

Write your answers on the answer sheets provided. You may refer to printouts of any source code you have written.

```
class Dog {
    private String name;
    private double weight;
    public Dog(String name, double weight) {
        this.name= name;
        this.weight= weight;
    }
}
class ListNode {
    int data;
    ListNode next;
}
class MyStack<T> {
    // internal implementation not specified, but does have these methods
    public void push(T elem) {}
    public T pop() {}
    public boolean isEmpty() {}
}
```

1. (4 pts) Expand the acronym FIFO. What data structure is it associated with?
2. (2 pts each) Briefly define each of the following terms (in the context of this course):
 - (a) wrapper class
 - (b) stack
3. Refer to the Dog class defined on the first page of this exam and to the following function and then answer these questions about containers and sorting:

```
public static void sortDogs(Dog [] dogs, int n) {
    int i,j;
    Dog temp;
    for (i=0; i<n-1; i++)
        for (j=i+1; j<n; j++)
            if (dogs[i]<dogs[j]) {
                temp= dogs[i];
                dogs[i]= dogs[j];
                dogs[j]= temp;
            }
}
```

- (a) (2 pts) What is the name of the sort depicted above?
 - (b) (2 pts) Upon completion of the following function, what order would the objects in the `dogs` parameter be placed?
 - (c) (6 pts) Write a standard `.compareTo()` method for the `Dog` class that will compare dogs by comparing their weight.
 - (d) (3 pts) Show how the if-statement in the sort above would be modified to use a standard `.compareTo()` method to cause the sort to list the dogs in order of weight with heaviest weight first.
 - (e) (3 pts) Suppose we want to store our `Dog` objects in a generic container that provides a `sort()` method of its own. What additional changes would we need to make to the `Dog` class to make that possible.
4. (8 pts) Write a `BlackHole` class that has an attribute called `core` that is the head of a linked-list of objects. The class should have only a single method called `engulf` that will (like an actual black hole) accept the an object and store it (but not provide a way to retrieve it). The `BlackHole` should be able to accept any type of object. To create a list you'll need to also write a `BlackHoleNode` class.
5. (6 pts) Show how the following array would be processed if it were partitioned one time using the partition algorithm that accompanies Quicksort.
- `< 64, 75, 140, 40, 132, 70, 83, 37, 50, 80, 100 >`
6. (4 pts) Using words, give a recursive description of how a binary search finds an element in a sorted list.
7. Use the `ListNode` class given at the beginning of the exam to to answer these questions.
- (a) (8 pts) Write a method called `removeLast` that will accept the head of a linked list (i.e., a `ListNode`) as a parameter and will traverse to the end of the list and remove the last element of the list.
 - (b) (4 pts) Modify the `ListNode` class to allow the data to be any object that implements the `Comparable` interface (rather than storing only integer data).
 - (c) (6 pts) Write a method called `isOrdered` that will accept the head of a linked list as a parameter and will return true if the elements in the list are arranged from smallest to largest. It should return false otherwise. If the list is empty it should return true. The parameter will be the modified version of the `ListNode` you created in the previous question (7b).
8. Answer these questions about stacks and the `MyStack` class given above.
- (a) (4 pts) Suppose that when a letter is encountered we push it on the stack and when an asterisk is encountered we pop the stack. Show the sequence in which letters would be popped for the following string:

`RM*RAEE**RNIT***IC**LC*U*****MO*0**`

- (b) (4 pts) Contrast an array-based stack implementation with a list-based stack implementation. Give advantages of each over the other.
- (c) (10 pts) Write a method called `matchParens()` that accepts a `String` parameter containing an expression with square brackets and braces (e.g., “[`x+y-z`]+[`a-b`]/`c`]+5-2”). The method will return `true` if the brackets are properly matched and `false` if not.

Assume the existence of a working (generic) `MyStack` class outlined on the first page of this exam. Use the stack to help determine whether or not the expression is valid.

9. Answer these questions about queues.

- (a) (4 pts) Suppose that when a letter is encountered we put it in the queue and when an asterisk is encountered we remove from the queue. Show the sequence in which letters would be removed for the following string:

`RM*RAEE**RNIT***IC***LC*U*****MO*O**`

- (b) (4 pts) Briefly describe (using words) how to provide an efficient array-based solution to implementing a queue.

10. (4 pts) Which homework assignment in this course was your favorite? Why?

11. (4 pts) Which topic in this course was your favorite?