Midterm Practice Exam
CS/COE 445

1. What does a reference variable hold?
    - The address of the object it refers to.
    - The object.
    - It can hold any value.

2. Which of the following are examples of polymorphism?
    - Child c = new Parent(); //assume Child inherits from Parent
    - Interface i = new ExampleClass(); // assume ExampleClass implements Interface
    - ExampleClass ex = new ExampleClass();

3. What is a singly linked list?
    - A linked list where each node points to only one of its neighbors
    - A linked list with just one value stored in the list
    - A linked list with only one reference to the list

4. The top of the stack always represents:
    - The last item put into the stack
    - The first item put into the stack
    - The item placed at index 0 in the stack
    - It is random; there is no order to items in the stack

5. True or False: The algorithm used to solve the 8 Quees Problem can also be used to solve the 7 Queens Problem.

6. True or False: All of the sorting algorithms we've seen require that each element in the list be unique.

7. Order the following runtimes from smallest to largest , starting with 1 for the smallest. If there are any ties, mark the tied runtimes with the same number.
    _____ O(N2)
    _____ O(1)
    _____ O(log(N))
    _____ O(N)
    _____ O(N + N/2)

8. True or False: Divide and conquer algorithms require two (or more) recursive calls.

9. Why are interfaces useful for data structures?

10. When writing a class to implement an ADT, you will be writing methods to implement the ADT's operations. What should you consider when implementing an operation such that it meets specifications of the ADT?

11. What is contiguous memory? What is non-contiguous memory? How do they differ?

12. Why is algorithm analysis useful?

13. Assume you are analyzing an algorithm and derive the following function for counting its key operation. What would be its Big-O runtime?

$$f(n)=1+2+3+...+n$$

14. Assume you are analyzing an algorithm and derive the following function for counting its key operation. What would be its Big-O runtime?

$$f(n)=\log(n^2)+n$$

15. What is the worst-case runtime of the following method?

```
public double sum(double [] array) {
    double total = 0;
    for (int i=0; i < array.length; i++) {
        total += array[i];
    }
    return total;
}
```

16. What is the worst-case runtime of the add method shown below. Assume `bag` is a class-level array.

```
public boolean add(T newEntry) {
    checkInitialization();
    if (isArrayFull()) {
        int newLength = 2 * bag.length;
        bag = Arrays.copyOf(bag, newLength);
    }

    bag[numberOfEntries] = newEntry;
    numberOfEntries++;
    return true;
}
```

17. Assume you have a stack of expiration dates. These expiration dates are in sorted order (with the closest-to-expire on top). You have a new expiration date that you want to insert into the stack in sorted order. Explain how you would insert it into the stack?

18. Any recursive algorithm can be written as an iterative algorithm. In all cases, the iterative version can use a stack but some iterative versions **must** use a stack. Why would a stack be needed for iterative versions of recursive algorithms?

19. How do Insertion Sort and Shellsort differ? How are they the same?