Cornell net id _____ **Name** _____

CS 100J Prelim 1 23 February 2006

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 so that you can see what is expected. Budget your time wisely. Use the back of the pages if you need more space.

0 _____	out of 02
1 _____	out of 16
2 _____	out of 18
3 _____	out of 22
4 _____	out of 22
5 _____	out of 20
Total _____	out of 100

Question 0 (2 points). Write your netid and your name, legibly, at the top of each page (Hint: do it now). Gries's netid is djg17.

Question 1 (16 points).

(a) What is the difference between a static method and a non-static method?

(b) Name three types of variables and their scopes.

(c) What is a parameter?

(d) For each of the following expressions, write its value beside the expression. If the evaluation leads to an error, write "error" instead.

```
false && (33/0 == 1)
("5" + 9)/2
true && (4 > 5)
```

Cornell net id _____ **Name** _____

Question 2 (18 points): At the bottom of the page are two class definitions. Draw one folder (object, instance) of class Politican and then a folder of class Governor. You need not include the partition for superclass Object.

```
public class  Politician {
    private String name= null;   // politician's name

    /** = "p1 and p2 are the same folder" */
    public static boolean areEqual (Politician p1, Politi-
                                        cian p2) {
        return p1 == p2;
    }

    /** Constructor: politician named n */
    public Politician (String n) {
        name= n;
    }

    /** Set this politician's name to n */
    public void setName (String n) {
        name= n;
    }

    /** = this politician's name */
    public String getName () {
        return name;
    }

    /** = the String representation of this Politician */
    public String toString () {
        return " politician " + getName ();
    }
}
```

```
public class Governor extends Politician {
    private String state= null; // the governor's
                                // state (e.g. NY)

    /** Constructor: a governor of state s, named n */
    public Governor (String n, String s) {
        super (n);
        state= s;
    }

    /** = this governor's state */
    public String getState () {
        return state;
    }

    /** Set this governor's state to s */
    public void setState (String s) {
        state= s;
    }

    /** = a String representation of this instance */
    public String toString () {
        return super.toString () + " the governor of " +
                getState ();
    }
}
```

Cornell net id _____ **Name** _____

Question 3 (21 points). Answer the following questions about classes Politician and Governor that are defined in Question 2.

(a) What is the superclass of class Politician?

(b) Name one method that class Politician inherits from its superclass.

(c) Name a method that class Politician overrides.

(d) Suppose the following two statements have been executed.

```
Politician myPoly= new Politician ("Arnold");
Politician yourPoly= new Politician ("Sean");
```

What is the value of the function call Politician.areEqual (myPoly, yourPoly)?

Now suppose this statement is executed:

```
yourPoly= new Politician ("Arnold");
```

Now what is the value of Politician.areEqual (myPoly, yourPoly)?

(e) State the general steps in evaluating the new-expression given below. Then describe how to execute the assignment statement.

```
g= new Governor ("Arnold", "California");
```

(f) What is the result of the expression g.toString ()?

Cornell net id _____ **Name** _____

Question 4 (20 points). This question refers to class `Politician` of Question 2.

(a) Write a subclass `President` of `Politician`, according to the following specifications:
(1) Class `President` contains a field for the number of months in office.
(2) The constructor of the class should let one specify the President's name and the number of months in office.
(3) It should have a method that returns the term of the President. ie. this method returns 1 if the President has been in office less than 4 years and 2 otherwise.
(4) The class should have a method "toString()" that returns more information than the default toString() method —Anything reasonable is fine.

Cornell net id _____ **Name** _____

Question 4 part (b) Complete the code below to make an appropriate JUnit test-case method to test your constructor, the method specified in part (3) of 4a, and toString().

```
public void testPresident() {



}
```

Cornell net id _____ **Name** _____

Question 5 (20 points). On the next page, write a method `english2PigLatin()` to translate an English word into its Pig Latin equivalent. A precondition is that the English word

(1) is not empty,
(2) contains only lower-case letters, and
(3) has a 'u' after each 'q' in it.

Pig Latin is not a different language but an encoding of English. Your method should use the following Pig Latin dialect:

(a) The vowels are 'a', 'e', 'i', 'o', 'u'. A 'y' that is NOT the first letter of a word is also considered a vowel. All other letters are consonants. E.g. "yearly" has three vowels ('e', 'a', and the last 'y') and three consonants (the first 'y', 'r', and 'l').

(b) If the English word begins with a vowel, append "hay" to the end of the word to get the Pig Latin equivalent.
e.g. "ask" becomes "askhay" "use" becomes "usehay"

(c) If the English word starts with 'q', assume it is followed by 'u', move "qu" to the end of the word, and append "ay".
e.g. "quiet" becomes "ietquay" "quay" becomes "ayquay"

(d) If the English word begins with a consonant, all the consonant letters up to the first vowel (if any) are moved to the end of the word, and "ay" is appended to get the Pig Latin equivalent.
e.g. "tomato" becomes "omatotay" "school" becomes "oolschay"
"you" becomes "ouyay" "my" becomes "ymay"
"ssssh" becomes "sshay"

You may use the following methods. The first five are from class String. s is any String. Assume that the last method is in the same class as english2PigLatin().

Return	Method	Purpose
`char`	`s.charAt(i)`	= the character at position `i` of s
`int`	`s.length()`	= the number of characters in `s`
`int`	`s.indexOf(n)`	= the index within `s` of the first occurrence of `String n` (`-1` if none)
`String`	`s.substring(h,k)`	= a String consisting of characters in `s[h..k-1]`, ie. `s[h], s[h+1], ..., s[k-1]`
`String`	`s.substring(h)`	= a String consisting of characters `s[h..s.length()-1]`
`int`	`indexOfFirstVowel(s)`	= the index within `s` of the first occurrence of a vowel (s.length() if none). Vowels are 'a', 'e', 'i', 'o', 'u', and 'y' when it is not in position 0.

Cornell net id _____ **Name** _____