# Object-Oriented Programming in C++

# Lab Exercise 5 (2%)

## Objective

The objectives of this exercise are to deepen our understanding of C++ and learn how to avoid common pitfalls. You can work on the lab in a group of two people.

## Description

Show the result of each of the below items to the instructor as you finish them. The following program is given:

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

namespace a {
   class Base {
   public:
      Base( std::string str ) : str_(str) {}
      void out() { cout << "Base: " << str_ << '\n';}
   protected:
      std::string str_;
   };
   class Derived : public Base { public:
      Derived( std::string str ) : Base( str ) {}
   };
}

using namespace a;

int main() {
   Base b("A");
   Derived d("B");
   Derived *pdd = new Derived("C");
   Base *pdb = pdd;

   // Part a) -f)
   b.out();
   d.out();
   pdd->out();
   pdb->out();

   // Part g)
   // … to be added …

   delete pdd;
}
```

a) Run the program. What is the output?

b) Provide an implementation of the out() method in the derived class (the implementation should behave the same except it prints out the prefix "Derived" instead of "Base"). Run the program. What is the output?

c) Make the out() method in the base class virtual. Run the program. What is the output? Now also explicitly write virtual before the out method in the derived class (is that needed?) What is the difference?

Above, in part b), we are *redefining* (non-virtual function), but here in part c) *overriding* (virtual function). One of these is a bad practice to provide derived objects with a new behavior and should be avoided. Which and why?

d) Add an integer argument to out(), that you print out at the end (both base and derived). Make the argument in the base and derived class get a default value of 5 if omitted. Compile and run the program. What is the output? Now, omit the default value in the derived class. What happens? Why?

e) Experimenting with providing a different default value, say 7, in the derived class than the base class. Run the program. What is the output? Why?

f) Implement the out() method using the NVI idiom (and a default value of 5 for all classes).

g) Remove the *int* argument from out. Now, instead of using a public out method, implement the same functionality by overloading the << operator. Change the main function accordingly.