Actividad 1

// Experiencia de práctica N°1

#include <iostream>

#include <string>

using namespace std;

class CuentaBancaria

{

private:

int numero;

string titular;

float saldo;

public:

CuentaBancaria(int num, string tit, float sal = 0)

{

this->numero = num;

this->titular = tit;

this->saldo = sal;

}

CuentaBancaria& depositar(double monto)

{

this->saldo += monto;

return \*this;

}

CuentaBancaria& retirar(double monto)

{

if (monto <= this->saldo)

{

this->saldo -= monto;

}

else

{

cout << "Fondos insuficientes para retirar " << monto << " soles.\n";

}

return \*this;

}

CuentaBancaria& mostrarSaldo()

{

cout << "Saldo actual: " << this->saldo << " soles" << endl;

return \*this;

}

};

int main()

{

CuentaBancaria cuenta(123456, "Juan Perez", 500);

cuenta.depositar(1000).retirar(200).mostrarSaldo();

system("pause");

return 0;

}

Texto

El contenido generado por IA puede ser incorrecto.

Actividad 2

#include <iostream>

#include <string>

using namespace std;

class Dispositivo

{

private:

string marca;

string modelo;

int anioFabricacion;

public:

// Constructor predeterminado

Dispositivo()

{

marca = "Desconocido";

modelo = "Desconocido";

anioFabricacion = 0;

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

}

// Constructor parametrizado

Dispositivo(string m, string mod, int anio)

{

marca = m;

modelo = mod;

anioFabricacion = anio;

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

}

// Constructor de copia

Dispositivo(const Dispositivo& otro)

{

marca = otro.marca;

modelo = otro.modelo;

anioFabricacion = otro.anioFabricacion;

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

}

// Destructor

~Dispositivo()

{

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

}

void mostrar()

{

cout << "Dispositivo: " << modelo << endl;

}

};

int main()

{

const int TAM = 2;

Dispositivo dispositivos[TAM] =

{

Dispositivo("Xiomi", "POCO M5s", 2021),

Dispositivo("Samsung", "S23", 2023)

};

cout << "\nMostrando dispositivos:" << endl;

for (int i = 0; i < TAM; i++)

{

dispositivos[i].mostrar();

}

return 0;

}

Una captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Actividad 3

#include <iostream>

#include <string>

using namespace std;

// Clase Fecha

class Fecha

{

private:

int dia, mes, anio;

public:

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

{

cout << "Constructor de Fecha llamado.\n";

}

void mostrar()

{

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

}

};

class Contrato

{

private:

string codigo;

string descripcion;

Fecha fechaInicio;

public:

Contrato(string c, string d, int dia, int mes, int anio)

: codigo(c), descripcion(d), fechaInicio(dia, mes, anio)

{

cout << "Constructor de Contrato llamado.\n";

}

void mostrar()

{

cout << "Código: " << codigo << endl;

cout << "Descripción: " << descripcion << endl;

cout << "Fecha de inicio: ";

fechaInicio.mostrar();

}

};

class Empleado

{

private:

string nombre;

Contrato contrato;

public:

Empleado(string nom, string cod, string desc, int dia, int mes, int anio)

: nombre(nom), contrato(cod, desc, dia, mes, anio)

{

cout << "Constructor de Empleado llamado.\n";

}

void mostrar()

{

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

contrato.mostrar();

}

};

int main()

{

Empleado empleado1("Juan Perez", "C123", "Contrato Temporal", 10, 5, 2024);

cout << "\nMostrando datos del empleado:\n";

empleado1.mostrar();

return 0;

}

Texto

El contenido generado por IA puede ser incorrecto.

Actividad 4

#include <iostream>

using namespace std;

class ComponenteA {

public:

ComponenteA() {

cout << "Constructor de ComponenteA\n";

}

~ComponenteA() {

cout << "Destructor de ComponenteA\n";

}

};

class ComponenteB {

public:

ComponenteB() {

cout << "Constructor de ComponenteB\n";

}

~ComponenteB() {

cout << "Destructor de ComponenteB\n";

}

};

class Sistema {

private:

ComponenteA a;

ComponenteB b;

public:

Sistema() {

cout << "Constructor de Sistema\n";

}

~Sistema() {

cout << "Destructor de Sistema\n";

}

};

// Clase A

class A {

public:

A() {

cout << "Constructor de A\n";

}

~A() {

cout << "Destructor de A\n";

}

};

// Clase B (contiene un objeto de tipo A)

class B {

private:

A objetoA;

public:

B() {

cout << "Constructor de B\n";

}

~B() {

cout << "Destructor de B\n";

}

};

int main() {

cout << "--- Creando Sistema ---\n";

Sistema sistema1;

cout << "\n--- Creando objeto B ---\n";

B objetoB;

cout << "\n--- Fin del programa ---\n";

return 0;

}

Texto

El contenido generado por IA puede ser incorrecto.

