

CS2134 Homework Assignment 3A  
Spring 2016  
Due\* 4:30 p.m. Wed. Feb 17, 2016

February 14, 2016

Assignment 3A include a programming portion and a written portion. The programming portion must compile and consist of a single file ( hw03A.cpp), and the written portion should consist of a single file (hw03Awritten) in a .pdf format. Be sure to include your name at the beginning of each file! You must hand in both files via NYU Classes.

### Programming Part:

1. Add the method `erase( Vector<Object>::iterator vItr)` to the `Vector` class. The signature of your method should be:

```
iterator erase( iterator vItr)
```

2. Write a generic function template called `print_if` that:

- takes three parameters: two iterators `start`, `end`, and a functor `pred`  
`start` and `end` have the capabilities of a forward iterator, and refer to a range `[start,end)` in a container  
`pred` is a functor that takes an element in the range `[start,end)` as an argument and returns a `bool`
- prints<sup>1</sup> all items in the range `[start,end)` which evaluates to `true`
- runs in  $O(n)$  time where  $n$  is the number of items in the range `[start,end)`

The signature of your generic function template is:

```
template< class Itr, class UnaryPred >  
void print_if( Itr start, Itr end, UnaryPred pred )
```

---

\*A bonus of %10 percent will be given if you turn in this homework assignment by Mon. Feb 15 at 11:00 p.m.

<sup>1</sup>Print each item on its own line

3. Create a functor called `GPA_in_range` where:
  - the class has two private member variables of type `double`: `low` and `high`
  - the constructor has two parameters of type `double`: `l` and `h` which it uses to initialize the private member variables `low` and `high`
  - the overloaded `operator()` takes a single argument of type `student` and returns `true` if the student's gpa is in the range `[low, high]`. Otherwise the overloaded `operator()` returns `false`
4. Test your answer to programming questions 3 and 2 by creating a vector of type `student`. Use your code from questions 3 and 2 to print out all the students in your vector whose GPA is within `[3, 4]`. Turn in your testing code.

## Written Part

1. For the `vector` class, and for the following code<sup>2</sup> snippet:

```
vector<int> c { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 };
vector<int>::iterator itr1 = c.begin()+2;
vector<int>::iterator itr2 = c.begin()+4;
vector<int>::iterator itr3 = c.begin()+8;
cout << *(c.begin( ) + ( c.end( ) - c.begin( ) )/2 ) << endl;

c.erase(itr2);

cout << *itr1 << endl;
cout << *itr2 << endl;
cout << *itr3 << endl;
cout << *(c.begin( ) + ( c.end( ) - c.begin( ) )/2 ) << endl;
```

What is printed? Explain your answer.<sup>3</sup>

2. In written question 1, which of the `iterators` were *valid* after the `erase` method was called. We will say an iterator is still “valid” if it refers to the same item as before the method was called.
3. Using big-Oh notation, give the worst case run time for the method `erase`, which you implemented programming problem 1.
4. Finish writing the following function that subtracts one from every item in the range `[start,end)`.

```
template< class Itr >
void subtractOne( Itr start, Itr end )
{
    // Fill in the code here
}
```

For example, the following code snippet:

```
vector< int > a = { 144, 524, 230, 8 };
subtractOne( a.begin( ), a.end( ) );
```

results in the vector `a` now containing `143, 523, 229, 7`.

---

<sup>2</sup>Remember `c.end() - c.begin()` returns the number of items in the range `[ c.begin(), c.end() )`

<sup>3</sup>Hint: think of how you implemented the `erase` method in the `Vector` class.

5. For the `Vector` class we discussed in class, if we removed the word `explicit` in front of the constructor, i.e.

```
template <class Object>
class Vector
{
public:
    Vector( int initSize = 0 ) :
        theSize( initSize ), theCapacity( initSize + SPARE_CAPACITY )
        { objects = new Object[ theCapacity ]; }

    // ... rest of class the same as before
}
```

What would be printed by the following code snippet?

```
Vector<int> v(3);
v = 110;
cout << v.capacity();
```