**CS 515 Exercise C10: Operator overloading**

\_\_\_\_\_ /50

**Names: Ryan J. Skelly\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Wildcats ID’s : 988977861\_\_\_\_\_\_\_\_\_**

**Lecture Section: 02 (circle one)**

Assume you are writing a set class like the one in assignment 4P, and you want to implement <= and >= as subset operations ⊆ and ⊇, respectively. Add the appropriate declaration for these functions to the (partial) header file on this side of the page, and then implement them on the back.

Class Set

{

public:

//Constructors, destructors, etc.

…

//Basic operations

…

//Iterators

class Iterator;

Iterator find(ELEMENT\_TYPE) const; //

Iterator begin() const; //iterator at the beginning

Iterator end() const; //iterator just "past" the end

…

Class Iterator

{

public:

//Constructors, etc.

…

//Overloaded operators

ELEMENT\_TYPE& operator\*();//dereference

Iterator operator++(int); //post-increment operator

Iterator operator--(int); //post-decrement operator

…

}

…

bool operator <=(const Set &rhs) const;

bool operator <=(const Set &rhs) const;

}

…

Write the implementation of <= and >= (as subset operators) below. Feel free to use a third function if it will help eliminate copied code.

bool Set::operator <-(const Set &rhs)const

{

For (set::iterator it – begin(); it !- end(); it++)

{

if (rhs.find(\*it) == rhs.end() )

return false;

}

return true;

}

bool operator >-(const Set &rhs)const;

{

return rhs <= \*this;

}