1. Here are several terms relevant to object-oriented (and particularly Java) programming.

(A) local variable(s)  (B) instance variable(s)  (C) actual argument/parameter
(D) class  (E) state  (F) formal argument/parameter
(G) mutator  (H) constructor  (I) observer
(J) static  (K) object  (L) scope

In each statement below, fill in each blank with the letter identifying the most appropriate term from among those listed above. (Each term should be chosen at least once.)

(i) An object, whose state at any given moment is determined by the values of its ________, is an instance of a ________.

(ii) A(n) ________ is a special kind of method whose purpose is to initialize the ________ of a newly-created ________.

(iii) A(n) ________ method is one whose purpose is to modify the state of a(n) ________.

(iv) A(n) ________ method is one whose purpose is to yield information about the ________ of a(n) ________.

(v) A(n) ________ is used by a method for the purpose of receiving data passed to it by a caller.

(vi) A(n) ________ is used by the caller of a method for the purpose of passing data to that method.

(vii) A(n) ________, which is declared inside the body of a method, comes into existence each time that the method begins execution (as a result of a call) and ceases to exist each time that the method finishes execution.

(viii) A(n) ________, which is declared inside the body of a class but outside any method, comes into existence each time that an object of that class is created, belongs solely to that object, and ceases to exist when the object ceases to exist.

(ix) If a method is declared to have void as its return type, it must not be a(n) ________ method.

(x) The ________ of a variable is that region of a program in which it is legal to refer to that variable.
2. For each statement, circle the response that best completes that statement.

(a) An instance of the TossableCoin class is a(n) ___________.
   a. variable b. class c. method d. object e. all of the above

(b) ___________ is not among the advantages of using methods.
   a. Reduced redundancy b. Improved modularity
c. Improved readability d. Reduced running time

(c) A designated symbol such as + or ! indicating that a particular operation is to be applied to one or two values is called a(n) ___________.
   a. operator b. method c. reserved word d. object

For the next few questions, consider this Java method:

```java
public static void animalFarm(int n) {
    if (n > 0) {
        if (n % 4 == 0) { // n is divisible by 4
            System.out.print("Bear ");
        }
        if (n % 3 == 0) { // n is divisible by 3
            System.out.print("Dog ");
        }
        else {
            System.out.print("Lion ");
        }
        System.out.print("Tiger ");
    }
    else {
        System.out.print("Mouse ");
        if (n % 2 == 0) { // n is divisible by 2
            System.out.print("Owl ");
        }
        else {
            System.out.print("Cat ");
        }
    }
}
```

(d) The call `animalFarm(-3)` would produce __________ as output.
   a. Dog Tiger b. Dog Mouse Cat
c. Cat d. Mouse Cat e. none of above

(e) The call `animalFarm(12)` would produce __________ as output.
   a. Bear Dog Tiger b. Bear Tiger
c. Lion Tiger d. Owl e. none of above

(f) The call `animalFarm(7)` would produce __________ as output.
   a. Bear Dog Lion Tiger b. Tiger Mouse
c. Mouse Cat d. Bear Lion Tiger e. none of above
(g) In Java, if \( \text{str} \) is of type \( \text{String} \), the expression \( \text{str.charAt}(i+2) \) can be evaluated (without a \( \text{StringIndexOutOfBoundsException} \) being thrown) only if the value of \( i \) satisfies __________.

a. \( 0 < i \leq \text{str.length()} \)  
   b. \( 0 \leq i < \text{str.length()} \)  
   c. \( -2 \leq i < \text{str.length()} \)  
   d. \( -2 \leq i < \text{str.length()} - 2 \)

(h) The Java statement if \( (k == m) \) { return false; } else { return true; } would best be written as __________.

a. if \( (k ! = m) \) { return true; } else { return false; }
   b. return \( k != m \);
   c. if \( (k == m) \) { return 0; } else { return 1; }
   d. return \( (k != m) == true \);

For the next several questions, consider this Java method:

```java
public static boolean strEquals(String s, String t) {
    boolean result = false;
    if (s.length() == t.length()) {
        int i = 0;
        while (i != s.length() && s.charAt(i) == t.charAt(i)) {
            i = i+1;
        }
        result = i == s.length();
    }
    return result;
}
```

(i) In the method above, each of \text{public} and \text{static} is a(n) __________.

a. reserved word  
   b. variable  
   c. expression  
   d. argument

(j) In the method above, \( i \) is a(n) __________.

