

CSC 230 Elementary Data Structures and Algorithms  
Spring 2014 - Assignment 3  
Due Friday, February 21, 2014, 8:00am

---

### Assignment 3 Skills

Using Inheritance  
Understanding Polymorphism  
Understanding Class design  
Using an Interface to define behaviors for a class

### Assignment 3 Background

For this assignment, you will write several classes that model different types of sequences.  
Recall:

1. An arithmetic sequence is a sequence that has a constant difference between terms. Given the first term and the constant difference, you can find any term in the sequence.
2. A geometric sequence is a sequence that has a constant ratio between terms. Given the first term and the constant ratio, you may find any term in the sequence.
3. The Fibonacci sequence is a sequence where each term in the sequence is the sum of the two terms that precede it. The first two terms in the Fibonacci sequence are 0 and 1.

*\*We will create a generalized Fibonacci sequence\**

### Assignment 3 Requirements

1. (10%) Write an **interface** called **SequenceOps**. This interface will define behavior for objects that have sequential properties. This interface should have the following methods defined:
  - (a) A method that gets the  $n$ th term in a sequence. This method requires a parameter for  $n$  and returns the  $n$ th term.
  - (b) A method that sums the first  $n$  terms in the sequence. This method requires a parameter for  $n$  and returns the sum of the first  $n$  terms.
2. (15%) Write an **abstract** class called **Sequence**. This class defines properties and methods that all sequences will have. This class **must** implement the **SequenceOps** interface. This class should include the following:
  - (a) a **private** field for the first term in the sequence, type: **double**.
  - (b) a default constructor that sets the first term to 1.
  - (c) a constructor that sets the first term using a parameter value.
  - (d) A method that returns the first term in the sequence.
  - (e) A **toString** method that returns a **String** of the first 10 terms in a sequence in a comma delimited list.

3. (15%) Write a class called `ArithmeticSequence` that extends the `Sequence` class and implements the `SequenceOps` interface. This class should include the following:
  - (a) a `private` field for the common difference.
  - (b) a default constructor that initializes the first term to 1 and the common difference to 0.
  - (c) a constructor that uses parameters to initialize the common difference and the first term.
4. (15%) Write a class called `GeometricSequence` that extends the `Sequence` class and implements the `SequenceOps` interface. This class should include the following:
  - (a) a `private` field for the common ratio.
  - (b) a default constructor that initializes the first term to 1 and the common ratio to 1.
  - (c) a constructor that uses parameters to initialize the common ratio and the first term.
5. (15%) Write a class called `FibonacciSequence` that extends the `Sequence` class and implements the `SequenceOps` interface. This class should include the following:
  - (a) a `private` field for the second term.
  - (b) a default constructor that initializes the first term to 0 and the second term to 1.
  - (c) a constructor that uses parameters to initialize the the first and second terms.
6. (20%) Write a `main` method in a file called `SequenceTest.java`. Your main method should:
  - (a) Create a `Sequence` variable
  - (b) Read from the input file `SEQ.dat`, which is formatted as follows:
    - i. The first item on a line is a letter that represents the type of sequence (*A*, *G*, or *F*).
    - ii. the second item on a line is a double value that represents the first term in the sequence.
    - iii. the thrid item on a line is a double value that represents either the common difference, common ratio, or second term, depending on the type of sequence.
  - (c) Use `toString` and other methods to produce the output in the following format (for each line of input):
    - i. Print to standard out: 'The first 10 terms in the sequence are:'
    - ii. Print to standard out a comma-delimited list of the first 10 terms.
    - iii. Print to standard out: 'The sum of the first 5 terms =' the sum of the first 5 terms.
7. (10%) Provide comments where appropriate

### Assignment 3 Restrictions

1. (-10) Do not use any reference variables for `ArithmeticSequence`, `GeometricSequence`, or `FibonacciSequence`. Use only a single `Sequence` variable in you main method.
2. (-10) Do not override `toString` method in `ArithmeticSequence`, `GeometricSequence`, or `FibonacciSequence` class.
3. (-5) The `toString` method **must** return a `String` and **cannot** print to standard out. All printing should be done in the main method.
4. (-10) Do not use recursion.

**Assignment 3 Submission** Submit on Blackboard:

1. SequenceOps.java
2. Sequence.java
3. ArithmeticSequence.java
4. GeometricSequence.java
5. FibonacciSequence.java
6. SequenceTest.java

*Required* Each submitted file should include your name and a statement that this is your own work. This should appear as a comment at the beginning of any code file.