Please answer the following questions, each of which is worth 20 points.

1. This question deals with object-oriented design principles.
   (a) Define the term *polymorphism*.
   (b) Consider the following Java classes:

   ```java
   public class Animal extends Object {
       Animal() { /* null constructor */ }
       public void printMe() { System.out.println("I’m an animal."); }
   }
   public class Dog extends Animal {
       Dog() { /* null constructor */ }
       public void printMe() { System.out.println("I’m a dog."); }
   }
   public class Poodle extends Dog {
       Poodle() { /* null constructor */ }
       public void printMe() { System.out.println("I’m a poodle."); }
   }
   public class Fluffy extends Poodle {
       Fluffy() { /* null constructor */ }
       public void printMe() { System.out.println("I’m Fluffy."); }
       public static void main(String[] args) {
           Animal pig = new Animal();
           Dog stinky = new Poodle();
           Object obj = new Fluffy();
           pig.printMe();
           stinky.printMe();
           ((Animal) obj).printMe();
           obj = pig;
           ((Animal) obj).printMe();
       }
   }
   ```

   What is the output from calling the main() method of the Fluffy class?
2. This question deals with algorithm analysis.
   
   (a) Show that the function \( f(n) = 3n \log n + 6n \) is \( O(n \log n) \).
   
   (b) Suppose that a certain Professor Amongus implements the \texttt{size()} method for a sequence \( S \) in the following way:
       
       Let \( \texttt{enum} = S \texttt{.elements()} \).
       
       Let \( c = 0 \).
       
       \textbf{while} \( \texttt{enum} \texttt{.hasMoreElements()} \) \textbf{do}
       
       Let \( c = c + 1 \).
       
       Let \( x = \texttt{enum} \texttt{.nextElement()} \).
       
       \textbf{end while}
       
       Characterize using the big-Oh notation the running time of the Professor's method in terms of \( n \), the number of elements in \( S \).
   
   (c) Assuming this implementation of the \texttt{size()} method, characterize using the big-Oh notation the running time of the following algorithm:
       
       Let \( B \) be an array of \( n \) integers, each of which is initially 0.
       
       Let \( S \) be an \( n \) element sequence.
       
       \textbf{for} \( i = 0 \) to \( n - 1 \) \textbf{do}
       
       Let \( B[i] = S \texttt{.size()} \).
       
       \textbf{end for}

3. This question deals with positions.
   
   (a) Define what is meant by the \textit{position} of an object in a sequence.
   
   (b) Describe two different implementations of a position in a sequence.

4. What is the output of the following code fragment of operations on a sequence, \( s \) (assume that each variable besides “\( s \)” and “\( p \)” is an Integer object representing the number it has as its name):

   ```java
   Position p = s.insertFirst(one);
   s.insertLast(two);
   s.insertAfter(p,three);
   s.insertAfter(p,four);
   System.out.println(((Integer) s.removeLast()).toString());
   s.insertAtRank(0,five);
   System.out.println(((Integer) s.removeAtRank(s.size()-1)).toString());
   System.out.println(((Integer) s.before(positionAtRank(1))).toString());
   ```

5. This question deals with heaps.
   
   (a) What is a heap?
   
   (b) Draw an example of a heap containing 10 integer objects.
   
   (c) Explain why a heap holding \( n \) elements has height \( O(\log n) \).
6. This question deals with AVL-trees.
   (a) What is an AVL-tree?
   (b) Draw an example of an AVL-tree $T$ with 6 internal nodes, each storing an even-numbered integer object, such that $T$ has at least one node $v$ with only one internal-node child $w$.
   (c) Draw the AVL-tree that results from your inserting an odd-numbered integer object in $T$ to initially be a child of $w$.

7. This question deals with hashing.
   (a) What is hashing?
   (b) Draw a representation of a 5-celled hash table $B$ and its contents after we use the hash function
   $$h(x) = 2x + 1 \mod 5$$
   to insert the elements in the set $\{10, 11, 3, 5, 4, 7, 16, 22\}$ into $B$, assuming we handle collisions with the chaining method.

8. This question deals with sorting.
   (a) Briefly describe the merge-sort algorithm.
   (b) What is the running time of the merge-sort algorithm? (You don’t need to prove this.)

9. This question deals with graphs.
   (a) Draw a connected graph that has 8 vertices, numbered 1, 2, $\ldots$, 8, and has 16 edges.
   (b) Draw an adjacency matrix representation of the graph you just drew.
   (c) Draw the edges of a depth-first spanning tree in the graph you drew for part (a), starting from vertex 1.
   (d) Draw the edges of a breadth-first spanning tree in the graph you drew for part (a), starting from vertex 1.

10. Suppose you are given an unordered sequence $S$ of $n$ integer objects in the range $[0, n - 1]$. Briefly sketch an $O(n)$ time method for determining if $S$ contains two elements that are equal to each other.