a. actual argument  
   b. formal argument  
   c. local variable  
   d. instance variable

(k) In the method above, \( s \) is a(n) __________.

a. actual argument  
   b. formal argument  
   c. local variable  
   d. instance variable

(l) In the method above, \( i != s.\text{length()} \) is a(n) __________.

a. boolean expression  
   b. arithmetic expression  
   c. assignment statement  
   d. method call

(m) In the method above, we can be sure that __________ is true when the loop terminates.

a. \( i == s.\text{length()} && s.\text{charAt}(i) == t.\text{charAt}(i) \)
   b. \( i == s.\text{length()} || s.\text{charAt}(i) != t.\text{charAt}(i) \)
   c. \( s.\text{charAt}(i) != t.\text{charAt}(i) \)
   d. \( i != s.\text{length()} \)
   e. none of the above are certain to be true upon termination of the loop
3. Complete the following method so that it satisfies its specification, as described by the comments preceding it. (Assume that the method appears in a Java class in which all the necessary import statements also appear.)

```java
/* Reads the integer values in the file with the given name and, for each one, * reports its parity (i.e., whether it is odd or even). (For example, if 4 ** were among the inputs, it should print "4 is even".) Assumed is that the ** named file contains a sequence of integer values, separated by whitespace. */

public static void ReportParities(String fileName) {
    Scanner scanner = new Scanner(new File(fileName));

    while (scanner.hasNext()) { // repeat until no more data
        int num = scanner.nextInt(); // read next number

        if (num % 2 == 0) {
            System.out.println(num + " is even"); // report that num is even
        } else {
            System.out.println(num + " is odd"); // report that num is odd
        }
    }
}
```
4. Recall the following regarding commonly-used methods in the String class (where \( s \) is of type String, \( ch \) is of type char, and \( i \) is type int):

- `s.length()`: length of \( s \)
- `s.charAt(i)`: character at position \( i \) of \( s \).
- `s.indexOf(ch)`: position of the first occurrence of \( ch \) in \( s \), or \(-1\) if there is none
- \( s + ch \): string obtained by concatenating \( ch \) to the end of \( s \).

Consider the following method.

```java
public static String blop(String s) {
    String result = "";
    for (int i=0; i != s.length(); i = i+1) {
        char ch = s.charAt(i);
        int k = "aeiou".indexOf(ch); // looks for ch in "aeiou"
        if (k == -1) {
            result = result + ch;
        }
    }
    return result;
}
```

Suppose that the expression `blop("The bear growled")` were used to call the method.

(a) The value returned by the method would be `Th_br_gwld`.

(b) Write a sentence or two that would serve as a good comment describing the method's purpose. What is crucial to describe is the relationship between the String passed to the method (via its parameter) and the String returned by the method.

`Returns the string obtained from the given one by omitting every occurrence of a vowel.`

5. On the following page is a TossableCoin class, followed by a sample Java application that is a client of that class. In the space below that, show the `main()` method of a similar Java application that tosses a single coin until three tosses in a row result in tails and then prints the number of tosses that were made. (Hint: Use a variable to "remember" how many tails have occurred since the last heads.)

For two bonus points, explain how the given program could be improved by using a `do-while` loop in place of the `while` loop.
public class TossableCoin {

    // instance variable
    private char face; // Stores result ('H' or 'T') of most recent toss

    // constructor
    public TossableCoin() { face = ' '; }

    // observer: Returns result of most recent toss ('H' or 'T')
    public char faceShowing() { return face; }

    // mutator: Performs a toss.
    public void toss() { /* ... details omitted ... */ }
}

// Sample application program:
public class CoinTossApp {
    public static void main(String[] args) {
        TossableCoin coin1 = new TossableCoin();
        TossableCoin coin2 = new TossableCoin();
        coin1.toss(); coin2.toss(); // toss both coins
        int counter = 1; // to count number of tosses

        while (coin1.faceShowing() != coin2.faceShowing()) {
            coin1.toss(); coin2.toss(); counter = counter + 1;
        }
        System.out.println("# of tosses: " + counter);
    }
}

public static void main(String[] args) {
    TossableCoin coin = new TossableCoin();
    int TossCnt = 0; int TailsIntRouCnt = 0;
    while (TailsIntRouCnt != 5) {
        coin.toss(); TossCnt ++;
        if (coin.faceShowing() == 'T') {
            TailsIntRouCnt ++;
        } else {
            TailsIntRouCnt = 0;
        }
    }
    System.out.println("# tosses was " + TossCnt);
}