Universidad Mariano Gálvez De Guatemala

Facultad de ingeniería, Matemática y ciencias Físicas

Campus Villa Nueva Guatemala

Ingeniería en Sistemas de Información y ciencias de la computación

Curso: programación 1

Código de curso: 012

Código de la carrera: 5090

Estudiante: COOSEMANS Spiegeler Max Eduardo

Sección: A

Numero de carné: 5090-23-3282

Fecha: 17/04/2024

Consulta de Empleados  
Instrucciones:  
• Implementa un programa que lea los datos del archivo empleados.txt; creado en el Programa 1.  
• Crea una función que reciba el nombre completo de un empleado como entrada y muestre su  
información completa (nombre, apellido, edad y salario) si existe en el archivo.  
• Permite al usuario realizar consultas múltiples de empleados hasta que decida salir del programa.  
• Asegúrate de manejar adecuadamente el caso en el que el empleado buscado no exista en el archivo.  
2. Registro de Empleados  
Instrucciones Programa 1:  
• Diseña una clase llamada Empleado con los siguientes atributos: nombre, apellido, edad y salario.  
• Implementa métodos para establecer y obtener los valores de los atributos.  
• Crea un programa principal que permita al usuario ingresar los datos de varios empleados y los  
almacene en un archivo de texto llamado empleados.txt;.  
• Asegúrate de que el programa pueda manejar la escritura y la lectura de los datos del archivo de  
manera adecuada.

#include <iostream>

#include <fstream>

#include <string>

// Estructura para almacenar datos de un empleado

struct Empleado {

std::string nombre;

std::string apellido;

int edad;

double salario;

};

// Función para buscar y mostrar información de un empleado

void consultarEmpleado(const std::string& nombreCompleto, const std::vector<Empleado>& empleados) {

for (const Empleado& emp : empleados) {

if (emp.nombre + " " + emp.apellido == nombreCompleto) {

std::cout << "Información del empleado:\n";

std::cout << "Nombre: " << emp.nombre << " " << emp.apellido << "\n";

std::cout << "Edad: " << emp.edad << " años\n";

std::cout << "Salario: $" << emp.salario << "\n";

return;

}

}

std::cout << "Empleado no encontrado.\n";

}

int main() {

// Leer datos del archivo empleados.txt (suponiendo que está en el mismo directorio)

std::ifstream archivo("empleados.txt");

std::vector<Empleado> empleados;

Empleado emp;

while (archivo >> emp.nombre >> emp.apellido >> emp.edad >> emp.salario) {

empleados.push\_back(emp);

}

// Consulta de empleados

std::string nombreBuscado;

do {

std::cout << "Ingresa el nombre completo del empleado (o 'salir' para terminar): ";

std::getline(std::cin, nombreBuscado);

if (nombreBuscado != "salir") {

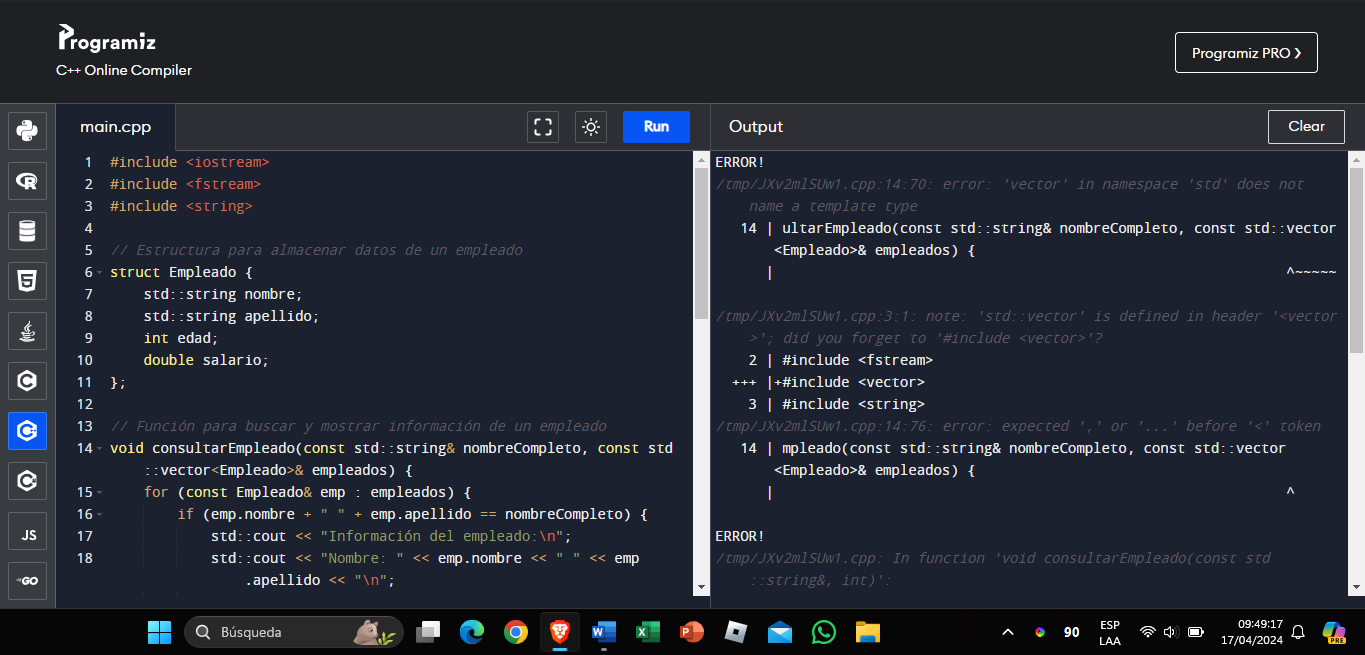
consultarEmpleado(nombreBuscado, empleados);

