Name (First Last): Victor Ejiasi

Lab 7 (LAST lab)

Q1. Fix the compilation errors and find the output of following program (25%)

```
#include
class Test {
private:
int x;
public:
Test(int x = 0) { this->x = x; }
void change(Test *t) { this = t; }
void print() { cout << "x = " << x << endl; }
};

 void main() {
Test obj(5);
Test *ptr = new Test (10);
obj.change(ptr);
obj.print();
}
```

"this" is a const pointer so there is an error in the statement "this=t;"

The highlighted statement can be replaced with "{t->x = 10;}"

Q1.cpp

The output of the program is x=5.

Q2. Define a class named 'Bank Account' to represent following members: (25%)

Data members :-

- Account Number
- Name of Depositor
- Account Type
- Balance Amount

Member functions:

- Initialize members
- Deposit Amount
- Withdraw Amount
- Display Balance

Q2.cpp

Write a C++ program to test the Bank Account class for 10 customers, and define a static member to keep track of how many objects are created. Print the number of account objects created at the end of the main function. Note: you can name the member variables and functions as you wish as long as the name is descriptive and follows best practices.

Q3. Write a class called `Adder` that stores the sum of all the `ints` given to it. Your `Adder` class should allow you to write the following code (and code like it): (25%)

```
// sample code
Adder sum1; // sum1 is initialized to 0
Adder sum2(2); // sum2 is initialized to 2
cout << "sum1 is " << sum1 << '\n'; // prints "sum1 is 0"
```

```
cout << "sum2 is " << sum2 << '\n'; // prints "sum2 is 2"
sum1 += 5; // adds 5 to sum1; now sum1 is 5
sum2 += -3; // adds -3 to sum2; now sum2 is -1
if (sum1 == sum2)
cout << "sum1 and sum2 are the same\n";
```

Q3.cpp

You should only write the functions that are necessary for `Adder` to be used as in the above program. Use `const` wherever appropriate, and do *not* write or use a cast operator. Make sure to include any necessary header files.

Q4. What will be the output of the following C++ code? What would be the output of the program, fully explain your answer. (25%)

```
#include <iostream>
#include <string>
using namespace std;
class X
{
        float d;
   public:
        virtual void function1(){
                cout<<"This is class X"<<endl;
        }
};

class Z: public X
{
        int x = 15;
   public:
        void function1(){
                cout<<"This is class Z"<<endl;
        }
};

int main()
{
        X *x = new X();
        Z z;
        x = &z;
        x->function1();
        return 0;
}
```

The output of this C++ code would be "This is class Z" because class Z inherits class X and overrides function1.