CSCI 2341 Midterm Test

Tuesday, 31 May 2016

Name: _____

Student #: _____

General Instructions

Read and follow all directions carefully.

When writing programs or program segments, use the course style for identifiers and indentation. Print clearly so that the grader can tell that the conventions are being followed.

Note: not all spaces need to be filled in, but you will not need more space than is provided. **You do not need to comment code unless the question says otherwise.** You may assume that all required packages have been included, unless the question says otherwise.

There are 62 points available. You have 90 minutes to complete it. Good luck.

Question	Grade	Out Of
1		2
2		8
3		8
4		2
5		8
6		6
7		4
8		4
9		8
10		4
11		8
Total:		**62**

1. **[2]** Consider the function shown below, which modifies the BagInterface given to it.

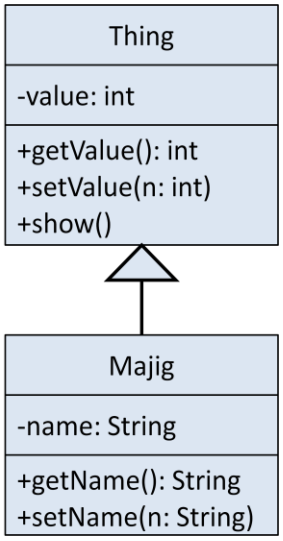
```
public static void modify(BagInterface<String> bag) {
    BagInterface<String> newBag;
    while (!bag.isEmpty()) {
        String s = bag.remove();
        if (!newBag.contains(s)) {
            newBag.add(s);
        }
    }
    while (!newBag.isEmpty()) {
        bag.add(newBag.remove());
    }
}
```

   Suppose `myBag` contains `"A"`, `"B"`, `"C"`, `"D"`, `"A"`, `"E"`, and `"D"`. What does `myBag` contain after `modify(myBag)` has been called?

2. **[8]** Create an interface (**Unknown**) that will hold one object of some specified type -- for example an **Integer** or a **String** or a **Scanner**. Each **Unknown** object can only hold the kind of object it's supposed to – so you're not allowed to put a **Scanner** into an Unknown that holds **Integer**s.

   The interface has a method to get the object in it, another to set that object, and a third to test whether that object exists. Define the interface. (*No comments necessary!*)

3. **[8]** Consider the UML diagram to the right. Write the class definition for **Majig**.



4. **[2]** Suppose we have an **ArrayBag** with the following array holding its contents:

0	1	2	3	4	5	6	7	8	9
"Anita"	"Bill"	"Chantal"	"Djeri"	"Eli"	"Frank"	"Gordon"	/	/	/

Show the contents of the array after `remove("Bill")` has been executed.

5. **[8]** Write the `getFrequencyOf` method for the **LinksBag** class:

```
public class LinksBag<T> implements BagInterface<T> {
    private class Node {
        private Node   next;
        private T      data;
    }
    private Node first;
    …
}
```

The method counts how many times a given entry appears in the **LinksBag** object. Do not use the toArray method, nor any method that need to be imported, and do not modify the contents of the bag in any way. **Note that this class does not have a numEntries field.**

6. **[6]** State the order of magnitude of each of the following functions. State a tight bound in each case.

   a) $n^3 + 6n + 5$ O(_____)

   b) $5\ log\ (n^5) + n^2 + 9$ O(_____)

   c) $(n^4 + 6n^3 + 5) / (5n + 4)$ O(_____)

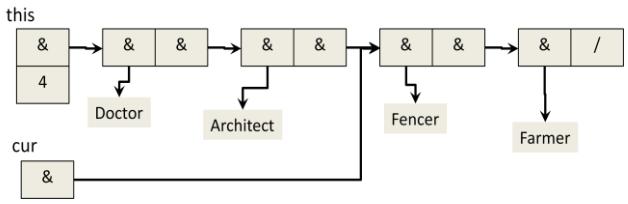
   d) $(n\ log\ n) + n$ O(_____)

   e) $3 + 6 + 8 + 10 + ... + 2(N - 1) + 2N + 2(N + 1)$ O(_____)

   f) $(3 + 4 + \ldots + n)(1 + 2 + 3 + ... + (n / 2))$ O(_____)

7. **[4]** Write a summation formula for the value of the expression W = 20 + 25 + 30 + 35 + 40 + 45 + ... + 195 + 200? Rewrite it into a difference of summations. Use the important formula from the notes to translate it into a math expression using only numbers, multiplication, division and subtraction. You do not need to calculate the final result.

