1. What is the output from the following code fragment:
   ```java
   for (int i = 0; i < 100; i++) {
       if (i < 10 || i % 2 == 0) {
           continue;
       }
       if (i == 17) {
           break;
       }
   }
   System.out.println(i);
   ```

2. Suppose there is an interface called `Edible` which is implemented by a class called `Apple`.
   a. Which of the following code fragments are valid:
      i. `Edible x = new Edible();`  **VALID / INVALID**
      ii. `Edible x = new Apple();`  **VALID / INVALID**
      iii. `Apple x = new Apple();`  **VALID / INVALID**
      iv. `Apple x = new Edible();`  **VALID / INVALID**
   b. Every method in the Edible interface must be implemented by the Apple class. **TRUE / FALSE**
   c. Every method in the Apple class must be listed in the Edible interface. **TRUE / FALSE**

3. Consider the following code fragment:
   ```java
   int[] a = {7, 9, 26};
   f(a[1]);
   g(a);
   ```
   a. Is it possible that the call to `f` will modify an element in the array `a`? **YES / NO**
   b. Is it possible that the call to `g` will modify an element in the array, `a`? **YES / NO**

4. Write a switch statement equivalent to the following structured if. You may assume that the variables `x` and `z` are declared of type `int` and that `x` has already been initialized to some value.
   ```java
   if (x > 5) {
       z = 22;
   } else if (x == 2) {
       z = 50;
   } else if (x > 0) {
       z = 17;
   } else {
       z = 22;
   }
   ```

5. Re-write the following code fragment so that it is just one statement. (Hint: Use the ternary `?:` operator.)
   ```java
   if (x <= 4) {
       y = x*2;
   } else {
       y = 9-x;
   }
   ```

6. Implement a main that will create one two dimensional array of char to hold stars (and anything else needed) the shape of a triangle. It will then print it twice so that it is once going one way and he second time going the other way. The int passed as the parameter tells the number of stars on the first line. Each successive line is ½ the previous. (The two shown here would be printed one after the other vertically – not horizontally.

```
**********          **********
*****                    *****
**                          **
*                            *
*                          *
*                        *
*                    *
*                *
*            *
*        *
*    *
*  *
* *
```
7. Assume that there is a mutable class called Dog, which is equipped with a copy constructor and proper equals method. Consider the class below, which is only partially shown.

```java
public class Kennel {
    private Dog[] dogs;
    public Dog[] getterOne() {
        Dog[] copy = new Dog[dogs.length];
        for (int i = 0; i < dogs.length; i++) {
            copy[i] = new Dog(dogs[i]);
        }
        return copy;
    }
    public Dog[] getterTwo() {
        return dogs;
    }
    public Dog[] getterThree() {
        Dog[] copy = new Dog[dogs.length];
        for (int i = 0; i < dogs.length; i++) {
            copy[i] = dogs[i];
        }
        return copy;
    }
}
```

a. What kind of copy is made in getterOne?  REFERENCE / SHALLOW / DEEP  (Circle one.)
b. What kind of copy is made in getterTwo?  REFERENCE / SHALLOW / DEEP  (Circle one.)
c. What kind of copy is made in getterThree?  REFERENCE / SHALLOW / DEEP  (Circle one.)
d. Which of these getters would allow someone to modify the contents of a particular dog object that is already in an existing kennel?  getterOne  getterTwo  getterThree  (Circle all that apply.)
e. Which of these getters would allow someone to replace a particular dog object (in an existing kennel) with a different dog object?  getterOne  getterTwo  getterThree  (Circle all that apply.)

8. Write a method that will implement "subset". A subset requires that every element in the first array must also be in the second array. You may assume neither array contains any duplicate entries. You may write a private helper method if you would find it useful (but you must also implement that method if you want it).

```java
public static boolean subset(String[] set1, String[] set2); /* returns true if and only if every element of set1 also appears in set2*/
```

