1. Given the following javadoc specification:
   static int add(int a, int b)
   Returns the sum of two int values.

   a. [5 points] Show how to use the add() function to add two integer variables x and y.
   int z = add(x, y);

   b. [5 points] Show how to use the add() function to add three integer variables x, y, and z.
   int w = add(x, add(y, z));

2. [10 points] Write a method called square() that takes a double parameter (call it x) and returns the value $x^2$.

   double square(double x) {
       return x * x;
   }
3. [10 points] Write a method called `issqrt()` that takes two double parameters (call them `x` and `s`) and returns `true` or `false` depending upon whether or not `s` is the square root of `x`. Do not call any other methods in your method.

```java
boolean issqrt(double x, double s) {
    return (x == s * s);
}
```

4. [10 points] Given the following numbered lines of code, write the numbers of the lines that constitute the scope of each variable.

```java
class Bar {
    public static void main(String[] args) {
        int i = 1;
        for(int j=0; j<100; j++) {
            i = i+j;
        }
        { int k;
            k = foo(i);
        }
        System.out.println(k);
    }
    public static double foo(int m) {
        double n = 1/m;
        return n;
    }
}
```

- `i`: 4-15
- `j`: 6-8
- `k`: 11-13
- `m`: 17-20
- `n`: 18-20

4. [10 points] What does this program print out?

```java
class TestProgram{
    public static void main(String[] args){
        int a = 1, b = 2;
        System.out.println(a);
        System.out.println(b);
        a = foo(a, b);
        System.out.println(a);
        System.out.println(b);
    }
    static int foo(int x, int y) {
        x = 3; y = 4;
        System.out.println(x);
        System.out.println(y);
        return y;
    }
}
```
5. Implement a method called \( t \cdot n(\ ) \) that takes an integer parameter \( n \) and returns \( 2^n = 2 \times 2 \times 2 \times 2 \dots \times 2 \) \( (n \) times).

a) [10 points] Write a non-recursive implementation of \( t \cdot n(\) \) using a for loop.

```c
int tn(int n) {
    int product = 1;
    for(int i = 0; i < n; i++)
        product = product * 2;
    return product;
}
```

b) [10 points] Write a recursive implementation of the same function. Recall that \( 2^n = 2 \times 2^{n-1} \) and \( 2^0 = 1 \).

```c
int tn(int n) {
    if(n <= 0)
        return 1;
    else
        return 2 * tn(n-1);
}
```
6. Arrays
   a) [5 points] Declare and create storage for an array of 24 ints called `hourlytemps`

   ```java
   int hourlytemps = new int[24];
   ```

   b) [10 points] Write a method called `smooth()` that takes an array of ints and finds the average of the numbers in the array, then adds 1 to each array element with a value below the average and subtracts 1 from each array element with a value above the average.

   ```java
   void smooth(int[] a) {
     int sum = 0; double average;
     for(int i = 0; i < a.length; i++)
       sum += a[i];
     average = (double)sum/(double)a.length;
     for(int i = 0; i < a.length; i++)
       if(a[i] > average) a[i]--;
       else if(a[i] < average) a[i]++;
   }
   ```

   c) [5 points] Show how you would call `smooth()` with the `hourlytemps` array you created as a parameter

   ```java
   smooth(hourlytemps);
   ```
7. [10 points] Write a method called max() that takes two integer arrays (of the same length) as parameters and returns a third array, where each element of the third array is the larger of the corresponding elements of the two input arrays. Call the arrays a, b, and c.

```java
int[] max(int[] a, int[] b) {
    int[] c = new int[a.length];
    for(int i = 0; i < a.length; i++)
        if(a[i] > b[i])  c[i] = a[i];
        else c[i] = b[i];
    return c;
}
```

8. [10 points] Write a method called multtable() that creates and returns a 2D array of ints containing the multiplication table from 0 to 9 where the (i, j)th element contains i times j.

```java
int[][] multtable() {
    int[][] mt = new int[10][10];
    for(int i = 0; i < mt.length; i++)
        for(int j = 0; j < mt[i].length; j++)
            mt[i][j] = i * j;
    return mt;
}
```
Extra credit. [10 points] Write a method called `average()` that takes as input a 2D array of doubles and replaces each element of the array with the average of the 9 pixel square around that point in the array.

```java
void average(double[][] a) {
    double avg;

    double[][] b = a.clone();

    for(int i = 0; i < a.length; i++)
        for(int j = 0; j < a[i].length; j++)
            a[i][j] = calcavg(b, i, j);
}

double calcavg(double[][] a, int i, int j) {
    double sum = 0;

    sum += a[i][j];
    if(i > 0 && j > 0) sum += a[i-1][j-1];
    if(i > 0) sum += a[i-1][j];
    if(i > 0 && j < a[i].length-1) sum += a[i-1][j+1];
    if(j > 0) sum += a[i][j-1];
    if(j < a[i].length-1) sum += a[i][j+1];
    if(i < a.length-1 && j > 0) sum += a[i+1][j-1];
    if(i < a.length) sum += a[i+1][j];
    if(i < a.length-1 && j < a[i].length-1) sum += a[i+1][j+1];

    return sum/9.0;
}
```