NAME ______________________________________ Cornell Net id __________

CS100J Prelim 1 22 February 2007

Q 0		/02
Q 1		/18
Q 2		/30
Q 3		/10
Q 4		/20
Q 5		/20
Total		/100

This 90-minute exam has 6 questions (numbered 0..5) 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 can tear the pages apart; we have a stapler at the front of the room.

Question 0 (2 points): Write your name and NetID, legibly, at the top of each page.

Question 1 (18 points): Answer the following questions concisely:

(a) 5 pts. What is an argument? A parameter?

(b) 5 pts. What is a local variable? What is its scope?

(c) 5 pts. Explain the three steps in evaluating a new-expression (e.g. `new Time(c,d)`). The previous sentence contains an example of a new-expression, but your answer should explain what *any* new-expression does and not simply the one in the example.

(d) 3 pts. Draw a frame for the call m(2+3, 6) of the following procedure m. We want to see what the frame for the call looks like after the argument values are assigned to the parameters but before the method body is executed.

```
public void m(int x, double y) {
     int z;
     z= x + y;
}
```

Question 2 (30 points). Use the back of the next page to answer this question.

(a) 8 pts. At the bottom of the page are two class definitions. Draw one folder (object, instance) of each. Do not show the partition for superclass `Object`.

(b) 12 pts. Write a class definition for a class `PercussionInstrument` that

1. Is a subclass of `MusicInstrument`;
2. Has suitable specifications on methods and definitions on fields;
3. Has a field `numDrums`, which is the number of drums in this percussion instrument;
4. Has a constructor with the name of the instrument and the number of drums as parameters;
5. Overrides function `play()` to return the number of “druuums”, similar to the way function `play` in class `StringInstrument` works.

(c) 10 pts. Suppose we want to keep track of the number of objects of class `MusicInstrument` that have been created. Write the declaration of the variable that will contain this value (this variable will be defined in class `MusicInstrument`) . Then, rewrite the constructors in the class to maintain this variable. (Don't change the constructors on this page; just rewrite them on the back of the next page.)

```
/** An instance represents an
    instrument */
public class MusicInstrument {
  // the instrument’s name
  private String name= null;

  /** Constructor: an instrument with
      name s */
  public MusicInstrument(String s) {
        name= s;
  }

  /** Constructor: an instrument with
      name "" */
  public MusicInstrument() {
        name= "";
    }

   /** = sound this instrument makes*/
    public String play() {
        return “music ”;
    }

    /** = a repr of this instrument */
    public String toString() {
        return "Instrument: " + name;
    }
}
```

```
/** An instance represents a string
instrument with no name */
public class StringInstrument extends
                  MusicInstrument {

  /** number of strings on this
      instrument */
  private int numStrings;

  /** Set the number of Strings on this
      instrument to n */
  public void setNumber(int n) {
      numStrings=  n;
  }

  /** = sound this instrument makes */
  public String play() {
      return super.toString() +
            numStrings + “triiings”;
  }
}
```

Question 3 (10 points). Consider classes `MusicInstrument` and `StringInstrument` of question 2. Write a subclass `Violin` of `StringInstrument` that:

1. Has suitable specifications on methods and definitions on fields;

2. Has a `String` field named manufacturer;

3. Has a constructor with this specification:

```
    /** Constructor: a violin with n strings made by
                     manufacturer s and no name */
    public  Violin(int  n, String s)
```

4. Has a constructor with the specification shown below. The body of this constructor should be a single statement.

```
    /** Constructor: a violin with 6 strings made by
                     manufacturer s and no name*/
    public  Violin(String s)
```

Question 4 (20 points). Below is a definition of class Student. Assume the following three assignment statements are executed:

Student p1= **new** Student(“Bill”, “bk12”);
Student p2= **new** Student(“Bill”, “bk13”);
Student p3= **new** Student(“William”, “bk12”);

```
/** An instance represents a Student*/
public class Student {
  // the student’s name
  private String name;

  // the student’s netid
  private String netid;

  /** Constructor: a Person with
        name n and netid i*/
  public Student(String n, String i)
      {name= n; netid= i; }

  /** = the Student’s name */
  public String getName()
      {return name; }

  /** set the Student’s name to n */
  public void setName(String n)
      {name= n; }

  /** = “this Student and s have the
      same netid” */
  public boolean equals(Student p)
      {return netid.equals(p.netid); }

  /** = a representation of this
        Student*/
  public String toString()
      {return netid + ":" + name; }
}
```

(a) What is the value of each of the following four expressions?

p1.equals(p2)

p1.equals(p3)

p1 == p2

p1 == p3

(b) Now consider these statements:

p1= **new** Student(“Bill”, “bk12”);
p2= **new** Student(“Bill”, “bk13”);
p3= p2;
p3.setName(“Jack”);

Below, first draw all three variables. Then execute the four statements —of course, draw any objects that are created during execution.

Question 5: (20 pts) Write function `fix`, which is specified below. You may use the following methods (you may not need them all). This might help you: when you break String s up into pieces, store the pieces in local variables and then use these pieces.

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)
`int`	`s.indexOf(n, k)`	= the index within `s` of the first occurrence of String `n` that begins at or after index `k` (`-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]`

/** = Date s in a more suitable form.

Precondition: s contains a date in the form month/day/year, with each part separated by "/". Examples are: 4/26/39 and 04/005/1939.

The output should be in the form year.month.day. Examples are: 39.26.4 and 1939.04.005.

Each of day, month, and year may be any length. They appear in exactly the same form in the input and output; just their order and the separator are changed. */

public static String fix(String s) {

}