9. Write a method that will implement "proper subset". A proper subset requires that every element in the first array must also be in the second array but there is at least one element of the second array which is not in the first array. You may assume neither array contains any duplicate entries. You may write a private helper method or use the previous method if you would find it useful.

```java
public static boolean properSubset(String[] set1, String[] set2); /* returns true if and only if every element of set1 also appears in set2 but there is at least one member of set2 which is not present in set1 */
```
10. Determine the output for the following program:

```java
public class ExceptionQuestion {
    private static void f(int val) {
        System.out.println("In F");
        int x = 5/val;   // Division by zero could happen
        System.out.println("Done F");
    }
    public static void main(String[] args) {
        try {
            f(1);
            System.out.println("After 1");
            f(0);
            System.out.println("After 0");
        } catch (ArithmeticException e) {
            System.out.println("AE caught");
        } catch (NullPointerException e) {
            System.out.println("NP caught");
        } finally {
            System.out.println("Finally Done");
        }
        System.out.println("After All");
    }
}
```

11. Implement the method that is passed a two dimensional array of ints as its first parameter and a single int as its second parameter. Your method must go through the whole array and remove any elements of the 2nd parameter that appear in the first parameter. It should return the count of how many were removed as the return value of the method. You may assume that each row has at least one value left in that row even after the removes are completed.

For example (assuming all single digit integers and rows are shown horizontally) – if the initial has:
```
2 4 6 3 4
3 4
4 1 4 7 4 8
```
if the second parameter is 4. The method should return the value 5 and the returned array should be:
```
2 6 3
3
1 7 8
```

```java
public static int cleanup(int[][] arr, int val){
    // Implementation
}
```

12. Explain the difference between the String and the StringBuffer and how that difference can be noticed when passing an object of that type as a parameter to a method. (Be sure to include an explanation of when there wouldn't be any difference.)

13. If I have an interface named MyInter and I create an array MyInter[] arr. Explain what objects can be referenced from that array and also what cannot be.

14. When an int variable is created, it is sometimes initialized to 0. Where would the int have to be in order to be initialized to 0? (Circle all that apply.)

   instance data member    element of an arrays    local variable
15. Consider the following class and answer the questions that follow assuming each is a method within the ShoppingList class:

```java
public class Toy {
    private int modelNum;
    (other possible private data here)
    public Toy() { ... }
    public Toy(int modNum) { ... }
    public int getModelNum() { ... }
    (other methods defined here)
}

public class ShoppingList {
    private Toy[] list;
    public ShoppingList() { ... }
    public int getSize() { ... }
    public boolean noDuplicates();
    (other methods defined here)
}
```

a. Implement an instance method for the ShoppingList class called `countModel` which returns an int. The method will take in a single int as its only parameter. It will return the count of the number of Toy objects in the current ShoppingList that have the model number indicated by the int passed as the parameter. (You may assume there is a `getModelNum` getter from in the Toy class which returns the int which is that Toy's model number.)

b. Write a junit test that tests `countModel`. (note: this one obviously is not in the ShoppingList class)

c. Implement the instance method for the ShoppingList class named `noDuplicates` (above) that returns true if and only if there are no duplicates (two or more Toys with the same modelNumber) in your Shopping List.

d. Write a junit test that tests `noDuplicates`. (note: this one obviously is not in the ShoppingList class)

e. Assume you have learned that a certain model number is a dangerous toy. You need to modify your shopping list to remove all items that have that model number. Write an instance method of the ShoppingList class named `removeBad`. It will take an integer as its one and only parameter. It will modify the list of Toys contained in the current object to remove all objects currently in the list that have the model number passed in as the parameter. It will then return the number of Toys that were removed from the list as the return value of the method.

f. Write one more method to help manage your Toy shopping - name this method `getMasterList`. This method is a static method that takes in two arrays as its only two parameters. The first array is a list of model numbers of the toys you need (each is an int) – named `modNums`. The second is a list of quantities needed for each of those Toys (here also, each is an int) - `quantArr`. These two arrays will have the same length so that the 0th item in the `quantArr` tells how many of the 0th item in the `modNum` array are needed, and so on for each item. Your must create a two dimensional array of Toy objects where each row corresponds to one model number where the row has the number of Toy objects (all with that model number) as indicated in the `quantArr`. Return this two dimensional array as the return value of the method.

For example if `modNums` contains 1239, 8272, 1999 and `quantArr` contains 3, 5, 2 the two dimensional array has:

```
<table>
<thead>
<tr>
<th>Toy Model 1239</th>
<th>Toy Model 1239</th>
<th>Toy Model 1239</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toy Model 8272</td>
<td>Toy Model 8272</td>
<td>Toy Model 8272</td>
</tr>
<tr>
<td>Toy Model 1999</td>
<td>Toy Model 1999</td>
<td></td>
</tr>
</tbody>
</table>
```