

# Exam 2

**EE363 - Fall 2015**

## **Software Components and Generic Programming**

**Thanksgiving Week / Dead Week Take Home**

### **Instructions**

Please read each question carefully, and answer each in the space provided. Use the back side of the pages if you need additional space, and be sure to indicate clearly which questions you are answering if you do.

---

Name

*For each of the following statements, state whether the statement is true or false, and provide a brief explanation. Simply stating "true" or "false" will result in no-credit, even if the answer is correct.*

1. Big Oh Analysis provides an exact measurement of the number of steps required to for a method to execute.

2. The following Big-Oh statements are equivalent:

$O(7N^2 + 3N)$

$O(N^2 + 99)$

$O(15N^2)$

3. Big Oh analysis is only useful for choosing which Java Collection instance to use.

*Provide a Big Oh analysis for each of the following code snippets.*

```
public void sortArray( Integer[] values ) {
    if ( values.length == 0 )
        return;

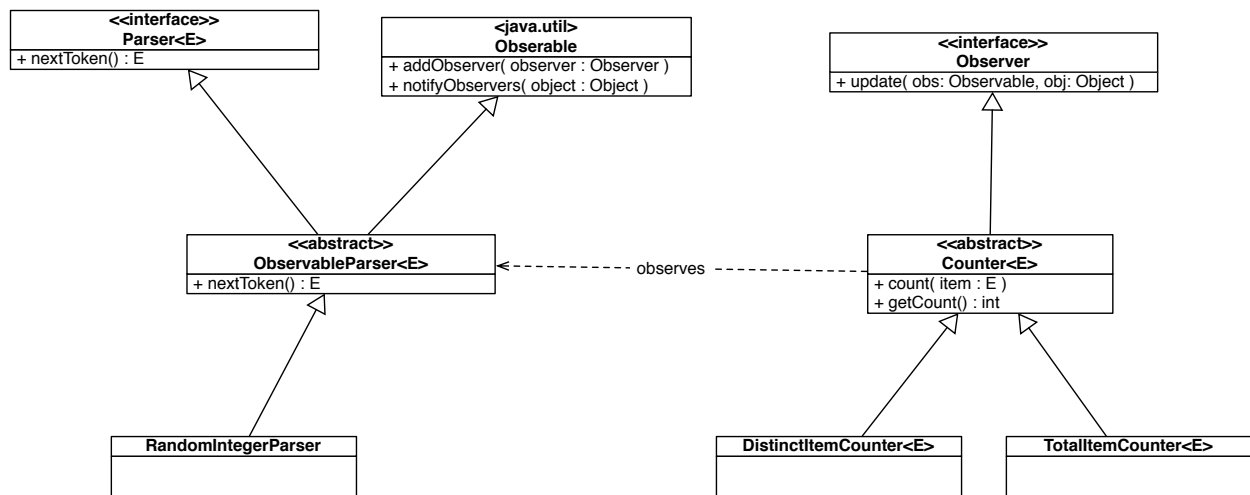
    for ( int i = 0; i < values.length; ++i ) {
        for ( int j = i + 1; j < values.length; ++j ) {
            // Is the value at position 'j' less than the value at
            // position 'i'?
            if ( values[j].compareTo( values[i] ) < 0 ) {
                // If so, let's swap that values.
                Integer temp = values[i];
                values[i] = values[j];
                values[j] = temp;
            }
        }
    }
}
```

```
public void sortArray( Integer[] values ) {
    TreeSet<Integer> temp = new TreeSet<>();
    for ( int i = 0; i < values.length; ++i )
        temp.add( values[i] );

    int i = 0;
    for ( Integer value : temp ) {
        values[i++] = value;
    }
}
```

*Provide a brief answer for each of the following questions:*

1. Which type of collections are *not useful* if you need to retain knowledge about the order in which elements were added to the collection?
  
  
  
  
  
  
  
  
  
  
2. What is the Big-Oh runtime of inserting an element into the first position of the following types of collections:
  - a. Linked List
  - b. Array List
  - c. Hash Set
  
  
  
  
  
  
  
  
  
  
3. True or False (and explain): In Java, you may compare any two objects using the “less than operator”: <
  
  
  
  
  
  
  
  
  
  
4. True or False (and explain): In Java, like in C++, all objects are pass by value unless you use the *pass by reference* operator: &.



Consider the UML diagram above

- 1) Identify each design pattern used in the design of this counter solution. For each design pattern:
  - Identify the which types participate in that pattern
  - Identify what role in the pattern each participating type plays
  - Describe the potential benefits of having used the pattern.

**Bonus Question:** Consider a public method in a class, with the following signature:  
 public String reverse( String foo );

This method will return a string, which is the reversed-representation of the input string 'foo'.

Which C++ method signature is the closes match to the way Java behaves, with respect to the parameter passing:

- std::string reverse( std::string &foo );
- std::string reverse( std::string \*foo );
- std::string reverse( const std::string \*foo );
- std::string reverse( std::string \*const foo );
- std::string reverse( const std::string \* const foo );