

# Assignment 1 - Iterative Algorithms

[Submit Assignment](#)

---

**Due** Jan 22 by 11:59pm      **Points** 100      **Submitting** a file upload      **File Types** java  
**Available** Jan 12 at 7pm - Mar 19 at 11:59pm 2 months

---

For this assignment you will be enhancing the BigInteger class. BigInteger is a Java class that implements an integer value of arbitrary size. Starter code is provided on the canvas module for Week 1. It currently provides these public constructors:

- BigInteger() - initialize a BigInteger to 0
- BigInteger( String ) - initialize a BigInteger from a String of digits (0..9)
- BigInteger( int ) - initialize a BigInteger from the digits of an integer stored in an int
- BigInteger( BigInteger ) - initialize a BigInteger from another BigInteger (deep copy)

There is also a private constructor:

- BigInteger( List<Byte> ) - a helper function called from within BigInteger to initialize a new BigInteger from the digits of an existing BigInteger. There is no error checking, so it is declared private to prevent other objects in the program from initializing a BigInteger with a string of possibly non-numeric values.

The digits of a BigInteger are stored in reverse order to simplify the syntax when doing arithmetic. For example, a BigInteger constructed from an int value, 12345, is stored in the Byte list as [5, 4, 3, 2, 1]. In other words, the lowest order digit in 12345, the value 5, is element 0 in the list.

Therefore, when constructing a string representation in the toString() method, the digits are reversed.

The class BigInteger in the starter code contains the method, add( BigInteger bi ). It returns the reference of a newly constructed BigInteger containing the sum of "this", and "bi" to the calling method. The BigInteger values contained in "this" and "bi" are left unchanged.

The assumption throughout the class is that the integers contain no leading zeroes.

For this assignment you are to add one of the two following methods:

- multiply( BigInteger bi ) - return a reference to a newly constructed BigInteger that contains the product of "this" and "bi". The original values stored in "this" and "bi" must not be changed by the method.
- subtract( BigInteger bi ) - return a reference to a newly constructed BigInteger that contains the difference of "this" and "bi". The original values stored in "this" and "bi" must not be changed by the method.