CS 122 – Review questions for test 2

Also look at questions from your first test review. Here are additional questions based on concepts you have seen in recent weeks. The three big topics have been methods, recursion and object-orientation.

1. How does the appearance of a function (or method) call differ from the function header?

2. Briefly describe the roles of each of the following types of class functions: constructors, get functions, set functions. Which of these types of functions should end in a return statement?

3. When we write a toString() method for a class, what parameters should we pass in? What is the return type?

4. Suppose a class Planet had a function findLife() that we call as follows in main:

   ```
   int value= p.findLife("goat", true, 0.5);
   ```

   How would the findLife() function be declared in Planet.java ?

5. (big question) Write a class called Rectangle that has 2 attributes called length and width. Write an initial value constructor that takes 2 parameters: the proposed length and width of the rectangle. Define two instance functions that compute the perimeter and area of any Rectangle. Also write a function called equals() that returns true if 2 rectangles have matching heights and widths. Write a short main() function that initializes a Rectangle object and outputs its perimeter and area. How would you test the equals() method?

6. (big question) Design a program that computes hourly pay for a set of employees. Write an Employee class whose attributes are name, hours worked, and hourly rate. Another class called Company will have one attribute: an array of employees. Somewhere you need a function that will return how much an employee earned in the week, and another function that will print the office payroll and figures out the total amount of money being paid to employees. There is more than one way to organize such a program, but for inspiration consider similar programs that involved 3 files.

7. Suppose we wish to extend the rational number class (discussed in lecture) to handle whole numbers. We could write 2 more instance methods: isWholeNumber(), which determines if a rational number is a whole number, and makeInt(), which returns the integer value of a rational number that is known to be a whole number. Implement these instance methods.

8. Suppose that you have you designed and implemented a class. A few days later, you decide to add one more attribute to the class. What modifications would you need to consider in the implementation?

9. Suppose that the header of an instance method in Pizza.java looks like this:

   ```
   public int eat(int slices)
   ```

   Write a suitable call to this instance method.

10. Write a recursive function (i.e. static method) called oddString() that takes a String parameter s, and returns a string that consists of the characters from s that are in the odd positions only (1st, 3rd, 5th, 7th, etc.). For example the expression oddString("wizard") should return "wzr".

11. What does it mean for the return type of a method to be void?

12. Which Java keyword is used when invoking a constructor?

13. In the Eight Queens problem, how many queens can occupy one column?

14. If we define f(1) = 1 and f(2) = 1 and for all n > 2, f(n) = f(n – 1) + f(n – 2), then how many recursive calls are there when we evaluate f(5)?

15. Suppose P(a, b) is defined as follows:

```
public int P(int a, int b)
{
  if (b == 0 || a == b)
    return 1;
  else
    return P(a – 1, b – 1) + P(a – 1, b);
}
```

    What is the total number of recursive function calls if we evaluate P(5, 3)?

16. How many parameters does each of the following types of constructors take?

    a. Initial-value
    b. Default
    c. Copy

17. If w is a palindrome, then which proper substring of w is also a palindrome?

18. Write a recursive static method in Java that will count how many array elements are negative, and return this number. The parameters to this function are an array of integers, and an integer denoting the index at which to start counting. Assume that the function will be initially called like this: int numNegative = count(a, 0);

19. Where do we use the Java keywords `implements` and `extends`? What do they mean?

20. How can we tell if a comparator class is sorting a numerical field ascendingly or descendingly?

21. The following recursive function attempts to find how many values in array a from a[start] through the end of the array are equal to a target number. But there are some mistakes in the algorithm. How would you fix the implemenation?

```
public static int getNumberEqual(int a[], int start, int target)
{
  int count;
  if (n >= a.length)
    return 0;
  else
  {
    if (a[start] == target)
      count = 1;
    else
      count = 0;
    return getNumberEqual(a, start – 1, target);
  }
}
```

22. Fill in the blanks below to complete this recursive implementation of a function that determines if a string is a palindrome. Feel free to insert additional (if) statements.

```
public static boolean isPalindrome(String s)
{
  if (____________________________________________________________)
    return true;

  else if (____________________________________________)
    return false;

  else
    return isPalindrome(____________________________________);
}
```

23. Write a recursive definition, in the form of a grammar, for the set of strings that begin with an ‘a’ and end with a ‘b’. You may assume that the alphabet is just these two letters.

