**Assignment 1: Single-Linked Lists –Rule of Three (Hint)**

**Objective:**

1. To provide the sample codes for the rule of three (for this assignment)

**Hints: Rule of Three**

* Copy constructor
* Assignment operator (“=”)
* Destructor

The following codes are examples of the rule of three. You can use your own codes. The goal is to provide you with the sample codes.

// copy constructor

//creates a new linked list where the contents are a “deep copy” of the provided list

Binary::Binary(const Binary &b)

{

firstTerm = nullptr;

for (BinaryNode \*current\_other = b.firstTerm; current\_other != nullptr;

current\_other = current\_other->next)

{

this->set\_bit(1, current\_other->degree);

}

}

// assignment operator

// sets the current link list to be a “deep copy” of the provided list.

// make sure to check if assigning to itself, and make sure to free old memory

// before making the copy.

Binary& Binary::operator=(const Binary &other)

{

if (this = = &other)

{

return \*this;

}

Binary bCopy = Binary(other);

/\* Swap firstTerm of other and bCopy. \*/

BinaryNode \*pTemp = this->firstTerm;

this->firstTerm = bCopy.firstTerm;

bCopy.firstTerm = pTemp;

return \*this;

}

// destructor

// make sure that all memory is returned (freed up) correctly

~Binary();