Question 1 ................................................................. (15 points)

(a) [5 points] Write a complete Java program that computes and prints the area of a circle with radius 5.

(b) [5 points] Show how you would modify the program above to prompt the user for a radius on the command line, and then compute and print the area of a circle for that radius. Do not rewrite the entire program. Just indicate which line(s) you would replace, and what you would replace them with.

(c) [5 points] Show how you would modify the program above to exit gracefully in the exceptional cases where (a) the user enters something that is not a number or (b) enters a negative radius. Again, do not rewrite the entire program.
Consider the method `foo` defined by:

```java
public static void foo(double[] a) {
    double b = 0.0;
    for (int i=0; i<a.length; ++i) {
        if (b<a[i])
            b = a[i];
    }
    return b;
}
```

(a) [2 points] This code will not compile without error. Fix it.

(b) [3 points] What does the following program fragment print?
```java
double[] a1 = { 1.0, 3.0, 2.0};
double[] a2 = { 1.0, 0.0,-1.0};
double[] a3 = {-1.0,-3.0,-2.0};
System.out.println(foo(a1));
System.out.println(foo(a2));
System.out.println(foo(a3));
```

(c) [3 points] Rename the method `foo` to better describe what it is supposed to do.

(d) [2 points] The three tests cases printed above reveal a likely bug. (It’s a common error.) Modify the method so that it does what is likely intended.
Question 3 ........................................................................................................... (15 points)

(a) [10 points] Implement the following method as specified:

```java
/**
 * while n is greater than 1 {
 *  prints the value of n followed by a blank space
 *  if n is even,
 *   replaces n with n/2
 *  otherwise if n is odd,
 *   replaces n with 3*n+1
 * }
 * prints the value of n followed by a newline
 */
public static void goofy(int n) {
    // implementation goes here
}
```

(b) [5 points] Assuming that you have implemented this method as specified, what sequence of numbers is printed by the following program fragment?

```java
for (int i=1; i<=4; ++i)
    goofy(i);
```

Question 4 ........................................................................................................... (5 points)

(a) [2 points] Given two Java floats `x` and `y`, how would you determine if they are equal?

(b) [3 points] Given two Java Strings `s` and `t`, how would you determine if they are equal?
Question 5 ................................................................. (10 points)

Complete the following methods:

(a) [5 points]
    /**
     * Returns a new array that contains the elements in two arrays.
     * Specifically, for a specified array x with m floats and array
     * y with n floats, this method returns an array with m+n floats,
     * with the elements in x followed by those in y.
     */
    public static float[] combine(float[] x, float[] y) {
    }

(b) [5 points]
    /**
     * Returns the number of floats in an array of array of floats.
     * (Assumes that the lengths of the arrays of floats may vary.)
     */
    public static int count(float[][] x) {
    }

Question 6 ................................................................. (5 points)

(a) [1 point] How many bits in a Java byte?
(b) [1 point] How many bytes in a Java int?
(c) [1 point] How many bytes in a Java short?
(d) [1 point] How many bytes in a Java float?
(e) [1 point] How many bytes in a Java double?
public class Point {

    /**
     * Constructs a point with specified (x,y) coordinates.
     * @param x the x coordinate.
     * @param y the y coordinate.
     */
    public Point(double x, double y) {
    }

    /**
     * Moves this point by the specified amounts.
     * @param dx amount to add to the x coordinate of this point.
     * @param dy amount to add to the y coordinate of this point.
     */
    public void move(double dx, double dy) {
    }

    /**
     * Returns the distance between this point and another point.
     * @param p the other point.
     * @return the distance.
     */
    public double distanceTo(Point p) {
    }

    /**
     * Returns a new point with the same coordinates as this point.
     * @return the new point.
     */
    public Point clone() {
    }
}
Question 8 .......................................................... (8 points)
Using every method in the class Point defined above, ...
(a) [2 points] Construct a Point p with coordinates (2, 1).
(b) [2 points] Get a copy Point q of the Point p.
(c) [2 points] Move q a distance of 3 in any direction.
(d) [2 points] Print the distance between p and q.

Question 9 .......................................................... (12 points)
Consider a binary file with exactly 100,000 floats that represent an image.
(a) [2 points] Why would we typically not store this image as a text file?
(b) [2 points] Augment the javadoc comments and declaration of the method readImage below to include any extra information you would need to extract the image from the file.
(c) [8 points] Complete the method readImage as specified.

```java
/**
 * Returns the image stored in the specified file.
 * @param fileName name of the file containing the image.
 * @return array of arrays of floats containing the image.
 */
static float[][] readImage(String fileName, ) {
    // implementation
}
```