Lab 7 Accompanying Document

Harry Haisty

September 2018

The Code

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

template <typename GenericType>
GenericType maxValue(const GenericType& value1, const GenericType& value2)
{
    if(value1 > value2)
        return value1;
    else
        return value2;
}

double maxValue(const double& value1, const double& value2)
{
    if (value1 > value2)
        return value1;
    else
        return value2;
}

int maxValue(const int& value1, const int& value2){
    if (value1 > value2)
        return value1;
    else
        return value2;
}

int main() {
    int i = 5, j = 6, k, z, y;
    long l = 10, m = 5, n;
    k = maxValue<double>(i, j);
    z = maxValue(i, j);
    y = maxValue<int>(i, j);
    cout << k << endl;
    cout << z << endl;
    cout << y << endl;
}
```

The Explanation

The object of this assignment was to explore different ways for a programmer to implement and replace overload functions. Using a template of a generic type, I was able to limit my code to only a few lines where before there had to be different methods to account for each different data type. The generic type template was able to take in a doubl, and int, or another value, and compare those two values taken in to see which one of the values was larger.

The Output