PLEASE WRITE YOUR NAME AND ANSWERS ON ALL QUESTION SHEETS. You may use the backs of the question sheets to continue your answers. You may also use the blank sheet after the last question to further continue your answers. Scrap paper is available. GOOD LUCK!!
1 (25 points) Consider the following two classes, which are simplified versions of the ones we studied in class.

```java
public abstract class GeometricObject {
    protected String color = "blue";
    public abstract double area();
}
```

```java
public class Point extends GeometricObject {
    protected double x, y;
    public Point(double x, double y) {
        this.x = x;
        this.y = y;
    }
    public double distTo(Point p) {
        return Math.sqrt(Math.pow(this.x-p.x,2) + Math.pow(this.y-p.y,2));
    }
    public double area() { return 0; }
}
```

Write `public class Line extends GeometricObject` with the following components.

i  `Line` has two data fields, both `Points`.

ii  `Line` has two constructors. The main constructor accepts two `Points` as parameters. The other constructor accepts one `Point` as parameter and uses the origin (0.0,0.0) as the other point.

iii  `Line` has an instance method `length()` that returns the length of the line.

iv  `Line` has a class method `longest()` that returns the longest line constructed so far.

v  There is another method that must be present for `Line` to compile. Name and implement this method.
2 (10 points). The tree on the right uses squares for interior nodes and circles for leaves. In what order are the nodes (interior and leaves) visited in a PREorder traversal?

3 (15 points). What is the output when the following `main()` is run?

```java
public static void main (String[] args) {
    int[][] matrix = { {11,12,13}, {21,22,23}, {31,32,33} };
    for (int i=0; i<matrix.length; i++) {
        for (int j=0; j<matrix[i].length; j++)
            System.out.printf("%d ", matrix[i][j]);
        System.out.printf("\n");
    }
    for (int i=0; i<matrix.length; i++) {
        for (int j=0; j<i; j++) {
            int t = matrix[i][j];
            matrix[i][j] = matrix[j][i];
            matrix[j][i] = t;
        }
    }
    for (int i=0; i<matrix.length; i++) {
        for (int j=0; j<matrix[i].length; j++)
            System.out.printf("%d ", matrix[i][j]);
        System.out.printf("\n");
    }
}
```
4 (25 points). Consider the following code which is the skeleton of a class entitled Prob4.

```java
public class Prob4 {
    String[] s;
    Prob4(String[] s) {}
    public boolean moreCharThan (Prob4 p, char c) {}
}
```

The only data field is an array of strings and NO additional data fields are required.

There is one constructor, which has an array of strings as parameter. The constructor create a Prob4 object initialized to contain the same strings as in the parameter. **You must write the body of this constructor.**

There is one boolean instance method entitled moreCharThan(), which takes two parameters, a Prob4 and a char. moreCharThan() returns true if the instance Prob4 contains more copies of the char parameter than does the Prob4 parameter. It returns false otherwise. **You must write the body of this method.**

For example, assume the Prob4 object prob4 consists of a string array containing the two strings "string1" and "string2". Also assume pr1 is the Prob4 object consisting of a string array containing the three strings "xy", "tss", and "ii". Then prob4.moreCharThan(pr1,'s') returns false since the instance contains only 2 occurrences of s, while the parameter contains 3. Also prob4.moreCharThan(pr1,'n') returns true.

You might find useful the standard String instance method charAt(int index), which returns the char at the specified index in the instance String.
5 (25 points). Write the method `int[] merge (int[]a, int[]b)`

The two input arrays are each sorted in increasing order. Your method is to return an `int` array of length equal to the sum of the two parameter lengths. This output array is to contain all the elements of the two input arrays merged together in sorted order.

For example if the first input array is `{1, 3, 5, 5, 23}` and the second is `int{1, 2, 5, 8, 20}`, then the array returned is `{1, 1, 2, 3, 5, 5, 5, 8, 20, 23}`.

You may assume both inputs are correctly sorted and do not have to check that this is so.
6 (10 Points). I initially forgot to include sample fill-in questions. Here are some from the post-midterm material. See the practice and real midterm for sample fill-ins on the early material. (I realize this makes the practice final 110 points.)

Fill in the blanks with the appropriate term or phrase.

i. The fact that a variable of a supertype can always refer to an object of any of its subtypes is the principle of _________________.

ii. If a base class and one of its derived classes each define a method with the same name and the same signature, we say the method name is _________________.

iii. In java a 2-dimensional array is really a _________________.

iv. o limit visibility of a data field to the class in which it is defined, we declare it to have ________________ visibility.

v. We can apply the method to_string() to any object because to_string() is defined _________________.
