Report of Assignment 6

------Polymorphism

Meng Wei

Results:

In my code, I give examples SinFunction and MyFunction (x-1) to test my methods and codes. When we set x0=-0.5, and maximum error eps = 1.0e-4, the solutions of these two functions are 1.2108e-14 and 1.0. This means my codes are correct.

PS: Codes as follows:

```
// polymorphism.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include "stdafx.h"
#include<iostream>
#include<math.h>
using namespace std;

//define base class RealFunction
class RealFunction
{
public:
    RealFunction(){}
    ~RealFunction(){}
public:
    virtual double f(double) = 0;
    virtual double fDerivative(double) = 0;

    //using Newton method to get root
    double getRoot()
    {
        double x0 = -0.5; //initial value
        double eps = 1.0e-4; //maximal error
        double x;
        while (1)
        {
            x = x0 - f(x0) / fDerivative(x0);
            if (fabs(x - x0) < eps)
            {
                return x;
            }
            else
            {
                x0 = x;
            }
        }
    }
};

//define a derived class from RealFunction
class SinFunction : public RealFunction
{
public:
    SinFunction(){}
    ~SinFunction() {}
public:

    double f(double x)
    {
        double y;
        y = sin(x);
        return y;
    }
    double fDerivative(double x)
    {
        double z;
        z = cos(x);
        return z;
    }


};

//define a derived class from RealFunction
class MyFunction : public RealFunction
{
public:
    MyFunction(){}
    ~MyFunction() {}
public:
    double f(double x)
    {
        double y;
        y = x-1;
        return y;
    }
    double fDerivative(double x)
    {
        double z;
        z = 1;
        return z;
    }

};

int main()
{
    SinFunction test1;
    cout << test1.getRoot() << endl;
    MyFunction test2;
    cout << test2.getRoot() << endl;
    return 0;
}
```