24. What is the recursive definition of factorial?

25. In Java, all classes extend which built-in class?

26. What is the advantage to using inheritance in object-oriented design?

27. The Towers of Hanoi problem makes exactly _________ moves when it starts with N disks.

28. In a recursive algorithm, what are the two properties necessary to ensure that the recursion will terminate?

29. When is the base case of the Knight's Tour problem reached?

30. Suppose we have a program that defines classes Animal, Bird and Reptile. The variable x is declared to be of type Animal. During the program, x is modified by the following statements.

```
x = new Bird();
…
x = new Reptile();
```

What feature of Java allows us to redefine the dynamic type of x?

31. Suppose c1 and c2 are objects of the class Circle. A Circle has a single data member, its radius. The Circle class has a default constructor (implemented correctly), but no other methods have been defined in the implementation of the Circle class. Consider the following code. Will it compile? If so, what is the value of same?

```
Circle c1 = new Circle(12.0);
Circle c2 = new Circle(12.0);
boolean same = c1.equals(c2);
```

32. Fill in the blanks in the following recursive function so that it returns the reversal of the string parameter. For example, reverse("car") should eventually return "rac".

```
public static String reverse(String s)
{
  if (__________________________________________)
    return ____________________________________________________;

  else
    return ____________________________________________________;
}
```

33. Consider the following recursive function. Suppose this function is initially called as f(7).

```
public static int f(int n)
{
  if (n < 2)
    return 1;
  else
  {
    int retVal = 1 + f(n – 2);
    System.out.println(n);
    return retVal;
  }
}
```

a. What is the output?

b. What is the value returned by f(7)?

34. Fill in the blanks in the following recursive function so that returns true if the input string is of the form $a^{2n}b^{n}$ (e.g. ε, aab, aaaabb, aaaaaabbb, etc.), and false otherwise. Feel free to insert additional code between the if-statements.

```
public static boolean isGood(String s)
{
  if (___________________________________________________________)
    return false;

  if (_____________________________________________________________)
    return true;

  else
    return isGood(________________________________________________);
}
```

35. Write a recursive definition for:

a. the set of positive odd numbers

b. the set of strings of a's and b's in which the length of the string is odd – Please write this definition in the form of a grammar.

36. If we use `Collections.sort(a, b)` to sort an ArrayList of objects, then `a` represents the list we want to sort. What is the data type of `b`?

37. Suppose that a Java program has classes Animal and Bird, and that Bird extends Animal. Consider the following code that appears in a main function:

```
Animal a = new Animal( );
Bird b = new Bird( );
String s1 = a.toString( );
String s2 = b.toString( );
```

Assume that the Animal class has an implementation for toString( ) but Bird does not.

a. What will happen when we try to run the above code segment? Is there an error?

b. Repeat part (a), but this time assume that the Bird class has an implementation of toString( ), but Animal does not.

38. Consider the following recursive algorithm.

```
solve(int i, int j)
{
  if (i < 0 || j < 0 || i >= 5 || j >= 5 || board[i][j] == 0 || alreadyCounted[i][j] == true)
    return 0;
  else
  {
    alreadyCounted[i][j] = true;
    int a = solve(i + 1, j);
    int b = solve(i, j + 1);
    return a + b + 1;
  }
}
```

Assume that all the elements of alreadyCounted are initially false. Suppose the board is a 5x5 array of 0s and 1s (in which the rows and columns are numbered 0-4) that looks like this:

```
0 1 1 0 0
1 0 1 1 0
1 0 1 0 1
0 1 1 1 1
0 0 1 0 0
```

Given that we initially call solve(2, 2), list all of the return values from solve( ), in the order in which they occur. Be sure to distinguish the instances of solve by giving the parameter values. For example, you can phrase your answers like this – “solve(7, 8) returns 15”.