In short, class templates that do not take specifically *types* as their template parameters cannot be used in instantiations of `C4`.

In the general case, template template parameter packs may appear together with other template parameters and follow the rules and restrictions discussed so far in the context of type parameter packs. ~~Any number of subtle but nevertheless perfectly meaningful matching cases can be defined involving combinations of fixed and variadic template parameters.~~ Suppose, for example, we want to define a class template `C5` that accepts a template that takes *at least* two parameters and possibly more:

```cpp
template <template<typename, typename, typename...> class X>
class C5
    // class template definition having one template template parameter
    // for which the template template accepts two or more type
    // arguments
{ /*...*/ };

// a few templates that match C5

template <typename, typename> class B1;
template <typename, typename, typename = int> class B2;
template <typename, typename = int, typename = int> class B3;
template <typename, typename, typename...> class B4;

C5<B1> c5a;  // OK
C5<B2> c5b;  // OK
C5<B3> c5c;  // OK
C5<B4> c5d;  // OK
```

However, templates that don't have two fixed type parameters in the first two positions will not match `C5`:

```cpp
template <typename> class B5;
template <int, typename, typename...> class B6;
template <typename, typename, int, typename...> class B7;
template <typename, typename...> class B8;

C5<B5> c5d;  // Error, argument mismatch
C5<B6> c5e;  // Error, argument mismatch
C5<B7> c5f;  // Error, argument mismatch
C5<B8> c5g;  // Error, argument mismatch
```

To match `C5`'s template parameter, a template must take types for its first two parameters, followed by zero or more type parameters (with default values or not). `B5` does not match because it takes only one parameter. `B6` does not match because it takes an `int` in the first position, as opposed to a type, as required. `B7` fails to match because it takes an `int` in the third position instead of a type. Finally, `B8` is not a match because its second argument is not fixed.