- Example

In this example, we will look at the usage of static_assert.

WE'LL COVER THE FOLLOWING ^

- Example
  - Explanation

Example

```
 // staticAssert.cpp
#include <iostream>
#include <type_traits>

template< class T >
struct Add{

  // check the assertion
  static_assert(std::is_arithmetic<T>::value, "Argument T must be an arithmetic type");

} ;

int main(){

  // will work
  static_assert(sizeof(void*) >= 8, "64-bit addressing is required for this program");

  // int is arithmetic
  Add<int> addInt= Add<int>();

  // double is arithmetic
  Add<double> addDouble= Add<double>();

  // char is arithmetic
  Add<char> addChar= Add<char>();

  // std::string is not arithmetic
  Add<std::string> addString= Add<std::string>(); // if you comment this line, the code will

}
```

Explanation

The program uses `static_assert` in the class scope (line 9) and the local scope (line 16). The assertions in the class definition guarantee that the structure is initialized with an arithmetic type, explaining why the template instantiation in line 28 is not valid.

Let’s test your understanding of this concept with an exercise in the next lesson.