

## CS1110 Prelim 2 10 Nov 2009

This 90-minute exam has 5 questions (numbered 0..4) worth a total of 100 points. Scan the whole test before starting. Budget your time wisely. Use the back of these pages if you need more space. You may tear the pages apart; we have a stapler at the front of the room.

**Question 0 (2 pts).** Write your last name, first name, and Cornell NetId, legibly, at the top of each page (hint: do it now).

Most of the questions on this exam have to do with RNA (Ribonucleic acid), an important molecule. You don't have to know about RNA to answer these questions, but it may help you to know that the questions are not just abstract but have a basis in real problems. Here's what you have to know:

RNA is a chain or string of "nucleotides", denoted by symbols like C (for cytosine) and G (for guanine). There are other such symbols, e.g. A (for adenine) and U (for uracil). Each nucleotide has a *complementary* nucleotide with which it bonds; C and G are a complementary pair, and so are A and U.

**Question 1a (11 points) Abstract classes.** Below, write an abstract class named `Nuc` (for "nucleotide"). Be sure to include proper specifications for all methods and a class invariant. We gave you some of the specifications; copy them. (Seeing the specs in the proper place should help you with some of the remaining questions.) The class should have these (and only these) components:

- A private **char** field, which is the nucleotide's symbol, e.g. 'C'
- A constructor with this specification:  
/\*\* Constructor: a Nuc with symbol sym.  
Precondition: sym is the character for a nucleotide. \*/
- A public getter method `getSymbol`, which returns the nucleotide's symbol
- A public *abstract* method `isComplement (Object ob)` with this specification:  
/\*\* = " ob is a nucleotide (i.e. a Nuc) and is complementary to this one." \*/

**Question 1b. (3 Points). Abstract classes.** What is the purpose of making a class abstract?

**Question 1c (11 Points) Subclasses.** This problem tests whether you can write constructors in subclasses and understand the concepts behind writing method `equals`, such as apparent and real classes and casting. Complete the methods shown in subclass `CNUC` of `Nuc` below. Note that you don't know the names of other subclasses of `Nuc`, and you don't have to. Notice that the **char** field in `Nuc` is private.

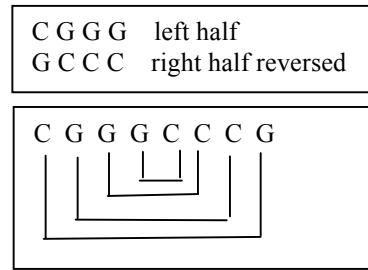
```
/** An instance is a cytosine nucleotide (symbol 'C') */
public class CNuc extends Nuc {
    /** Constructor: a new cytosine molecule */
    public CNuc() {

    }

    /** = "ob is a Nuc whose symbol is 'G' (guanine)"
        (remember, C and G are a complementary pair) */
    public boolean isComplement(Object ob) {

    }
}
```

**Question 2. Recursion.** We call the sequence C G G G C C C G a *perfect hinge* because if the right half is reversed and placed under the left half, as shown to the right, corresponding characters are complementary. Structures similar to perfect hinges play an important role in biology.



Another way to view the hinge property is shown in the diagram to the right. This diagram shows that the sequence C G G G C C C G is a perfect hinge because of the way each element is connected to another element by a line to form a complementary pair.

From the definition of a perfect hinge, we can deduce that an empty sequence is a perfect hinge and a sequence with an odd number of elements is not a perfect hinge.

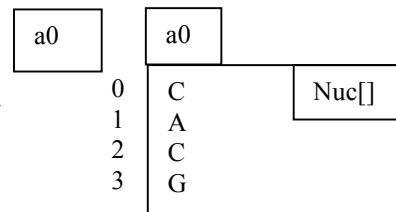
**Part 2a (19 points).** An array `b` of type `Nuc[]` can contain an RNA (a sequence of nucleotides), and we might ask whether a segment `b[s..e]` of `b` is a perfect hinge. Below, write the body of function `isHinge`, to be placed in class `Nuc`. Do not use a loop. Use recursion. Hint: This shows some resemblance to the palindrome problem.

```
/** = "b[s..e] is a perfect hinge" */
public static boolean isHinge(Nuc[] b, int s, int e) {
```

}

**Part 2b (6 points)** To the right, variable `nuc` contains the name of an object of type `Nuc[]`. In that object, we have represented the elements with the symbols of the Nucs whose names they contain, rather than drawing the `Nuc` objects themselves. Execute the call

```
isHinge(nuc, 0, nuc.length-1)
```



drawing all frames for calls on `isHinge` (*do not draw any other frames*); stop when the call is finished OR as soon as two frames have been drawn.

**Question 3 (23 points).** The array { 'C', 'A', 'C', 'G', 'C' } has 2 “dips” —positions in which the following character is smaller. (As a reminder, characters are ordered by their ASCII integer representation, and upper case letters are in alphabetical order.) The two dips here are from 'C' to 'A' and from 'G' to 'C'. As another example, the array { 'C', 'A', 'U' } has 1 dip.

Complete the following method. We have written the command “Set n to the number of dips in b”, and the postcondition, which asserts that the command has been performed. You have to write (1) the three parts of the for-loop header **for** (...), (2) the loop invariant, (3) the initialization to make the loop invariant true, and (4) the repetend. Naturally, all code must be consistent with the loop invariant.

/\*\* = number of dips in b.

Precondition: b contains at least one element. \*/

```
public static int numberOfDips(char[] b) {
    int n;

    // Set n to the number of dips in b.

    // loop invariant:

    for (           ;           ;           ) {

    }

    // postcondition: n = number of dips in b[0..b.length-1]
    return n;
}
```

**Question 4 (25 points) Arrays, for-loops, and methods.** We need a method that, given a string of symbols that represent nucleotides, for example, the string "CCGAUGCC", produces an array of Nuc objects corresponding to the symbols. For the string shown, the array would have 8 elements.

(a) Below, write such a method. We have given only its specification; you write the rest. Its name should be `nucArray`, and it is to be placed in class `Nuc`. Its parameter is a string `s` of symbols that represent nucleotides.

Your method must use a for-loop to fill in array values and must return a new array of type `Nuc[]`. You have to make sensible choices about whether the method should be public or private, static or non-static, what type the parameter should have, etc. You do not have to use a loop invariant, but you might find it useful.

For this problem, and this problem only, assume that the subclasses of `Nuc` for each of the nucleotides C, G, A, and U are named `CNuc`, `GNuc`, `ANuc`, and `UNuc`.

`/** = an array of NucS corresponding to the symbols in s.`

`Precondition: The only characters that appear in s are 'C', 'G', 'A', and 'U'*/`

0	_____	out of 02
1	_____	out of 25
2	_____	out of 25
3	_____	out of 23
4	_____	out of 25
Total _____		out of 100

(b) To the right is a variable `nu`, which contains the name of an object. (We do not write the components in the object.) Write below the classes to which `nu` may be cast. Also, indicate which of these casts will be done automatically when necessary and which must be given explicitly.

