

Unless specified otherwise, write your answers on the provided answer sheets. You may refer to the command sheets (without commentary) that were handed out in class.

1. (2 pts) Define recursion in the context of computer programming.
2. (2 pts) You are having a scintillating conversation with friends about programming and recursion. The question arises as to why anyone would learn recursion. One friend says: “Every programmer needs to learn recursion because some problems can’t be solved without it.” React to your friend’s statement.
3. As you work this problem it may be useful to know that the command `str.substring(1)` returns a copy of `str` with its first character removed. Consider the following problem:

“Count the number of space characters in a string.”

- (a) (4 pts) Using words provide a recursive description of this problem. Be sure to identify a base condition as well.
  - (b) (6 pts) Write a recursive function called `countSpaces` that takes a `String` as a parameter and returns the number of space characters in it.
4. Consider this recursive function:

```
public static int f(int a, int b) {  
    int remainder = a % b;  
    if (remainder == 0)  
        return b;  
    else  
        return f(b, remainder);  
}
```

- (a) (4 pts) What value is returned by this method if invoked with  $f(20, 8)$ ?
  - (b) (2 pts) Is this method tail recursive, or not?
5. Suppose you have been asked by a cable company to create a program to track customer status for various regions. In doing so you have created the following class to represent a customer:

```

public class Customer
{
    private String email; // customer email address
    private int cablePackage; // 1=basic; 2=supreme; 3=everything
    private boolean wantsPromotions; // false= don't send promotional emails

    public Customer(String email) {
        this.email= email;
        cablePackage= 1;
        wantsPromotions= true;
    }
    public void setCablePackage(int val) { cablePackage= val; }
    public int getCablePackage() { return cablePackage; }
    public double calcBill() {
        if (cablePackage == 1)
            return 19.99;
        else if (cablePackage == 2)
            return 39.99;
        else
            return 64.99;
    }
}

```

- (a) (4 pts) Some customers pay for Internet service in addition to their TV services. Create a new class called **ICustomer** that inherits from **Customer** and adds an integer attribute called **speed** that indicates the Mbps (10, 20, or 50) the customer has requested for their Internet service.
- (b) (6 pts) In the new **ICustomer** class create a constructor that accepts the email address of the customer and the speed of Internet service. It should invoke the constructor of it's super class.
- (c) (8 pts) In the **ICustomer** class override the **calcBill** method so that the monthly bill includes the amount for TV services and adds the following amount for Internet service: 10Mbps=\$24.99, 20Mbps=\$34.99, 50Mbps= \$44.99. Then apply a 10% discount as a reward to customers for purchasing both services.
- (d) (4 pts) Draw a UML diagram depicting these two classes. When depicting attributes you should specify their type. When depicting methods you should specify their return type, name, and parameters.

6. Suppose the **Customer** class and **ICustomer** class from problem 5 have been completed.
- (a) (3 pts) Write code to declare an array of 10 customers.
  - (b) (3 pts) In the first slot of the array create an ordinary (non-Internet) customer with an email address of your choice.
  - (c) (3 pts) In the second slot of the array create an Internet customer who has the fastest available service.
  - (d) (3 pts) Modify the second customer's cable package to the most expensive package.
  - (e) (3 pts) Display the expected bill for each customer.
7. Minesweeper is a game that was popularized by its inclusion with the Windows operating system. In the game a rectangular grid representing a mine field with a number of bombs is displayed with all grid positions initially represented as a bland gray box. The user attempts to map out the mine field without exploding any of the mines. To begin the mapping the user clicks on any box. If the box contains a mine the game is over. If not the box clicked-on grid position reveals a number indicating how many bombs are adjacent (including diagonals) to that grid location. If there are no bombs adjacent to the clicked-on area there is no number to display, just an empty grid location. If a grid location happens to be blank, that means it is safe to click on all 8 grids that surround it. For that reason, the game will automatically open up all blank grid positions surrounding a clicked-on blank grid until it encounters borders where there are numbers. The user can also right click to mark a suspected bomb location.

Suppose the following class has been developed to represent the value and state of a given grid element:

```
class GridElement
{
    public int value; // 0 if blank; -1 if bomb; -2 if border of the board;
                    // or 1-8 based on # of adjacent bombs
    public char state; // 'H' if hidden; 'V' if clicked/visible;
                    // 'M' if marked by user as a bomb
    public GridElement() { value= 0; state='H'; }
}
```

- (a) (4 pts) Create a class called **Grid** that is used to represent a  $n \times m$  minesweeper board. The board should be a two-dimensional array of **GridElement** objects. The class should also store the number of rows and columns in the board. Use the names **board**, **rows**, and **cols** for your attributes.
- (b) (8 pts) Write a method for the **Grid** class called **countBombs** that will look at an existing (initialized) board and return the number of bombs that exist.