Course: Programming Fundamentals – **ENCM 339**
Lab #: Lab 8
Instructor: S. Norman
Student Name: **Mitchell Sawatzky**
Lab Section: **B02**
Date Submitted: **Nov 23, 2015**

## Exercise D

```
LabString& LabString::insert(size_t pos, const char *s)
{
    int i = 0, j, k;

    while (s[i] != '\0')
        i++;
    if (i == 0)
        return *this;
    char* new_storage = new char[i + lengthM + 1];
    for (j = 0, k = 0; j <= lengthM; j++) {
        for (; j == pos && k < i; k++) {
            new_storage[j + k] = s[k];
        }
        if (storageM)
            new_storage[j + k] = storageM[j];
    }
    delete [ ] storageM;
    storageM = new_storage;
    lengthM += i;
    storageM[lengthM] = '\0';
    return *this;
}
```

## Exercise E

```
LabVector::LabVector(const LabVector& src)
{
    size_t i, count = src.capacity();
    storeM = new ElType[count];
    end_storeM = storeM + count;
    for (i = 0; i < src.size(); i++)
        storeM[i] = src.storeM[i];
    end_validM = storeM + i;
}

LabVector& LabVector::operator=(const LabVector& rhs)
{
    if (this == &rhs)
        return *this;
    delete [ ] storeM;
    size_t i, count = rhs.end_storeM - rhs.storeM;
```

```
    storeM = new ElType[count];
    end_storeM = storeM + count;
    for (i = 0; i < rhs.end_validM - rhs.storeM; i++)
        storeM[i] = rhs.storeM[i];
    end_validM = &storeM[i];
    return *this;
}

void LabVector::resize(size_t new_size, const ElType& extra_val)
{
    if (new_size == 0)
        storeM = end_validM = end_storeM = 0;
    else {
        ElType* new_store = new ElType[new_size];
        size_t count = size(), i;
        for (i = 0; i < new_size; i++) {
            new_store[i] = (i < count ? storeM[i] : extra_val);
        }
        end_storeM = end_validM = new_store + i;
        delete [ ] storeM;
        storeM = new_store;
    }
}
```

```
Mitchell@ttys000 22:41 {0} [lab8]$ ./test.out
default constructor ...
  EXPECT:
  ACTUAL:
push_back ...
  EXPECT: -40 -30 -20 -10 0 10 20 30 40
  ACTUAL: -40 -30 -20 -10 0 10 20 30 40
initialization with 4 copies of 65 ...
  EXPECT: 65 65 65 65
  ACTUAL: 65 65 65 65
initialization from built-in array ...
  EXPECT: 0 1 4 9 16
  ACTUAL: 0 1 4 9 16
copy constructor source, after updates ...
  EXPECT: -88 -30 -20 -10 0 10 20 30 40 -99
  ACTUAL: -88 -30 -20 -10 0 10 20 30 40 -99
copy constructor destination ...
  EXPECT: -40 -30 -20 -10 0 10 20 30 40
  ACTUAL: -40 -30 -20 -10 0 10 20 30 40
copy assignment operator, v4 after update ...
```

```
  EXPECT: -77 -30 -20 -10 0 10 20 30 40
  ACTUAL: -77 -30 -20 -10 0 10 20 30 40
copy assignment operator, v5 result ...
  EXPECT: -40 -30 -20 -10 0 10 20 30 40
  ACTUAL: -40 -30 -20 -10 0 10 20 30 40
copy assignment operator, v6 result ...
  EXPECT: -40 -30 -20 -10 0 10 20 30 40
  ACTUAL: -40 -30 -20 -10 0 10 20 30 40
resize ...
  EXPECT: -88 -30 -20 -10
  ACTUAL: -88 -30 -20 -10
  EXPECT: -88 -30 -20 -10 0 0 0 0 0
  ACTUAL: -88 -30 -20 -10 0 0 0 0 0
  EXPECT: -88 -30 -20 -10 0 0 0 0 0 42 42
  ACTUAL: -88 -30 -20 -10 0 0 0 0 0 42 42
  EXPECT:
  ACTUAL:
  EXPECT: 99 99 99 99 99
  ACTUAL: 99 99 99 99 99
  EXPECT: 99 99 99 99 99 12 12 12 12 12 12
  ACTUAL: 99 99 99 99 99 12 12 12 12 12 12
```