void guardarDatosEnArchivo(ofstream& archivo) {
39         archivo << "Nombre: " << datos.nombre << "\n";
```

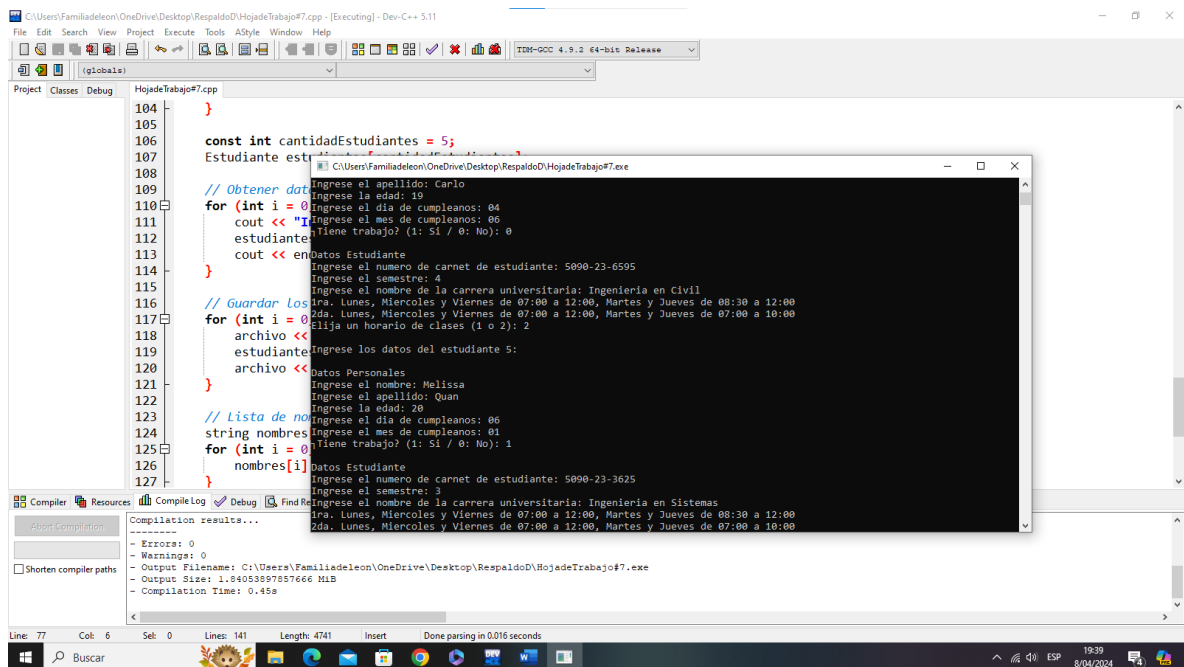
Output:

```
Ingrese los datos del estudiante 1:
Datos Personales
Ingrese el nombre: Roberto
Ingrese el apellido: Gramajo
Ingrese la edad: 20
Ingrese el día de cumpleaños: 03
Ingrese el mes de cumpleaños: 05
¿Tiene trabajo? (1: Si / 0: No): 0
Datos Estudiante
Ingrese el n-mero de carnet de estudiante: 5090-23-1762
```

La primer clase nos ayudara a ingresar datos de una persona en general, como es su nombre, apellido, edad, día de cumpleaños, mes de cumpleaños, y si trabaja.



Al tener todos los datos personales, ingresamos los datos de estudiante, obteniendo así más información del usuario, debemos de recordad que a nivel de programación, esta es una clase heredada de la clase persona.



Ingresamos 5 estudiantes a nuestro sistema, los cuales son evaluados por varias características como el nombre de su carrera, y el semestre al que pertenecen.

```
77  
78  
79 void guardarDatosEnArchivo(ofstream &archivo) {  
80     Persona p;  
81     archivo << "Elija un horario de clases (1 o 2): ";  
82     archivo << "Ingrese los datos del estudiante 5:";  
83     archivo << "Datos Personales  
84     Ingrese el nombre: Melissa  
85     Ingrese el apellido: Quan  
86     Ingrese la edad: 20  
87     Ingrese el día de cumpleaños: 00  
88     Ingrese el mes de cumpleaños: 01  
89     Ingrese si tiene trabajo? (1: Si / 0: No): 1  
90     // Función para el ordenamiento de burbujas  
91     void ordenarNombres(string nombres[]) {  
92         for (int i = 0; i < nombres.size() - 1; i++)  
93             for (int j = i + 1; j < nombres.size(); j++)  
94                 if (nombres[i] > nombres[j])  
95                     swap(nombres[i], nombres[j]);  
96     }  
97  
98     int main() {  
99         ofstream archivo("Hohadetrabajo7.txt");  
100     }
```

Process exited after 107.6 seconds with return value 0
Presione una tecla para continuar . . .

Utilizamos un ordenamiento de burbuja para los estudiantes, con sus nombre, y saber cual iría primero.

```
77  
78  
79 void guardarDatosEnArchivo(ofstream &archivo) {  
80     Persona p;  
81     archivo << "Número de carnet: 5090-23-1762";  
82     archivo << "Semestre: 3";  
83     archivo << "Carrera: Ingeniería en Sistemas";  
84     archivo << "Horario: Lunes, Miércoles y Viernes de 07:00 a 12:00, Martes y Jueves de 08:00 a 12:00";  
85     archivo << "Datos del estudiante 1:  
86     Nombre: Roberto  
87     Apellido: Garcia  
88     Edad: 20  
89     Cumpleaños: 3/5  
90     Tiene trabajo: No  
91     Número de carnet: 5090-23-1762  
92     Semestre: 3  
93     Carrera universitaria: Sistemas  
94     Horario de clases: Lunes, Miércoles y Viernes de 07:00 a 12:00, Martes y Jueves de 08:00 a 12:00";  
95     archivo << "Datos del estudiante 2:  
96     Nombre: James  
97     Apellido: Hank  
98     Edad: 21  
99     Cumpleaños: 1/1  
100     Tiene trabajo: No  
101     Número de carnet: 5090-23-1762  
102     Semestre: 3  
103     Carrera universitaria: Ingeniería en Sistemas  
104     Horario de clases: Lunes, Miércoles y Viernes de 07:00 a 12:00, Martes y Jueves de 08:00 a 12:00";  
105     archivo << "Nombres de estudiantes ordenados:  
106     Angelyn  
107     Sean  
108     James  
109     Melissa  
110     Roberto";  
111     archivo.close();  
112 }
```

Process exited after 107.6 seconds with return value 0
Presione una tecla para continuar . . .

Por ultimo verificamos que estén cada uno de los estudiantes en el bloq de notas que creamos, al iniciar el programa, el archivo se llama, Hohadetrabajo7.txt.

GitHub de mi programa, donde se encuentra código completo, detallando cada paso realizado. [RobertodeLeonUmg/HojadeTrabajo-7: Programación orientado a objetos, con clases, y procedimientos de todo \(github.com\)](https://github.com/RobertodeLeonUmg/HojadeTrabajo-7: Programación orientado a objetos, con clases, y procedimientos de todo (github.com))

Conclusiones

Fue un trabajo llamativo, gracias a que no había utilizado una programación orientada en objetos, con un sistema de guardado entre archivos de la computadora, utilizando bibliotecas en concreto, algo que se me complicó fue utilizar string, yo pensaba que era la sustitución de char, pero no solo es eso, sino se debe de saber como utilizar para utilizar de la forma correcta, como en este caso es por medio de struct, facilitando su uso, y ayudando a organizar mejor todos nuestros atributos.