**I. ¿Compila exitosamente el siguiente programa? ¿Por qué?**

Sí, el programa compila exitosamente porque se usa bien la herencia y la inicialización del constructor de la clase base. La clase derivada llama al constructor de la clase base.

¿Qué resultados se obtienen luego de su ejecución?

La ejecución del programa imprime:

unaVar = 50

**2. Implementar una clase para cargar un vector de 5 elementos:**

#include <iostream>

using namespace std;

class Vector {

private:

int elementos[5];

public:

Vector() {

cargarDatos();

}

void cargarDatos() {

cout << "Ingrese 5 elementos:\n";

for (int i = 0; i < 5; i++) {

cout << "Elemento " << (i + 1) << ": ";

cin >> elementos[i];

}

}

void imprimir() {

cout << "Elementos del vector:";

for (int i = 0; i < 5; i++) {

cout << " " << elementos[i];

}

cout << endl;

}

void imprimir(int hasta) {

for (int i = 0; i < hasta && i < 5; i++) {

cout << elementos[i] << " ";

}

cout << endl;

}

void imprimir(int desde, int hasta) {

for (int i = desde; i < hasta && i < 5; i++) {

cout << elementos[i] << " ";

}

cout << endl;

}

};

int main() {

Vector v;

v.imprimir();

v.imprimir(3);

v.imprimir(1, 4);

return 0;

}

**3. Clase Punto con dos constructores:**

#include <iostream>

using namespace std;

class Punto {

private:

int x, y;

public:

Punto() : x(0), y(0) {}

Punto(int x, int y) : x(x), y(y) {}

void imprimir() {

cout << "Punto: (" << x << ", " << y << ")" << endl;

}

};

int main() {

p1.imprimir();

p2.imprimir();

return 0;

}

**4. Clase Club y Socio:**

#include <iostream>

#include <string>

using namespace std;

class Socio {

private:

string nombre;

int antiguedad;

public:

Socio() {

cout << "Ingrese nombre del socio: ";

cin >> nombre;

cout << "Ingrese antigüedad en años: ";

cin >> antiguedad;

}

int obtenerAntiguedad() const {

return antiguedad;

}

string obtenerNombre() const {

return nombre;

}

};

class Club {

private:

Socio socios[3];

public:

void imprimirMayorAntiguedad() {

int mayorAntiguedad = 0;

string nombreMayor;

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

if (socios[i].obtenerAntiguedad() > mayorAntiguedad) {

mayorAntiguedad = socios[i].obtenerAntiguedad();

nombreMayor = socios[i].obtenerNombre();

}

}

cout << "El socio con mayor antigüedad es: " << nombreMayor << " con " << mayorAntiguedad << " años." << endl;

}

};

int main() {

Club club;

club.imprimirMayorAntiguedad();

return 0;

}

**5. Clase `Persona` y `Empleado`:**

#include <iostream>

#include <string>

using namespace std;

class Persona {

private:

string nombre;

int edad;

public:

void cargarDatos() {

cout << "Ingrese nombre: ";

cin >> nombre;

cout << "Ingrese edad: ";

cin >> edad;

}

void imprimirDatos() {

cout << "Nombre: " << nombre << ", Edad: " << edad << endl;

}

};

class Empleado : public Persona {

private:

float sueldo;

public:

void cargarSueldo() {

cout << "Ingrese sueldo: ";

cin >> sueldo;

}

void imprimirSueldo() {

cout << "Sueldo: " << sueldo << endl;

}

};

int main() {

Persona p;

p.cargarDatos();

p.imprimirDatos();

Empleado e;

e.cargarDatos();

e.cargarSueldo();

e.imprimirDatos();

e.imprimirSueldo();

return 0;

}