ECE2036: Week 12 – Polymorphic Global Function

Instructions: Find the output of the following code

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

class Base
{
 public:
  virtual void Func1() = 0;
  void Func2();
  virtual void Func3();
  void Func4(); };
class Sub1 : public Base
{public:
    virtual void Func1();
    void Func2(); };
class Sub2 : public Sub1 {
public:
  virtual void Func1();
  void Func2();
  virtual void Func3(); };
class Sub3 : public Sub2
{public:
  virtual void Func1();
  void Func2(); };

  void Base::Func2()
  {  std::cout << "Hello from Base::Func2()" << std::endl; };
  void Base::Func3()
  {  std::cout << "Hello from Base::Func3()" << std::endl; };
  void Base::Func4()
  {  std::cout << "Hello from Base::Func4()" << std::endl;
  Func1();
  Func2(); };
  void Sub1::Func1()
  { std::cout << "Hello from Sub1::Func1()" << std::endl; };
  void Sub1::Func2()
  { std::cout << "Hello from Sub1::Func2()" << std::endl; };
  void Sub2::Func1()
  { std::cout << "Hello from Sub2::Func1()" << std::endl; };
  void Sub2::Func2()
  { std::cout << "Hello from Sub2::Func2()" << std::endl;  };
  void Sub2::Func3()
  { std::cout << "Hello from Sub2::Func3()" << std::endl; };
  void Sub3::Func1()
  { std::cout << "Hello from Sub3::Func1()" << std::endl; };
  void Sub3::Func2()
  { std::cout << "Hello from Sub3::Func2()" << std::endl; };

  void globalFunction(Base * b)
  {
    b->Func1();
    b->Func2();
    b->Func3();
    b->Func4();
  }

int main()
{
 Sub2 s2;
 Sub3 s3;

 globalFunction(&s2);
 cout << endl;
 globalFunction(&s3);

}
```

Example 1: Solution

```
Hello from Sub2::Func1()
Hello from Base::Func2()
Hello from Sub2::Func3()
Hello from Base::Func4()
Hello from Sub2::Func1()
Hello from Base::Func2()

Hello from Sub3::Func1()
Hello from Base::Func2()
Hello from Sub2::Func3()
Hello from Base::Func4()
Hello from Sub3::Func1()
Hello from Base::Func2()
```