

# CS2134 Homework Assignment 3B

## Spring 2016

Due 11:00 p.m. Mon. Feb 22, 2016

February 20, 2016

Assignment 3B include a programming portion and a written part. The programming portion must compile and consist of a single file ( hw03B.cpp). The written portion should consist of a single file (hw03Bwritten) in a .pdf format. Be sure to include your name at the beginning of each file! You must hand in the file via NYU Classes.

### Programming Part:

**Enter data from the file MTA\_train\_stop\_data.txt.** The data from this assignment is from <http://www.mta.info/developers/download.html>. (Please note that we will only be using some of the information in this file for this assignment.<sup>1</sup>)

In the batch phase you will read all the data from the file called `MTA_train_stop_data.txt` into a container of type `vector<trainStopData>`.

Your program will define the class `trainStopData`. It has the following private member variables :

```
string stop_id;    // id of train stop (1st token)
string stop_name;  // name of station (4th token)
double stop_lat;   // latitude of train stop location
double stop_lon;   // longitude of train stop location
```

Your class should also have a constructor and the following public member functions:

```
string get_id( ) const
string get_stop_name( ) const
double get_latitude( ) const
double get_longitude( ) const
```

---

<sup>1</sup>The data is in a common format; please read <https://developers.google.com/transit/gtfs/reference?csw=1> for more information.

## Written Part:

The C++ STL has many functions and functors. Here is your chance to try some of them. In a program when you use an STL algorithm add `#include<algorithm>`, and when you use an STL functor add `#include<functional>`.

For many of these problems you will need to look up information online. Here are some sources:

<http://en.cppreference.com/w/cpp>

<http://www.cplusplus.com/reference/algorithm/>

<http://www.cplusplus.com/reference/std/functional/>

<http://www.sgi.com/tech/stl/>

Fill in the correct code where you see a \*\*\*\*

```
vector<int> A {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
vector<int> B {1, 2, 1, 2, 1, 2};
vector<int> C{1, 2, 3, 1, 2, 3};
vector<int> D(6);
vector<int> E(10);
```

1. Copy the first 6 items of vector A into vector D

```
copy(****, ****, ****);
// D now equals {1, 2, 3, 4, 5, 6}
```

2. Count the number of ones in vector B

```
cout << count(****, ****, 1);
//prints out the number of times a one appears in B
```

3. In C++, there is a way to construct a unary functor from a binary functor. To do this you use an adapter, a function called `bind1st` or `bind2nd`. We use `bind1st` in this example to convert the STL binary predicate functor `not_equal_to` into a unary predicate by setting its first value to 1.

Count the number of items that are not equal to one in B

```
cout << count_if(B.begin( ), ****, bind1st(not_equal_to<int>( ), 1));
/*prints out the number of times a one doesn't appear in B.*/
```

4. Test to determine if the first 3 items of A are the same as C.

```
bool same;
same = equal(A.begin( ), ****, ****);
```

5. Find the first item in vector A which equals 5

```
vector<int>::iterator vecItr;
vecItr = find(****, ****, 5);
    // returns an iterator to 5 in vector A
if (vecItr != A.end( ))
    cout << ****;
    // prints out the value pointed to by vecItr
```

6. Find the first item in C that is greater than 2

```
vecItr = find_if(****, ****, bind2nd(greater<int>(), 2));
    // returns an iterator to 3 in C
if (vecItr != C.end( ))
    cout << ****;
    // prints out the value pointed to by vecItr
```