PISTAS.

Aquí están cinco pistas progresivas para ayudar a crear una plantilla de clase en C++ con una función miembro que esté definida fuera de la clase:

1. ¿Cómo se define una plantilla de clase en C++?
2. ¿Cómo se define una función miembro dentro de una clase en C++?
3. ¿Cómo se define una plantilla de función fuera de una clase en C++?
4. ¿Cómo se accede a una variable de instancia desde una función miembro?
5. ¿Cómo se llama a una función miembro definida como plantilla de función fuera de una clase en C++?

Aquí está el código completo con las pistas incluidas:

#include <iostream>

// Pista 1: define una plantilla de clase ClassTemplate que tome un argumento de tipo T

template <typename T>

class ClassTemplate {

public:

// Pista 4: define un constructor que tome un argumento value y lo asigne a m\_value

ClassTemplate(T value) : m\_value(value) {}

// Pista 2: define una función miembro llamada getValue

T getValue() const;

private:

T m\_value;

};

// Pista 3: define la definición completa de getValue como una plantilla de función

template <typename T>

T ClassTemplate<T>::getValue() const {

// Pista 4: devuelve m\_value

return m\_value;

}

int main() {

// Pista 5: llama a getValue en un objeto de ClassTemplate<int>

ClassTemplate<int> obj(5);

std::cout << obj.getValue() << std::endl;

return 0;

}