

# Programming Design, Spring 2016

## Homework 11 Solution

Solution provider: Chien-Yu Huang  
Department of Information Management  
National Taiwan University

### Problem 1

(a) Using `const` after a function declaration means that the function is not allowed to change any class members. If there are both a constant and a non-constant version: A constant function is invoked by a constant object. A non-constant function is invoked by a non-constant object.

If there is only a non-constant instance function: A constant object cannot invoke it.

(b) In this case a constructor and a destruction would be invoked every time we returned the value because we would need to create a new copy of the object when we returned the value. In contrast, returning by reference requires much fewer resources and the value of an object is simply copied to another object without invoking constructors and destructors.

(c) As we declare a local variable `sum`, the life span of which is only limited to the scope of the function. That is to say, when the function finishes, this local variable and the value stored in its memory address will both be deleted. Therefore, if we returned by reference, the program would have nothing to refer to after the function terminated. And thus would go wrong.

(d)

```
const MyVector MyVector::operator- (const MyVector& v) {
    MyVector difference(*this);
    for (int i = 0; i < n; i++)
        difference.m[i] -= v.m[i];
    return difference
}
```

### Problem 2

Please see the attached CPP file.

### Problem 3

Please see the attached CPP file.