CS 101 Fall 2005 Midterm 3 Name: _____ Email ID: _____

This exam is open text book but closed-notes, closed-calculator, closed-neighbor, etc. Questions are worth different amounts, so be sure to look over all the questions and plan your time accordingly. Please sign the honor pledge here:

_____

_____

_____

_____

Page 1	____ / 4
Page 2	____ / 20
Page 3	____ / 14
Page 4	____ / 20
Page 5	____ / 16
Page 6	____ / 16
Page 7	____ / 10
Total	**____ / 100**

Note: When an integer type is required use ***`int`****, when a floating-point type is required use* ***`double`****.*

1. (2 points) What section are you in?

   ____ CS 101-E ____ CS 101-2 (lab 7-8:30 PM Thu)

   ____ CS 101-3 (lab 12-1:30 PM Fri) ____ CS 101-4 (lab 2-3:30 PM Fri)

2. (2 points) What is your overall impression of the class so far? Check only one in *each* column.

   ___ Too slow ___ Too easy

   ___ Too fast ___ Too hard

   ___ Just right ___ Just right

3.

___ / 4

(10 points) Given the following code:

```
int i = 2;
int a[] = {1, 4, 2, 5, 7, 2, 3, 4};
int b[] = new int[i];
int[] c;
int d[] = a.clone();
int e[];
e = b;
```

Draw a memory diagram after the above code has executed.

3. (10 points) Evaluate the following expressions and give the value, or write "causes an error":

A. `a[1]` _____

B. `a[a[i]]` _____

C. `a[a.length-1]` _____

D. `a[4.0 / i]` _____

E. `b[1]` _____

F. `b[i]` _____

G. `a[1] < a[i++]` _____

H. `c[0]` _____

I. `d[b.length]` _____

J. `b.toString()` _____

___ / 20

4. (4 points) Consider the following definition:

```
int mystery[][] = { {4,5}, {5,1,3}, {1,2} };
```

Does this code compile? If so, draw the resulting memory diagram; if not, explain why.

5. (10 points) Write a public static method **`average()`** with a return type of **`double`** that takes a single parameter called **`data`**, which is an **`int`** array, and returns the average value of all elements in the array. Note that the average value may be a fraction, so you should be careful not to accidentally truncate it to an **`int`**. You may assume that **`data`** has at least one element.

___ / 14

6. Assume the following code snippet has executed:

```
String [] silverware = {"knife", "fork", "spoon"};
String [] flatware = silverware.clone();
```

A. (4 points) Draw a memory diagram after this code executes.

B. (4 points) Explain in **20 words or less** the difference between a shallow copy and a deep copy.

C. (2 points) Is the code shown above performing a shallow or a deep copy?

D. (6 points) If your answer to (c) is "shallow copy", then write a code snippet which instead makes **`flatware`** a deep copy of **`silverware`**. If your answer to (c) is "deep copy", then write a code snippet that instead makes **`flatware`** a shallow copy of **`silverware`**.

E. (4 points) Draw a memory diagram after your code for (d) executes.

___ / 20

A *stack* is like a vector or an array, in that a lot of elements can be stored in it. In particular, with a stack, elements are always added to the *end* of the stack, and removed from that same end. Thus, you cannot remove any element you want – you can only remove the element at the very end of the stack. The next few questions will have you developing a Stack class to represent such a means of storage. For purposes of this exam, we will assume that the maximum size of a stack is 1000 elements, and that the stack cannot hold more than that. Also, the stack will only contain `int` values. *Note that you must use an array to hold the values in the stack – no Vectors allowed.*

7. (4 points) **Instance variables:** Give the definition for the instance variables needed for this `Stack` class, using the information in the paragraph above. The instance variables can either be initialized here or in the constructor, below.

8. (4 points) **Constructor:** Give the definition for the default constructor of the stack. You can either initialize your instance variables in the constructor, or when they are defined above. The class name is `Stack`.

9. (8 points) **Push:** Adding an element onto the stack is called *pushing* an element onto the stack. Give the definition for the `push()` method. It should take in a single `int` value that will be pushed onto the stack, and does not return a value. You can assume that there is enough space in the array for the element (i.e. the array is not yet filled). Recall that an element is pushed (added) onto the *end* of the stack.

___ / 16

10. (8 points) **Pop:** Removing an element from the stack is called *popping* that element. Give the definition for the `pop()` method. It should return the value removed from the stack. Recall that an element is popped (removed) from the end of the stack.

11. (4 points) **Peek:** Sometimes one may want to examine the top element on the stack (i.e. the last element pushed, which is also the next element to be popped). Give the definition for the **`peek()`** method, which returns that element, but does not remove it from the stack.

12. (4 points) **Emptiness:** An **`empty()`** method should return **`true`** if the stack is empty, and **`false`** otherwise. Give the definition for **`empty()`**.

___ / 16

13. (10 points) Choose **ONE** of the following two questions. You will not get extra credit for doing both questions. Please **clearly** indicate which question you would like us to grade. If this is not clearly indicated, we will choose one at random, **not** grade both and give you the better grade.

Search: One may want to see if the stack contains a given element. Give the definition for the `search()` method, which returns a `boolean` answer of whether the stack contains the passed element or not. Note that because this method is within the `Stack` class, it can access all the elements of the array instance variable directly (you are not restricted to using `push()` and `pop()`).

Printing: The `toString()` method is used to print out the elements of the stack, separated by commas. Give the definition for the `toString()` method. Note that because this method is within the `Stack` class, it can access all the elements of the array instance variable directly (you are not restricted to using `push()` and `pop()`).

___ / 10

This page intentionally left blank

XX / XX