

---

Unless specified otherwise, write your answers on the provided answer sheets. You may refer to printouts of any source code you have written.

1. (3 pts) To what does the term “overhead” refer in the context of recursion. Identify specific sources of overhead.
2. Consider the following problem (adapted from codingbat.com):

The topmost row has 1 block, the next row down has 2 blocks, the next row has 3 blocks, and so on. Calculate the total number of blocks in such a triangle with the given number of rows. A triangle with 0 rows, of course, has 0 blocks.

- (a) (2 pts) Using words provide a recursive description of this problem. You might start with something like: “The number of blocks in a triangle of  $n$  rows is ...”. Be sure to identify a base condition as well.
  - (b) (6 pts) Write a recursive function called `numBlocks` that takes a single parameter representing the number of rows in the triangle and returns the number of blocks necessary to create the triangle.
3. (6 pts) Consider the following recursive function. What value is returned by `stuff(4,6)`? REMEMBER: an integer divided by an integer results in an integer (e.g.,  $7/2 = 3$ ).

```
public static int stuff(int a, int b) {  
    if (b==0) return 0;  
    if (b%2==0)  
        return stuff(a+a,b/2);  
    else  
        return stuff(a+a,b/2)+a;  
}
```

4. Suppose we continue from the previous exam with our concept of a representing a book club member in the following class. Each month they are charged \$19.99 for membership and in return are awarded 1 credit. Each credit can be redeemed for a book.

```
class Member
{
    protected String memberName; // name of the book club member
    protected int months;        // number of months of membership
    protected int credits;       // number of unredeemed credits
    public Member(String name) {
        memberName= name;
        months= 1;
        credits= 1;
    }
    public String toString() {
        return name+" "+credits+" "+calcDues();
    }
    public double calcDues() { return 19.99; }
    public int booksBought() { return months-credits; }
}
```

Suppose the book club has decided to offer an enhanced membership opportunity in which some members can choose to be “recruiting members”. A “recruiting member” pays an additional \$5.00 per month but in return they are eligible to receive discounts based on how many people they have recruited to join the book club. For each new member they recruit they get a one-time discount of \$10.00. Discounts are applied monthly and do not exceed the actual bill.

As an example, suppose in the first month of becoming a recruiting member, Jane recruits 1 new member. Her bill at the end of the month would be \$15.00 (i.e., \$25.00 monthly fee - \$10.00 for the one recruit). At the end of the second month her bill would be \$25.00 since the discount is applied only once. Suppose in the third month she recruits 3 customers to join the book club. In that month she will be able to apply 2 of the discounts and so would pay \$5.00 for her membership fee. The 3rd discount to which she is entitled would be applied the following month.

- (a) (4 pts) Create a new class called **RMember** to represent a “recruiting member”. This new class should inherit from **Member** and should add an integer attribute called **recruits** that is used to track the number of recruits this member has brought in but has not yet been awarded a discount for.
- (b) (4 pts) In the new **RMember** class create a constructor that accepts the customer name and an initial recruit count as parameters. It should invoke the constructor of it’s super class and should set the recruit count appropriately.
- (c) (6 pts) Draw a UML diagram depicting these two classes. When depicting attributes you should specify their type. When depicting methods you should spec-

ify their return type, name, and parameters. Do not include constructors or `toString()` in the diagram.

- (d) (6 pts) In the `RMember` class override the `calcDues` method so that the amount reported takes into account the number of folks recruited according the rules described above. NOTE: It should not make adjustments to the recruit count.
  - (e) Write a section of code that makes use of the `Member` and `RMember` classes as follows:
    - i. (4 pts) Establish an array (called `cust`) of 10 references to book club members.
    - ii. (4 pts) The first element of the array should refer to a regular member whose name is "Fred".
    - iii. (2 pts) The second element of the array should refer to a recruiting member whose name is Jane and who has 2 initial recruits.
    - iv. (2 pts) Display the monthly bill of each customer.
  - (f) (2 pts) In our current design would it make sense to declare the `Member` class as abstract or not? Briefly explain.
5. Suppose we want to represent rainfall amounts (in inches) for Abilene, TX as a  $12 \times 31$  two dimensional array of double values. The idea is that the first dimension represents the month and the second dimension represent the date.
- (a) (2 pts) Create a class called `AnnualRain` that contains an attribute called `rain` that provides a 2-dimensional array representation of annual rainfall described above.
  - (b) (4 pts) In the constructor reserve memory for the `rain` attribute.
  - (c) (4 pts) Also, in the constructor write code that will initialize the entire array to have a value of -1.0 (to indicate no data has yet been entered).
  - (d) (4 pts) Write a function called `setRainFall()` that accepts a month (integer in range 0 to 11) and a date (integer in range 0 to 30) and a rainfall amount (double in range 0.00 to 200.00) as parameters and, after verifying the parameters are in a proper range stores the result in the array. If any of the values are invalid then do nothing.
  - (e) (8 pts) Write a function called `countMissing()` that will count the number of days for which rainfall data is missing. Assume that a year has 365 days and that invalid dates (e.g., Feb 30) contain a -1.
6. (4 pts) Show something you know from this course that did not get asked on this exam.