CS 6311 – Programming Languages I – Fall 2006
Quiz #7 (20 pts)
Monday, 10/30/6

NAME: KEY

1. What information is stored in the dynamic link in an activation record? (2 pts)

   The address of the top of the previous activation record.

2. Does the callee or the caller put the dynamic link on the activation record? (1 pt)

   Caller

3. In terms of a recursive method, what is wrong with this version of the division() method, that performs integer division of x divided by y? (3 pts)

```
public int division(int x, int y) {
  if (x<y) {
     return 0;
  }

  return 1 + division(x-y, y);
}
```

   a. There is nothing wrong with the recursive method. // Correct answer
   b. The base case check must be done with an = = not a <, so the recursion does not end.
   c. The recursive case cannot be included in the return statement.
   d. The method does division so the recursive case must do division.

4. Write a recursive method that sums the positive even integers up to and including n (when n is even), e.g. SumEven(5) = 2 + 4 = 6, and SumEven(8) = 2 + 4 + 6 + 8 = 20. The method declaration is given. (5 pts)

```
int SumEven(int n)
{
        if (n % 2 == 1) // See if odd, if so make n even
                n = n – 1;

        if (n == 0)
                return 0;
        else
                return n + SumEven(n-2);
}
```

5. Given the following C# code segment, answer the following questions. The Temp property of the Temp class is used to access the double data member that stores the temperature in the object.

```
public void createTemps() {
      double myTemp = 50.0;
      Temp t1 = new Temp(myTemp);
      Temp t2 = new Temp(myTemp + 20);
      t1 = t2;

      t2.Temp = 75;

      Temp t3 = modifyTemp(t2, ref myTemp);
      t2.Temp = myTemp + 15;

      System.out.println("t1: {0}", t1.Temp);
      System.out.println("t2: {0}", t2.Temp);
      System.out.println("t3: {0}", t3.Temp);
      System.out.println("myTemp: {0} ", myTemp);

      t1 = null;
      // HERE
}

public Temp modifyTemp(Temp t1, ref double temp) {
      t1 = new Temp();
      t1.Temp = temp + 10;
      temp = temp + 20;
      return t1;
}
```

a. What is the output of the above code, given an initial call to createTemps? (4 pts)

```
t1: 85
t2: 85
t3: 60
myTemp: 70
```

b. How many Temp objects are garbage when the program reaches the // HERE comment in createTemps? (2 pts)

One

6. Explain why a recursive algorithm is considered a divide-and-conquer approach. (3 pts)

A recursive algorithm works by breaking a problem down (dividing it) into smaller equivalent problems. Once the problem has been divided down to a very simple problem that can be solved easily (conquered), then the solution to the smaller problem can be used to solve (conquer) the solution to the next bigger problem, etc.