Practica 1

#include <iostream>

#include <string>

using namespace std;

class CuentaBancaria {

private:

string numero;

string titular;

double saldo;

public:

CuentaBancaria(string num, string tit) {

numero = num;

titular = tit;

saldo = 0.0;

}

CuentaBancaria& depositar(double monto) {

saldo += monto;

return \*this;

}

CuentaBancaria& retirar(double monto) {

if (monto <= saldo) {

saldo -= monto;

} else {

cout << "Saldo insuficiente.\n";

}

return \*this;

}

void mostrarSaldo() {

cout << "Saldo actual de la cuenta " << numero << ": " << saldo << endl;

}

};

int main() {

CuentaBancaria cuenta("12345", "Juan Perez");

cuenta.depositar(1000).retirar(200).mostrarSaldo();

return 0;

}

Texto

El contenido generado por IA puede ser incorrecto.

Practica 2

#include <iostream>

#include <string>

using namespace std;

class FechaRegistro {

int dia, mes, anio;

public:

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

void mostrar() const {

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

}

};

class Direccion {

string calle;

int numero;

public:

Direccion(string c, int n) : calle(c), numero(n) {}

void mostrar() const {

cout << calle << " #" << numero << endl;

}

};

class Cliente {

string nombre;

FechaRegistro fecha;

Direccion direccion;

public:

Cliente(string nom, int d, int m, int a, string c, int n)

: nombre(nom), fecha(d, m, a), direccion(c, n) {}

void mostrar() const {

cout << "Nombre: " << nombre << endl;

cout << "Direccion: ";

direccion.mostrar();

cout << "Fecha de Registro: ";

fecha.mostrar();

}

};

int main() {

Cliente cliente("Juan Perez", 12, 5, 2025, "Av. Principal", 123);

cliente.mostrar();

return 0;

}

Texto

El contenido generado por IA puede ser incorrecto.

Practica 3

#include <iostream>

#include <string>

using namespace std;

class Habitacion {

int numero;

string tipo;

double precioPorNoche;

public:

Habitacion(int num, string t, double precio)

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

double getPrecio() const {

return precioPorNoche;

}

void mostrar() const {

cout << "Habitacion " << numero << " (" << tipo << ") - " << precioPorNoche << " por noche" << endl;

}

};

class ReservaHotel {

Habitacion\* habitaciones;

int cantidad;

public:

ReservaHotel(int cant) : cantidad(cant) {

habitaciones = new Habitacion[cantidad] {

{101, "Simple", 100.0},

{102, "Doble", 150.0},

{103, "Suite", 300.0}

};

}

~ReservaHotel() {

delete[] habitaciones;

}

double calcularTotalPorNoche() const {

double total = 0;

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

total += habitaciones[i].getPrecio();

}

return total;

}

void mostrarHabitaciones() const {

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

habitaciones[i].mostrar();

}

}

};

int main() {

ReservaHotel reserva(3);

reserva.mostrarHabitaciones();

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

return 0;

}

Texto

El contenido generado por IA puede ser incorrecto.

Practica 4

#include <iostream>

#include <string>

using namespace std;

class Gerente {

string nombre;

public:

Gerente(string n) : nombre(n) {

cout << "Constructor Gerente: " << nombre << endl;

}

~Gerente() {

cout << "Destructor Gerente: " << nombre << endl;

}

};

class Direccion {

string ciudad;

public:

Direccion(string c) : ciudad(c) {

cout << "Constructor Direccion: " << ciudad << endl;

}

~Direccion() {

cout << "Destructor Direccion: " << ciudad << endl;

}

};

class Empresa {

Gerente gerente;

Direccion direccion;

public:

Empresa(string nomGerente, string ciu)

: gerente(nomGerente), direccion(ciu) {

cout << "Constructor Empresa" << endl;

}

~Empresa() {

cout << "Destructor Empresa" << endl;

}

};

int main() {

Empresa empresa("Carlos Lopez", "Arequipa");

return 0;

}

Una captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.