8. **[4]** Suppose we have the following arrangement of nodes in memory (where `this` is a `LinkedBag`):



Show how the structure would look after the command

```
cur.next =
this.firstNode;
```

has been executed by putting X(s) thru old arrow(s) and adding new arrow(s).

9. **[8]** The button withdrawButton is supposed to call the method makeWithdrawal() when it is clicked. Give it a suitable ActionListener using

   a) an anonymous class

   b) a lambda expression

10. **[4]** Revise the following code so that the method returns a count of every operation (assignment or comparison) that happens while it's running.

```
public static void doThis(int n) {

    int k;

    while ( n > 0 ) {

        k = n;

        while ( k <= n ) {

            if ( k > n ) {

                k = k / 2;

            } else {

                k = 3 * k + 1;

            }

        }

        n = n - 1;

    }

}
```

11. **[8]** Multiple Choice: select the *best available* answer from the options shown.

- If we have a `Object` variable `obj`, and we want to check to see if it holds a `Circle` object, the way to do that is
  a) `if (Circle extends Object)`
  b) `if (Circle implements Object)`
  c) `if (Circle instanceof Object)`
  d) `if (obj extends Circle)`
  e) `if (obj instanceof Circle)`

- The core methods of BagInterface<T> are
  a) add(T) and toArray()
  b) clear() and isEmpty()
  c) equals(T) and getFrequencyOf(T)
  d) getCurrentSize() and contains(T)
  e) remove() and remove(T)

- What is the output of the following code, assuming that setColor and getColor do as their names imply?

```
Car myCar = new Car("Blue");
Car disCar = myCar;
Car stevesCar = new Car("Green");
disCar.setColor("Red");
myCar.setColor("Yellow");
System.out.print(disCar.getColor());
```

  a) Blue
  b) Green
  c) Red
  d) Yellow
  e) (it's impossible to tell from the given information)

- The `Comparable<T>` interface requires the ___ method to be defined:
  a) `boolean compareTo(T one, int result)`
  b) `boolean compareTo(T one, T other)`
  c) `boolean compareTo(T other)`
  d) `int compareTo(T one, T other)`
  e) `int compareTo(T other)`

- The command to create a new array of size N and type T (where T is the base type of a templated class) is:
  a) `Object[] arr = (Object[])new T[N];`
  b) `Object[] arr = new T[N];`
  c) `T[] arr = (T[])new Object[N];`
  d) `T[] arr = new (T[])Object[N];`
  e) `T[] arr = new Object[N];`
  f) `T[] arr = new T[N];`

- Among the reasonable options for dealing with a possible operational failure in an ADT operation is NOT:
  a) have the method return a boolean value: true for success, false for failure.
  b) print an error message.
  c) return a special value (such as null or -1, if they are available).
  d) throw an exception.
  e) (all of the above are reasonable options for dealing with failure)

- Which of the following is NOT a problem that might arise when the add method is called on a bagInterface?
  a) the argument might already be in the bag.
  b) the argument might be null.
  c) the bag may be full.
  d) the bag might not have been properly initialized.
  e) (all of the above are possible problems for the add method)

- A program reads a number, N, and then prints out a triangle like the following:

```
1
1 2
1 2 3
 . . .
1 2 3 ... N
```

What is the magnitude of work for this program, assuming it's as efficient as it can be?

a) O(1)
b) O(*log* N)
c) O(N)
d) O(N *log* N)
e) $O(N^2)$
f) $O(2^N)$