RONDAN MEDINA JESUS SALVADOR 2024000228

Experiencia 01:

#include <iostream>

#include <string>

using namespace std;

class CuentaBancaria {

private:

string numero;

string titular;

double saldo;

public:

CuentaBancaria(string num, string tit, double sal) : numero(num), titular(tit), saldo(sal) {}

CuentaBancaria& depositar(double cantidad) {

saldo += cantidad;

return \*this;

}

CuentaBancaria& retirar(double cantidad) {

if (cantidad <= saldo) {

saldo -= cantidad;

} else {

cout << "Saldo insuficiente!" << endl;

}

return \*this;

}

void mostrarSaldo() {

cout << "Saldo actual: " << saldo << endl;

}

};

int main() {

CuentaBancaria cuenta("160506", "Salvador Rondan", 500.0);

cuenta.depositar(1000).retirar(285).mostrarSaldo();

return 0;

}

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Experiencia 02:

#include <iostream>

#include <string>

using namespace std;

class Dispositivo {

private:

string marca;

string modelo;

int anioFabricacion;

public:

Dispositivo() : marca("Desconocida"), modelo("Desconocido"), anioFabricacion(0) {

cout << "Constructor predeterminado para " << modelo << endl;

}

Dispositivo(string m, string mod, int anio) : marca(m), modelo(mod), anioFabricacion(anio) {

cout << "Constructor parametrizado para " << modelo << endl;

}

Dispositivo(const Dispositivo& otro) {

marca = otro.marca;

modelo = otro.modelo;

anioFabricacion = otro.anioFabricacion;

cout << "Constructor de copia para " << modelo << endl;

}

~Dispositivo() {

cout << "Destruyendo dispositivo: " << modelo << endl;

}

};

int main() {

Dispositivo dispositivos[3] = {

Dispositivo(),

Dispositivo("Samsung", "Galaxy S21", 2021),

Dispositivo("Apple", "iPhone 15", 2023)

};

return 0;

}

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Experiencia 03:

#include <iostream>

#include <string>

using namespace std;

class Fecha {

private:

int dia, mes, anio;

public:

Fecha(int d, int m, int a) : dia(d), mes(m), anio(a) {}

void mostrar() const {

cout << dia << "/" << mes << "/" << anio;

}

};

class Contrato {

private:

string codigo;

string descripcion;

Fecha fechaInicio;

public:

Contrato(string cod, string desc, int d, int m, int a)

: codigo(cod), descripcion(desc), fechaInicio(d, m, a) {}

void mostrarInfo() const {

cout << "Contrato " << codigo << ": " << descripcion << endl;

cout << "Fecha inicio: ";

fechaInicio.mostrar();

cout << endl;

}

};

class Empleado {

private:

string nombre;

Contrato contrato;

public:

Empleado(string nom, string codContrato, string descContrato, int d, int m, int a)

: nombre(nom), contrato(codContrato, descContrato, d, m, a) {}

void mostrarInfo() const {

cout << "Empleado: " << nombre << endl;

contrato.mostrarInfo();

}

};

int main() {

Empleado emp("Salvador Rondan", "CT001", "Contrato indefinido", 16, 5, 2006);

emp.mostrarInfo();

return 0;

}

Texto

El contenido generado por IA puede ser incorrecto.

Experiencia 04:

#include <iostream>

#include <string>

using namespace std;

class ComponenteA {

public:

ComponenteA() {

cout << "Constructor de ComponenteA" << endl;

}

~ComponenteA() {

cout << "Destructor de ComponenteA" << endl;

}

};

class ComponenteB {

public:

ComponenteB() {

cout << "Constructor de ComponenteB" << endl;

}

~ComponenteB() {

cout << "Destructor de ComponenteB" << endl;

}

};

class Sistema {

private:

ComponenteA compA;

ComponenteB compB;

public:

Sistema() : compA(), compB() {

cout << "Constructor de Sistema" << endl;

}

~Sistema() {

cout << "Destructor de Sistema" << endl;

}

};

class A {

public:

A() {

cout << "Constructor de A" << endl;

}

~A() {

cout << "Destructor de A" << endl;

}

};

class B {

private:

A objA;

public:

B() : objA() {

cout << "Constructor de B" << endl;

}

~B() {

cout << "Destructor de B" << endl;

}

};

int main() {

cout << "--- Creando Sistema ---" << endl;

Sistema sistema;

cout << "--- Fin de Sistema ---" << endl << endl;

cout << "--- Creando B ---" << endl;

B objB;

cout << "--- Fin de B ---" << endl;

return 0;

}

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Ejercicio 01:

#include <iostream>

#include <string>

using namespace std;

class Producto {

private:

string nombre;

double precio;

double descuento;

public:

Producto& setNombre(string n) {

nombre = n;

return \*this;

}

Producto& setPrecio(double p) {

precio = p;

return \*this;

}

Producto& setDescuento(double d) {

descuento = d;

return \*this;

}

double calcularPrecioFinal() {

return precio - (precio \* descuento / 100);

}

void mostrarInfo() {

cout << nombre << ": $" << precio << " (Descuento: " << descuento << "%) - Precio final: $" << calcularPrecioFinal() << endl;

