Example True or False (10 points):

Instructions: Circle either True or False based on the validity of the statement.

1. Java has a virtual machine comprised of a compiler, linker, and run-time environment.
   True False

... There are usually 5-10 questions here ...
Example Multiple Choice (10 points):

Instructions: Circle the letter of the best answer.

1. Which of the following keywords will immediately exit a running function?
   A. break
   B. continue
   C. **return**
   D. goto
   E. None of the above

… There are usually 5-10 questions here ...
Example Short Answer (10 points):
Instructions: In your own words/figures, answer the following as best as possible in one or two sentences.

1. Explain the difference between a float and int data type.

An int stores a numerical whole number.
A float stores a numerical value that has a decimal point.

… There are usually about 5 questions here …
Example Entomology - The Study of Bugs (15 points):

Instructions: Find 3 bugs in the following program (there are more than three):

1) Specify the line number of the bug.
2) Check the box next to the type of error; either "Logic" for logic errors or "Other" for (syntax, type, etc.).
3) Write a short explanation or fix for the error.

Note: There can be more than one bug per line.

```
1 // Stop when the user inputs this many odd numbers
2 public static final int MAX_ODDS=5;
3
4 public static void main(String[] args) {
5    int oddsFound = 0;
6    int numberOfTries = 1;
7    Scanner scanner = new Scanner(System.in);
8
9    // Get user input until user has inputted MAX_ODDS odd numbers
10   while (oddsFound <= MAX_ODDS) {
11      System.out.print("Enter a whole number: ");
12      int number = scanner.nextFloat();
13      numberOfTries += 1;
14      if (num % 2 == 0) {
15         oddsFound = 1;
16      }
17   }
18
19    // Report to user
20   System.out.print("Entered " + numberOfTries + " numbers before maxing at "
21     + MAX_ODDS + " odds.");
22   scanner.close();
23 }
```

<table>
<thead>
<tr>
<th>Bug</th>
<th>Line #</th>
<th>Type of Error</th>
<th>Explanation/Fix</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>[X] Logic</td>
<td>Number of tries should start at 0.</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>[X] Logic</td>
<td>Off-by-one error. Should be oddsFound &lt; MAX_ODDS</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>[ ] Logic</td>
<td>Type mismatch error. Gets a float, but stores it in a int. Should be scanner.nextInt()</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>[ ] Logic</td>
<td>Syntax error. Trying to do assignment when a comparison is expected. Should be a comparison operator such as ==, or !=.</td>
</tr>
<tr>
<td>5</td>
<td>14</td>
<td>[ ] Logic</td>
<td>There is no variable called 'num'. So we have an undefined identifier. 'num' should be 'number'.</td>
</tr>
<tr>
<td>6</td>
<td>14</td>
<td>[X] Logic</td>
<td>Code seems to be trying to check if number is even, but it should be checking if number is odd. It should be either num != 0, or num == 1.</td>
</tr>
<tr>
<td>7</td>
<td>15</td>
<td>[X] Logic</td>
<td>OddsFound will always be set to one, which means we will never reach MAX_ODDS. This should be ‘+=’ or ‘++’ instead of ‘=’.</td>
</tr>
</tbody>
</table>
Example What is the Output (15 points):

Instructions: What is the output if the following code is run?

```java
// Global/class variable answer
static int answer = 21;

private static int printAddition(int x, int y) {
    // Local 'answer' variable
    int answer = x + y;
    System.out.println("printAddition():answer = "+answer);
    return answer;
}

dprivate static int subtraction(int x, int y) {
    return x - y;
}

public static void main(String[] args) {
    System.out.println("main(): answer = "+answer);
    printAddition(5,6);
    System.out.println("main(): answer = "+answer);
    System.out.println("main():subtraction(5, 2) = "+subtraction(5, 2));
    System.out.println("main(): answer = "+answer);
}
```

Answer:

```
main(): answer = 21
printAddition():answer = 11
main(): answer = 21
main():subtraction(5, 2) = 3
main(): answer = 21
```
Example Document the Code (10 points):

Instructions: The following code was written, but the documentation was left out. Analyze the code to determine what is going on. Fill in the missing comments (both the /* */ and // parts) to describe what the code is doing.

```java
/**
 * Determines if <code>num</code> is even or not.
 * @param num is the number to check.
 * @return true if <code>num</code> is even, otherwise false.
 */
public static boolean isEven(int num) {
    return (num%2 == 0);
}

/**
 * Determines the factorial of <code>num</code>. Where the factorial of
 * <code>num</code> is equal to num * (num-1) * (num-2) * … * 1.
 * @param num the number to compute factorial of.
 * @return the factorial of <code>num</code>.
 */
public static int factorial(int num) {
    int factorial=1;
    for (int i = 1; i <= num; i++) {
        factorial*=i;
    }
    return factorial;
}

/**
 * Determines the sum of all factorials from 0! to
 * <code>num</code>!
 * i.e. 0! + 1! + 2! + 3! + … + num!
 * i.e. 1 + 1*2 + 1*2*3 + … + 1*2*…*num
 * @param num the last factorial to compute.
 * @return the sum of the first <code>num</code>+1 factorials starting at 0.
 */
public static int sumFactorial(int num) {
    int sum = 0;
    for (int i=0; i<=num; i++) {
        sum += factorial(i);
    }
    return sum;
}
```
Example Fill in the Code (10 points):

Instructions: Fill in the missing code. Some program functionality/comments have been omitted for brevity.

```java
import java.util.Scanner;

/**
 * A game that sees how many tries it takes someone to guess a number
 */
public class GuessNumber {

/**
 * Main program driver
 */
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    // Default to not being solved
    boolean solved = false;

    // Initialize the number of guesses made
    int numGuesses = 0;

    // Generate a new secret number
    int secretNumber = (int) (Math.random() * Integer.MAX_VALUE);

    // Get guesses from the user until they solved the problem
    while (!solved) {
        // Get next guess
        System.out.print("What is your guess (0-
                        + Integer.MAX_VALUE + ")? ");
        int guess = scanner.nextInt();
        // increment counter
        numGuesses = numGuesses + 1;
        // Did they guess the right number?
        if (guess < secretNumber) {
            System.out.println("Your guess is too low.");
        } else if (guess > secretNumber) {
            System.out.println("Your guess is too high.");
        } else {
            solved = true;
            System.out.println("It took you " + numGuesses
                                + " to guess the secret number;");
        }
    }
    scanner.close();
}
```
Example Write the Code (20 points):

Instructions: In java code, write a function (feel free to write helper functions) "printSequence" that takes the parameter num and then prints numbers out in according to the sample output:

A call to printSequence(4) should print:

1
21
321
4321
4321
321
21
1

Answer:

```java
static void printSequence(int count) {
    // Print rows ascending from 1 to count
    for (int row = 1; row <= count; row++) {
        // Print numbers starting with the row number counting down to 1
        for (int number=row; number >= 1; number--) {
            System.out.print(number);
        }
        System.out.println();
    }

    // Print rows descending from count to 1
    for (int row = count; row >= 1; row--) {
        // Print numbers starting with the row number counting down to 1
        for (int number=row; number >= 1; number--) {
            System.out.print(number);
        }
        System.out.println();
    }
}
```