**Cornell net id** _____ **Name** _____

# CS 100J Prelim 3 18 April 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 answering any. Budget your time wisely. Use the back of the pages, if you need more space. We have a stapler at the front of the room, so you can tear the pages apart.

| | |
|---|---|
| 0 _____ | out of 02 |
| 1 _____ | out of 15 |
| 2 _____ | out of 18 |
| 3 _____ | out of 20 |
| 4 _____ | out of 25 |
| 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).

**`Question 1 (15 points).` Recursion.** In assignment A2, you implemented a class Rhino that had, among other things, the fields shown in the partially completed class Rhino shown below. Write recursive function `numberAncestors`, whose specification and heading appear in the class below. Writing a loop is inappropriate and will cause many points to be deducted. Here's what an ancestor is: an ancestor of a rhino is a (non-null) father or mother, or a (non-null) grandfather or grandmother, or …, etc.

Don't write any other methods; just write the body of `numberAncestors`.

```
/** An instance describes a rhino.*/
public class Rhino {
   private String name;       // The name of the rhino.
   private boolean gender; //  = "this rhino is female"
   private Rhino father;      // This rhino's father (null if unknown)
   private Rhino mother;    // This rhino's mother (null if unknown)

   …

   /** = the number of known ancestors of Rhino r.
         Precondition: r is not null  */
   public static int numberAncestors(Rhino r) {




   }
}
```

**Cornell net id** _____ **Name** _____

**Question 2 (18 points). GUIs.** Consider class GUI on the right, which, for the purposes of this question, has few comments.

(a) In the evaluation of:

```
new GUI(4);
```

how many instances of class `Box` are created in the `JFrame`?

(b) In the same new-expression as in (a), how many buttons are placed in the `JFrame`?

(c) Draw the `JFrame` that results from evaluating the expression

```
new GUI(3);
```

If evaluation of this expression requires generating a random number, choose any random number you like, as long as it is one that the program could choose.

Note that the important thing is not the beauty of what you draw or the shape and shading of components but the placement of components and the titles of the buttons.

```
public class GUI extends JFrame {

   private static Random randomGen= new Random();

   public GUI(int n){
      super("quest 2");
      int r= randomGen.nextInt(n);
      int c= randomGen.nextInt(n);
      addButtons(n, r, c);
      pack();    show();
   }

   public void addButtons(int n, int r, int c){
      Box[] box= new Box[n];
      for (int i= 0; i != n; i= i+1){
         // Store in box[i] a box with n buttons
         box[i]= new Box(BoxLayout.X_AXIS);

         for (int j= 0; j != n; j= j+1){
            String s= "(" + i + ", " + j + ")";
            if (i == r && j == c)
               s= "Click";

            // Add new button with title s to box[i]
            box[i].add(new JButton(s));
         }
      }

      Box verticalBox= new Box(BoxLayout.Y_AXIS);
      for (int i= 0; i != n; i= i+1){
         verticalBox.add(box[i]);
      }

      Container cp= getContentPane();
      cp.add(verticalBox);
   }
}
```

**Cornell net id** _____ **Name** _____

**Question 3 (20 points). Arrays and methods.**

A tridiagonal array m is a square array in which, in each row k ($0 \le k < m.length$), all elements are 0 except perhaps elements m[k][k-1], m[k][k], and m[k][k+1] (if they exist).The following matrix is tridiagonal (* is any integer)

```
* * 0 0 0 0
* * * 0 0 0
0 * * * 0 0
0 0 * * * 0
0 0 0 * * *
0 0 0 0 * *
```

Complete method isTridiagonal, whose specification is given below.

```
/** = "array m is tridiagonal".
      Precondition: m is square (number of rows = number of columns). */
public static boolean isTridiagonal(int[][] m) {




}
```

**Cornell net id** _____ **Name** _____

## Question 4 (25 points) Classes

**(a)** In modular arithmetic, integers are kept in a range `0..(modulus-1)`, where `2 ≤ modulus`. This is called "arithmetic mod `modulus`". Examples abound. Minutes are always in the range 0..59 (so the modulus is 60), hours of the day in the range 0..23. In modular arithmetic, the numbers wrap around. For example, the integers mod 5 are, in succession, 0, 1, 2, 3, 4, 0, 1, … —adding 1 to 4 in mod 5 arithmetic yields 0, adding 2 to 4 in mod 5 arithmetic yields 1, etc. Interestingly enough,

```
(b mod modulus) + (c mod modulus) = (b+c mod modulus)
```

Thus, to add two integers in modular arithmetic, just add them and then change the answer to be in the right range.

An instance of class `Mod`, given on the next page, represents an integer in mod `modulus` arithmetic. Note the "class invariant" —the restrictions on fields `i` and `modulus` given as comments. These restrictions must be maintained.

Write the bodies of the constructor (note carefully its specification) and functions add and equals. In writing them, you do not have to check that preconditions mentioned in the specs are met. You may want to use procedure `modularize`, which defines how to change any integer into that integer mod `modulus`.

**(b) Answer the following four questions (they have nothing to do with class Mod).**

**b1.** State the reason for making a class abstract and give an example that shows that you know how to declare an abstract class.

**b2.** State whether the following is true or false: In a constructor of a subclass, the first statement must be a call of a constructor of the superclass. Explain your answer.

**b3.** Suppose a local variable of a method is declared in a loop within the method body. When the method is called, when is the local variable created?

**b4.** When is a static variable created?

**Cornell net id _____ Name _____**

```
/** An instance is an integer in mod "modulus" arithmetic */
public class Mod {
   private int modulus; // The modulus. modulus > 1
   private int i;           // The integer. 0 ≤ i < modulus

   /** Constructor: integer i in mod modulus arithmetic.
       Precondition: modulus > 1. There are no restrictions on i.*/
   public Mod(int i, int modulus) {









   }

   /** Change i, if necessary, so that 0 ≤ i < modulus, keeping the value i mod modulus the same */
   public void modularize() {
      if (0 <= i && i < modulus) return;
      i= i % modulus;
      if (i < 0)  i= i+modulus;
   }

   /** = String rep of this number */
   public String toString() {
      return i + " mod " + modulus;
   }

   /** If  r1 and r2 do not have the same modulus, return null; otherwise return r1 + r2 */
   public static Mod add(Mod r1, Mod r2) {









   }

   /** = "r is a non-null Mod with the same modulus
           and value as this one" */
   public boolean equals(Object ob) {









   }
}
```

**Cornell net id** _____ **Name** _____

**Question 5 (20 points) Known algorithms.** Write a specification and header for function binary search. Your specification may be in terms of pictures, English, mathematical notation, or a mixture of all of them. But it must be a specification of binary search as we have presented the algorithm in class.

Then, develop the body of the function. When developing its body, write the invariant for the loop first. Then develop the loop and initialization using the four loopy questions. An answer that does not have a suitable loop invariant will receive less than half credit.