}

};

int main() {

Producto productos[3];

productos[0].setNombre("Laptop").setPrecio(2500).setDescuento(10);

productos[1].setNombre("Mouse").setPrecio(50).setDescuento(5);

productos[2].setNombre("Teclado").setPrecio(120).setDescuento(15);

double total = 0;

for (int i = 0; i < 3; i++) {

productos[i].mostrarInfo();

total += productos[i].calcularPrecioFinal();

}

cout << "\nTotal acumulado: $" << total << endl;

return 0;

}

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Ejercicio 02:

#include <iostream>

#include <string>

using namespace std;

class Fecha {

private:

int dia, mes, anio;

public:

Fecha(int d, int m, int a) : dia(d), mes(m), anio(a) {}

void mostrarFecha() {

cout << dia << "/" << mes << "/" << anio;

}

};

class Direccion {

private:

string calle;

string ciudad;

string codigoPostal;

public:

Direccion(string c, string ciu, string cp) : calle(c), ciudad(ciu), codigoPostal(cp) {}

void mostrarDireccion() {

cout << calle << ", " << ciudad << " (" << codigoPostal << ")";

}

};

class Cliente {

private:

string nombre;

Fecha fechaRegistro;

Direccion direccion;

public:

Cliente(string nom, int d, int m, int a, string calle, string ciu, string cp)

: nombre(nom), fechaRegistro(d, m, a), direccion(calle, ciu, cp) {}

void mostrarInfo() {

cout << "Cliente: " << nombre << endl;

cout << "Fecha de registro: ";

fechaRegistro.mostrarFecha();

cout << endl << "Direccion: ";

direccion.mostrarDireccion();

cout << endl;

}

};

int main() {

Cliente cliente("Ana", 15, 5, 2020, "Av. Peru 69", "Arequipa", "14001");

cliente.mostrarInfo();

return 0;

}

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Ejercicio 03:

#include <iostream>

#include <string>

using namespace std;

class Habitacion {

private:

int numero;

string tipo;

double precioPorNoche;

public:

Habitacion(int num, string tip, double precio)

: numero(num), tipo(tip), precioPorNoche(precio) {}

double getPrecio() {

return precioPorNoche;

}

void mostrarInfo() {

cout << "Habitacion " << numero << " (" << tipo << "): $" << precioPorNoche << "/noche" << endl;

}

};

class ReservaHotel {

private:

Habitacion\*\* habitaciones;

int cantidad;

public:

ReservaHotel(int cant) : cantidad(cant) {

habitaciones = new Habitacion\*[cantidad];

for (int i = 0; i < cantidad; i++) {

string tipo = (i % 2 == 0) ? "Individual" : "Doble";

double precio = (i % 2 == 0) ? 80.0 : 120.0;

habitaciones[i] = new Habitacion(i+1, tipo, precio);

}

}

~ReservaHotel() {

for (int i = 0; i < cantidad; i++) {

delete habitaciones[i];

}

delete[] habitaciones;

cout << "Memoria liberada para " << cantidad << " habitaciones" << endl;

}

double calcularTotalPorNoche() {

double total = 0;

for (int i = 0; i < cantidad; i++) {

total += habitaciones[i]->getPrecio();

}

return total;

}

void mostrarReserva() {

cout << "Reserva con " << cantidad << " habitaciones:" << endl;

for (int i = 0; i < cantidad; i++) {

habitaciones[i]->mostrarInfo();

}

cout << "Total por noche: $" << calcularTotalPorNoche() << endl;

}

};

int main() {

ReservaHotel reserva(3);

reserva.mostrarReserva();

return 0;

}

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Ejercicio 04:

#include <iostream>

#include <string>

using namespace std;

class Gerente {

private:

string nombre;

int aniosExperiencia;

public:

Gerente(string nom, int anios) : nombre(nom), aniosExperiencia(anios) {

cout << "Constructor de Gerente para " << nombre << endl;

}

~Gerente() {

cout << "Destructor de Gerente para " << nombre << endl;

}

void mostrarInfo() {

cout << "Gerente: " << nombre << " (" << aniosExperiencia << " anios de experiencia)" << endl;

}

};

class Direccion {

private:

string calle;

string ciudad;

public:

Direccion(string c, string ciu) : calle(c), ciudad(ciu) {

cout << "Constructor de Direccion para " << ciudad << endl;

}

~Direccion() {

cout << "Destructor de Direccion para " << ciudad << endl;

}

void mostrarInfo() {

cout << "Direccion: " << calle << ", " << ciudad << endl;

}

};

class Empresa {

private:

string nombre;

Gerente gerente;

Direccion direccion;

public:

Empresa(string nom, string nomGerente, int exp, string calle, string ciu)

: nombre(nom), gerente(nomGerente, exp), direccion(calle, ciu) {

cout << "Constructor de Empresa para " << nombre << endl;

}

~Empresa() {

cout << "Destructor de Empresa para " << nombre << endl;

}

void mostrarInfo() {

cout << "\nEmpresa: " << nombre << endl;

gerente.mostrarInfo();

direccion.mostrarInfo();

}

};

int main() {

cout << "--- Creando Empresa ---" << endl;

Empresa empresa("Tilin Soluciones", "Salvador Rondan", 10, "Av. Ejercito 456", "Arequipa");

empresa.mostrarInfo();

cout << "--- Fin del main ---" << endl;

return 0;

}

Una captura de pantalla de una computadora

El contenido generado por IA puede ser incorrecto.