**LAB 4**

**Problem 1**

The test.cpp file contains a function called func() that takes in a pointer argument, modifies it, then returns the address of the pointer as reference. In the main function we set x to be 100. Before the function is called, x is 100. Once the function is called, x is referenced and is changed to be 200 and then returned. We then get an output of 200.

Compile: g++ UseHandle.cpp -o test

Run: ./test

NO INPUT

**Problem 2**

In the Test.cpp file, there is a modify function that will take in a reference of a pointer to a pointer. In order to have this work, the main function has x equal 100, a pointer to that x to be 100, and a pointer to the other pointer to also be 100. When printing the pointer to pointer, we get 100. Sending that pointer to pointer in the modify function as reference, we can set it to the value of 10. Now when printing the pointer to pointer, it’s 10.

Compile: g++ test.cpp -o test

Run: ./test

NO INPUT

**Problem 3**

In the test.cpp file, there is a base class called Subject, which contains 3 functions, f, g, h. The Proxy, Implementation1, and Implementation2 classes inherits the Subject class. Proxy class has a pointer to a subject, and the member functions use the pointer to call the functions in Subject. There is also a changeImplementation contains a function that takes in a implementation and repoints it to that implementation. The two implementation classes simply hold f, g, h functions which tell the user they have been called to make it obvious when an implementation is changed. In the main function, 2 implementations are created. The first 3 functions are called and then the implementation is changed to be the other and those 3 functions are then called.

Compile: g++ test.cpp -o test

Run: ./test

NO INPUT