

1-i-

Este código se ejecuta y funciona correctamente, porque al compilarlo no te aparece ningún error y te genera el programa normalmente.

1-ii-

Luego de que el programa se ejecutara correctamente este muestra una variable cuyo en el código se ve que empieza en 0 y luego lo rellena en este caso con el número 50.

```
#include <iostream>
#include <string>
using namespace std;

class ClaseBase {
protected:
    int unaVar = 0;

public:
    ClaseBase(int x) : unaVar(x) {}

    void unMetodo(void) {
        cout << "unaVar = " << unaVar << endl;
    }
};

class ClaseDerivada : public ClaseBase {
public:
    ClaseDerivada(int x) : ClaseBase(x) {}
};

class Vector {
private:
    int v[5];

public:
    Vector() {
        for (int i = 0; i < 5; i++) {
            cout << "Ingrese el valor para el elemento " << i + 1 << ": ";
            cin >> v[i];
        }
    }

    void imprimir() {
        cout << "El vector completo es: ";
        for (int i = 0; i < 5; i++) {
            cout << v[i] << " ";
        }
        cout << endl;
    }
}
```

```

void imprimir(int hasta) {
    cout << "Elementos hasta el índice " << hasta << ": ";
    for (int i = 0; i <= hasta && i < 5; i++) {
        cout << v[i] << " ";
    }
    cout << endl;
}

```

```

void imprimir(int desde, int hasta) {
    cout << "Elementos desde el índice " << desde << " hasta el índice " << hasta << ": ";
    for (int i = desde; i <= hasta && i < 5; i++) {
        cout << v[i] << " ";
    }
    cout << endl;
}
};

```

```

class Punto {
private:
    int x, y;

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

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

    void imprimir() {
        cout << "Coordenadas del punto: (" << x << ", " << y << ")" << endl;
    }
};

```

```

class Socio {
private:
    string nombre;
    int antigüedad;

public:
    Socio() {
        cout << "Ingrese el nombre del socio: ";
        cin >> nombre;
        cout << "Ingrese la antigüedad del socio (en años): ";
        cin >> antigüedad;
    }

    int obtenerAntigüedad() const { return antigüedad; }
    string obtenerNombre() const { return nombre; }
};

```

```

class Club {
private:
    Socio socios[3];

public:
    void mostrarSocioMasAntiguo() {
        int maxAntigüedad = 0;
        string nombreMasAntiguo;

        for (int i = 0; i < 3; i++) {
            if (socios[i].obtenerAntigüedad() > maxAntigüedad) {
                maxAntigüedad = socios[i].obtenerAntigüedad();
                nombreMasAntiguo = socios[i].obtenerNombre();
            }
        }

        cout << "El socio con mayor antigüedad es: " << nombreMasAntiguo << endl;
    }
};

```

```

class Persona {
private:
    string nombre;
    int edad;

public:
    void cargarDatos() {
        cout << "Ingrese su nombre: ";
        cin >> nombre;
        cout << "Ingrese su edad: ";
        cin >> edad;
    }

    void imprimirDatos() {
        cout << "Nombre: " << nombre << ", Edad: " << edad << endl;
    }
};

```

```

class Empleado : public Persona {
private:
    double sueldo;

public:
    void cargarSueldo() {
        cout << "Ingrese su sueldo: ";
        cin >> sueldo;
    }
}

```

```

void imprimirSueldo() {
    cout << "Sueldo: " << sueldo << endl;
}
};

int main() {
    int opcion;
    do {
        // Menú de opciones
        cout << "\nSelecciona una opcion:" << endl;
        cout << "1. Crear un objeto de ClaseDerivada y mostrar unaVar" << endl;
        cout << "2. Crear un vector de 5 elementos y mostrarlo" << endl;
        cout << "3. Crear un punto e imprimir sus coordenadas" << endl;
        cout << "4. Crear un Club y mostrar el socio con mayor antigüedad" << endl;
        cout << "5. Crear un objeto Persona y un objeto Empleado y mostrar sus datos" <<
endl;
        cout << "0. Salir" << endl;
        cout << "Opcion: ";
        cin >> opcion;

        switch (opcion) {
            case 1: {
                int x;
                cout << "Ingrese un valor para unaVar: ";
                cin >> x;
                ClaseDerivada obj1(x);
                obj1.unMetodo();
                break;
            }
            case 2: {
                Vector vec;
                vec.imprimir();
                int hasta;
                cout << "Ingrese el indice hasta donde imprimir (0-4): ";
                cin >> hasta;
                vec.imprimir(hasta);
                int desde, hastaRango;
                cout << "Ingrese el indice desde: ";
                cin >> desde;
                cout << "Ingrese el indice hasta: ";
                cin >> hastaRango;
                vec.imprimir(desde, hastaRango);
                break;
            }
            case 3: {
                int x, y;
                cout << "Ingrese las coordenadas del punto (x y): ";

```

```

        cin >> x >> y;
        Punto p(x, y);
        p.imprimir();
        break;
    }
    case 4: {
        Club club;
        club.mostrarSocioMasAntiguo();
        break;
    }
    case 5: {
        Persona p;
        p.cargarDatos();
        p.imprimirDatos();

        Empleado e;
        e.cargarDatos();
        e.cargarSueldo();
        e.imprimirDatos();
        e.imprimirSueldo();
        break;
    }
    case 0:
        cout << "Saliendo..." << endl;
        break;
    default:
        cout << "Opción no válida. Intente de nuevo." << endl;
    }
} while (opcion != 0);

return 0;
}

```