Computer Science 136 Exam 1

Sample exam

Show all work. No credit will be given if necessary steps are not shown or for illegible answers. Partial credit for partial answers. Be clear and concise. Write your name on **each page** of the exam.

You may use one page of handwritten notes (turn this in also). The use of other reference materials or electronic devices is a violation of the honor code. **Good luck!!**

Question	Points Earned	Points Available
1		14
2		10
3		26
4		20
5		12
Total		82

Your name (please print): ____________________________

I have neither given nor received aid on this examination:

____

(sign here)

Name: ____________________

1. True/false statements (2 points each). Justify each answer with a sentence or two.

a. *Free question from 2/28.* Two instances of `class Association` in the `structure` package are equal if and only if their `key`s are equal, regardless of their `value`s.

b. An instance variable declared as `protected` can be accessed by any method of the class in which it is declared.

c. A binary search can locate a value in a sorted `Vector` in $O(\log n)$ time.

d. A binary search can locate a value in a sorted `SinglyLinkedList` in $O(\log n)$ time.

e. A method with no precondition should return with its postcondition true every time it is called.

f. Elements stored in an instance of `class Vector` can be of any Java data type.

g. Instance variables are specified in an interface file.

Name: ___________________

2. Consider the following Java program, which should look very familiar. (10 points)

```
class Container {
    protected int count;
    protected static int staticCount;

    public Container(int initial) {
        count = initial;
        staticCount = initial;
    }

    public void setValue(int value) {
        count = value;
        staticCount = value;
    }

    public int getCount() {
        return count;
    }

    public int getStaticCount() {
        return staticCount;
    }
}

class WhatsStatic {

    public static void main(String[] args) {
        Container c1 = new Container(17);
        System.out.println("c1 count=" +c1.getCount()+
                           ", staticCount=" + c1.getStaticCount());

        Container c2 = new Container(23);
        System.out.println("c1 count=" +c1.getCount()+
                           ", staticCount=" + c1.getStaticCount());
        System.out.println("c2 count=" +c2.getCount()+
                           ", staticCount=" + c2.getStaticCount());

        c1.setValue(99);
        System.out.println("c1 count=" +c1.getCount()+
                           ", staticCount=" + c1.getStaticCount());
        System.out.println("c2 count=" +c2.getCount()+
                           ", staticCount=" + c2.getStaticCount());

        c2.setValue(77);
        System.out.println("c1 count=" +c1.getCount()+
                           ", staticCount=" + c1.getStaticCount());
        System.out.println("c2 count=" +c2.getCount()+
                           ", staticCount=" + c2.getStaticCount());
    }
}
```

a. What will the output be when the program is run (`java WhatsStatic`)? Assume no exceptions occur. (4 points)

b. What memory is allocated for `Container`s `c1` and `c2` at the time the line `c1.setValue(99)` is executed? Show any existing local variables and instance variables. (6 points)

Name: ___________________

3. (26 points) In this problem you are to design a Java interface and class for a data structure which represents sets of characters. As usual for sets, no repeated elements are allowed. Thus, the collection `'a'`, `'e'`, `'i'`, `'o'`, `'u'` is a legal set, but `'a'`, `'e'`, `'a'` is not. This data structure will have two methods:

- `insert(char newChar)` adds `newChar` to the set.
- `contains(char findChar)` returns a boolean value indicating if `findChar` is an element of the set.

a. Write a legal Java `interface` called `CharSetInterface` for this data structure. Include preconditions and postconditions for the methods. (6 points)

Name: ___________________

b. Suppose we decide to implement `CharSetInterface` by a class in which a singly-linked list holds all of the elements. Write the definition of this class. This should be a full and legal Java class definition *with all method bodies filled in.* Don't forget to declare instance variables, include a constructor, and use qualifiers such as `public` and `protected` when appropriate. You need not repeat your pre- and post- conditions from part a. Please call your class `CharSet`. (10 points)

Name: ____________________

c. If `CharSet` is implemented as in part b, what would the worst-case time complexity be for the insert operation when the set has n elements? (Use "Big O" notation.) (4 points)

d. Suppose we design an alternative implementation in which the set is represented by an array of `booleans` called `rep` with subscripts ranging from 0 to 65535 (these represent the codes for all of the characters representable in Unicode). For example, the Unicode for `'a'` is 97, so `'a'` is in the set if and only if `rep[97]` is `true`. What is the worse-case complexity of insert with this representation? (You may assume there is a constant-time function available which computes the Unicode value of a given character.) (6 points)

Name: ___________________

4. (20 points) Consider the following class, `ReversibleList`, that extends the `SinglyLinkedList` class by adding a method for reversing the list.

```
public class ReversibleList extends SinglyLinkedList {

  public ReversibleList() {
    super();
  }

  public void reverse() {
    // Pre: list is not empty.
    // Post: list is reversed.
    Assert.pre(!isEmpty(), "Cannot reverse an empty list");
    head = recReverse(head);
  }

  private static SinglyLinkedListElement recReverse(SinglyLinkedListElement current) {
    // Pre: current is not null.
    // Post: list headed by current is reversed;
    //       head elemet of reversed list is returned.
    if (current.next() == null)
      return current;
    else {
      SinglyLinkedListElement newHead = recReverse(current.next());
      current.next().setNext(current);
      current.setNext(null);
      return newHead;
    }
  }
}
```

Name: ___________________

a. Prove by induction that `reverse()` behaves correctly. (Hint: focus on `recReverse(current)`) (8 points).

b. What is the running time of `reverse()` (2 points)?

c. Prove using mathematical induction that your answer to part b is correct. (10 points)

Name: ____________________

5. Growth of functions. Using “Big O” notation, give the rate of growth for each of these functions. Justify your answers. (3 points each, 12 total)

a. $f(x) = x^2 + 17x + 2001$

b. $f(x) = \cos(x^4 + \log x)$

c. $f(x) = 7x$ when x is odd, $f(x) = \frac{x}{7}$ when x is even.

d. $f(x) = 5x^3$ for $x < 23$, $f(x) = 37$ otherwise.