Cornell net id _____ **Name** _____

Section day _____ **Section time** _____

CS 100J Prelim 2 — 17 October 2006

Have a good break!!!

This 90-minute exam has 6 questions (numbered 0..5) worth a total of 100 points. Spend a few minutes looking at all questions before beginning. Use the back of the pages, if you need more space.

Question 0 (2 points). Fill in the information, legibly, at the top of each page (Hint: do it now.)

Question 1 (10 points). (a) Define "local variable" and state when a local variable is created during a method call.

0	_____ out of 02
1	_____ out of 10
2	_____ out of 15
3	_____ out of 14
4	_____ out of 22
5	_____ out of 37
Total	_____ out of 100

(b) Below is a loop. Fill in the invariant, then the initialization, then the loop body

```
// Store in c the number of positions i in Strings s1 and s2 such that s1[i] == s2[i]
int n= Math.min(s1.length(), s2.length());
int c=          ;
// invariant:

for (int k= 0; k < n; k= k+1) {




}
// c = no. of positions i in s1[0..n-1] and s2[0..n-1] such that s1[i] == s2[i]
```

Cornell net id _____ **Name** _____

Section day _____ **Section time** _____

Question 2 (15 points). At the bottom of the page are definitions of three classes: Student, Frosh, and Senior. Assume that these three statements have been executed:

```
Student a= new Frosh("A", 3.1);
Frosh c= new Frosh("C", 3.2);
Senior d= new Senior("D", 3.3);
```

Write the value of each expression given below; if one leads to an error, write "ERROR". It may help you to draw the objects that are created by execution of the three statements above.

(2a) c.Bigger(a)

(2b) a.getJob()

(2c) c.getJob()

(2d) d.getJob()

(2e) c.getSound()

(2f) d.getSound()

(2g) ((Frosh)a).getSound()

(2h) ((Frosh)d).getSound()

(2i) ((Senior)d).getSound()

(2j) ((Senior)a).getSound()

```
public class Student {
    private String name; // name of student
    private double gpa; // gpa of student
    /** Constructor: Student with name n,
        gpa gpa */
    public Student(String n, double gpa) {
        name= n;
        this.gpa= gpa;
    }
    /** = "this Student has a better gpa than s"
        Precondition: s is not null */
    public boolean Bigger(Student s) {
        return this.gpa > s.gpa;
    }
    /** = this student's statement */
    public String getSound() {
        return "";
    }
    /** = name of this Student */
    public String getName() {
        return name;
    }
    /** = gpa of this student */
    public double getGpa() {
        return gpa;
    }
}
```

```
public class Frosh extends Student {
    /** Constructor: a Frosh with name n, gpa a */
    public Frosh(String n, double gpa) {
        super(n, gpa);
    }
    /** = this student's statement */
    public String getSound() {
        return "I'm new here.";
    }
    /** = next job of this frosh */
    public String getJob() {
        return "work-study";
    }
}

public class Senior extends Student {
    private String major= ""; // eg. "English"
    /** Constructor: a Senior with name n, gpa a */
    public Senior(String n, double a) {
        super(n, a);
    }
    /** = first job of this Senior */
    public String getJob() {
        return "Cabbie";
    }
}
```

Cornell net id _____ **Name** _____

Section day _____ **Section time** _____

Question 3 (14 points). Write the bodies of the two methods whose specifications and headers are given below, assuming that these methods are in class Senior that appears on the previous page.

```
/** Constructor: a Senior whose major is "CS", has name m, and has a gpa of 3.8. */
public Senior(String m) {




}

/** = “ob is a non-null Senior with the same fields as this Senior” */
public boolean equals(Object ob) {




}
```

Cornell net id _____ **Name** _____

Section day _____ **Section time** _____

Question 4 (22 points). We need your help in writing the body of a method, given below, to edit sentences in a String s. Example: change sentences 'the title is "Airport." it is ok, i.e. alright.' to 'The title is "Airport". It is ok, i.e. alright.'

A string s contains a sequence of sentences, each ending in a period, with no blank (space) chars at the beginning and end and one blank char between each word, as usual. But it may have the following mistakes.

(1) The first letter of the first word of a sentence may not always be capitalized; it should be. Here's the rule: Capitalize the first char of a word if it is preceded by ". " (a period and a space) but not preceded by "i.e. " (since "i.e." does not end a sentence).

(2) If a period appears before ", switch them, i.e. put the period after the quote mark. In this question, quotes are used only for titles of books, and the period is not part of the title!

Instructions: Points (1) and (2) must be handled in a single for-loop that processes a range of integers, building up the result in a String variable res (say). Here's a hint: the repetend can process a character of s according to point (1), looking at res if necessary, and then handle point (2) by dealing with res alone.

If you have to write a string literal with a space in it, please indicate the space somehow, e.g. "‡b‡b" could indicate four chars: space, b, space b. Just use some sign that lets us know where a space is.

Write the whole method on the back of some other page, if you wish.

Here are methods you can use, for any string s or character c		
Return	**Method**	**Purpose**
char	`s.charAt(i)`	= `s[i]`
int	`s.length()`	= number of chars in s
`String`	`s.substring(i,j)`	= s[i..j-1]
bool	`s.endsWith(s1)`	= true if s ends with string s1, false otherwise
char	`Character.toUpperCase(c)`	= c in upper case, if it is a letter; otherwise, c
`String`	`s.trim()`	= copy of s with blanks on either end removed

```
/** = a copy of s edited as discussed above */
public static String edit (String s) {




}
```

Cornell net id _____ **Name** _____

Section day _____ **Section time** _____

Question 5 (37 points). (a) Write the bodies of the following two methods. Neither should use a loop. The first should not be recursive. The second must be recursive —and it may call the first if you think it useful.

```
/** = i as a string but with leading 0's if necessary, so that the result has length 3.
      Precondition: 0 <= i < 1000
      Example: digit3(5) is "005".  digit3(61) is "062".  digit3(123) is "123". */
public static String digit3(int i) {




  }
```

```
/** = A string that contains a representation of integer i, but with a comma every three digits
      from the right. Precondition: i >= 0.
      Example: toString(43) is "43". toString(5243642) is "5,243,642". */
public static String toString(int i) {




  }
```

Cornell net id _____ **Name** _____

Section day _____ **Section time** _____

(b) In Assignment A2, you implemented a class Elephant, part of which is shown below —we put only the fields and methods needed for this question. The family tree of an elephant consists of the elephant, its known parents, their known parents, etc. To the right, we show a family tree for an elephant r1. Elephant r1's mother is r2; its father is r3. Elephant r3's parents are unknown. And, r2's parents are known, but *their* parents are not. The size of the tree —the number of elephants in it— is 5.

Write the body of recursive function treeSize, which has been put into class Elephant. You may not use a loop; you must use recursion.

```
public class Elephant {
    private Elephant father; // father (null if unknown)
    private Elephant mother; // mother (null if unknown)
    /** = number of elephants in this Elephant's family tree */
    public int treeSize() {









    }
}
```