}

} while (nombreBuscado != "salir");

return 0;

}



#include <iostream>

#include <fstream>

#include <string>

class Empleado {

public:

// Constructor

Empleado(const std::string& nombre, const std::string& apellido, int edad, double salario)

: nombre\_(nombre), apellido\_(apellido), edad\_(edad), salario\_(salario) {}

// Métodos para obtener valores

std::string getNombre() const { return nombre\_; }

std::string getApellido() const { return apellido\_; }

int getEdad() const { return edad\_; }

double getSalario() const { return salario\_; }

private:

std::string nombre\_;

std::string apellido\_;

int edad\_;

double salario\_;

};

int main() {

std::ofstream archivo("empleados.txt", std::ios::app); // Abrir en modo de añadir

if (!archivo) {

std::cerr << "Error al abrir el archivo.\n";

return 1;

}

std::string nombre, apellido;

int edad;

double salario;

std::cout << "Registro de Empleados\n";

std::cout << "Ingresa los datos de un empleado:\n";

std::cout << "Nombre: ";

std::cin >> nombre;

std::cout << "Apellido: ";

std::cin >> apellido;

std::cout << "Edad: ";

std::cin >> edad;

std::cout << "Salario: ";

std::cin >> salario;

Empleado emp(nombre, apellido, edad, salario);

// Escribir datos en el archivo

archivo << emp.getNombre() << " " << emp.getApellido() << " " << emp.getEdad()

Captura de pantalla de computadora

Descripción generada automáticamente

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

// Estructura para almacenar datos de un empleado

struct Empleado {

std::string nombre;

std::string apellido;

int edad;

double salario;

};

// Función para buscar y mostrar información de un empleado

void consultarEmpleado(const std::string& nombreCompleto, const std::vector<Empleado>& empleados) {

for (const Empleado& emp : empleados) {

if (emp.nombre + " " + emp.apellido == nombreCompleto) {

std::cout << "Información del empleado:\n";

std::cout << "Nombre: " << emp.nombre << " " << emp.apellido << "\n";

std::cout << "Edad: " << emp.edad << " años\n";

std::cout << "Salario: $" << emp.salario << "\n";

return;

}

}

std::cout << "Empleado no encontrado.\n";

}

int main() {

// Leer datos del archivo empleados.txt (suponiendo que está en el mismo directorio)

std::ifstream archivo("empleados.txt");

std::vector<Empleado> empleados;

Empleado emp;

while (archivo >> emp.nombre >> emp.apellido >> emp.edad >> emp.salario) {

empleados.push\_back(emp);

}

// Consulta de empleados

std::string nombreBuscado;

do {

std::cout << "Ingresa el nombre completo del empleado (o 'salir' para terminar): ";

std::getline(std::cin, nombreBuscado);

if (nombreBuscado != "salir") {

consultarEmpleado(nombreBuscado, empleados);

}

} while (nombreBuscado != "salir");

return 0;

}

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente#include <iostream>

#include <fstream>

#include <string>

class Empleado {

public:

// Constructor

Empleado(const std::string& nombre, const std::string& apellido, int edad, double salario)

: nombre\_(nombre), apellido\_(apellido), edad\_(edad), salario\_(salario) {}

// Métodos para obtener valores

std::string getNombre() const { return nombre\_; }

std::string getApellido() const { return apellido\_; }

int getEdad() const { return edad\_; }

double getSalario() const { return salario\_; }

private:

std::string nombre\_;

std::string apellido\_;

int edad\_;

double salario\_;

};

int main() {

std::ofstream archivo("empleados.txt", std::ios::app); // Abrir en modo de añadir

if (!archivo) {

std::cerr << "Error al abrir el archivo.\n";

return 1;

}

std::string nombre, apellido;

int edad;

double salario;

std::cout << "Registro de Empleados\n";

std::cout << "Ingresa los datos de un empleado:\n";

std::cout << "Nombre: ";

std::cin >> nombre;

std::cout << "Apellido: ";

std::cin >> apellido;

std::cout << "Edad: ";

std::cin >> edad;

std::cout << "Salario: ";

std::cin >> salario;

Empleado emp(nombre, apellido, edad, salario);

// Escribir datos en el archivo

archivo << emp.getNombre() << " " << emp.getApellido() << " " << emp.getEdad()

<< " " << emp.getSalario() << "\n";

std::cout << "Empleado registrado correctamente en empleados.txt.\n";

return 0;

}

Enlace a GitHub

Conclusión