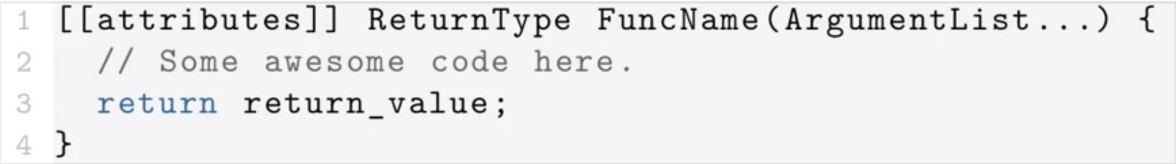
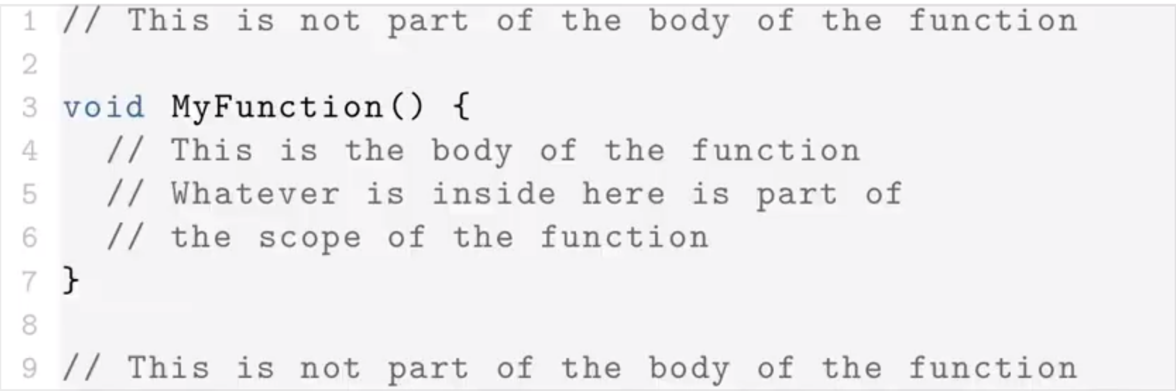
C++ Functions

Function Anatomy:



```cpp
[[attributes]] ReturnType FuncName(ArgumentList...) {
  // Some awesome code here.
  return return_value;
}
```

Function Body:

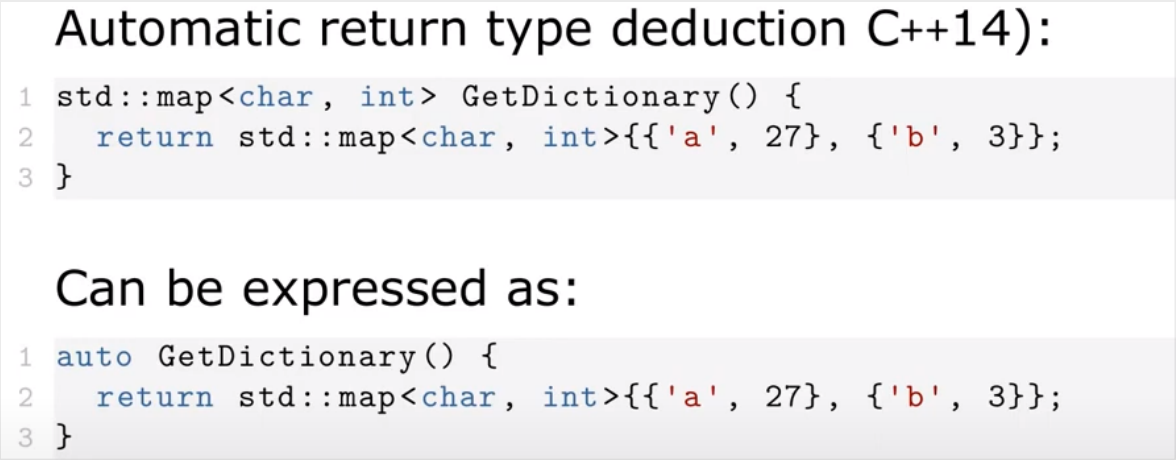


```cpp
// This is not part of the body of the function

void MyFunction() {
  // This is the body of the function
  // Whatever is inside here is part of
  // the scope of the function
}

// This is not part of the body of the function
```

Return Type:



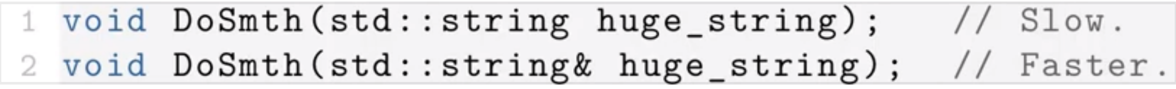
Automatic return type deduction C++14):

```cpp
std::map<char, int> GetDictionary() {
  return std::map<char, int>{{'a', 27}, {'b', 3}};
}
```

Can be expressed as:

```cpp
auto GetDictionary() {
  return std::map<char, int>{{'a', 27}, {'b', 3}};
}
```

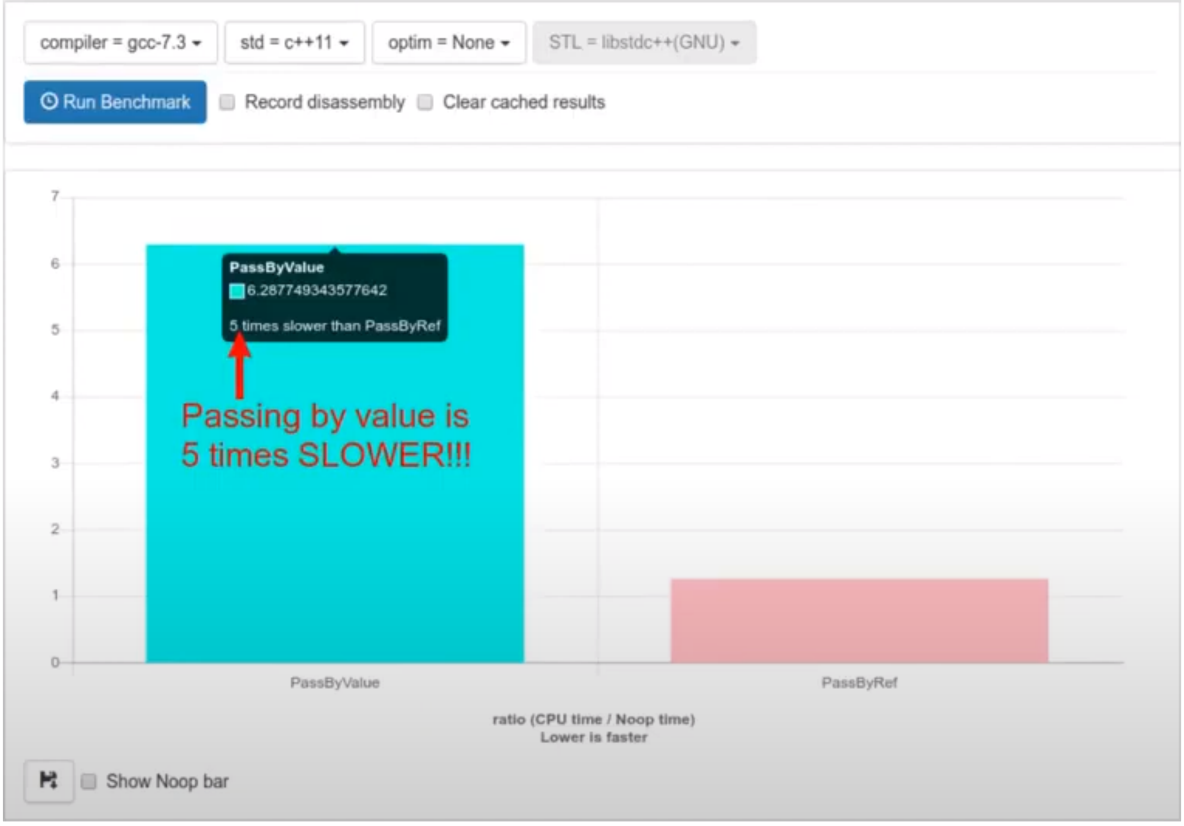
Passing big objects:



```cpp
void DoSmth(std::string huge_string);   // Slow.
void DoSmth(std::string& huge_string);  // Faster.
```

Passing const reference is best if we don't want out argument to be changed.

Cost of passing by Value:



compiler = gcc-7.3

std = c++11

optim = None

STL = libstdc++(GNU)

PassByValue

6.287749343577642

5 times slower than PassByRef

Passing by value is 5 times SLOWER!!!

Inline functions:

- function calls are expensive
- IF a function is rather small, you could help the compiler.
- **inline** is a hint to the compiler
  - Should attempt to generate code for a call
  - Rather than a function call
- Sometimes the compiler do it anyways