
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 each) Briefly define each of the following terms in the context of this course:
 - (a) polymorphism
 - (b) pure object model
 - (c) abstract class
2. (2 pts) Name one criticism sometimes leveled at the use of recursion as a programming technique.
3. (4 pts) Name some useful strategies when writing successful recursive functions.
4. Consider the following problem:

Calculate the sum of all the numbers in an array.

 - (a) (3 pts) Using words provide a recursive description of this problem. You might start with something like: “The sum of all the numbers in an array is ...”. Be sure to identify a base condition as well.
 - (b) (6 pts) Write a recursive function called `addNumbers` that takes an array of double values as a parameter. You’ll also want a integer parameter that specifies the size of the array you are working with.
5. Imagine the game of Othello being played on a board with 15 columns and 10 rows. In this game there are two players and lots of two-sided pieces with red on one side and blue on the other. One player wants to get as many blue pieces on the board as possible while the other play wants to get as many red pieces on the board as possible. For the purposes of these questions all you need to know about the game is that the player who ends up with the most pieces of their color is the winner. Suppose we want to represent an Othello board as a 2d array of `chars`. If an array element has a space character then the corresponding position on the board is blank. If an array element contains ‘R’ then the board position contains a red piece (and, of course, ‘B’ for blue).
 - (a) (2 pts) Create a class called `OthelloBoard` that contains an attribute called `board` that provides a 2-dimensional array representation of the pieces as described above.
 - (b) (4 pts) In the constructor reserve memory for the `board` attribute.
 - (c) (4 pts) Also, in the constructor write code that will initialize the entire board to be empty.

- (d) (8 pts) Write a function called `winner` that will return 'B' if the board contains more blue pieces than red. It should return 'R' if there are more red pieces than blue. It should return 'X' if there are the same number of red and blue pieces.
6. (6 pts) Consider the following famous recursive function called Ackermann's function. What value is returned by `ack(1,2)`?

```
public static int ack(int m, int n) {
    if (m==0) return n+1;
    if (n==0) return ack(m-1,1);
    return ack(m-1, ack(m,n-1));
}
```

7. Suppose you have been asked by the Major League Baseball commissioner to create a program to track statistics of various teams. In doing so you have created the following class to represent a baseball player:

```
public class Player
{
    private String name;
    private int atBats;
    private int baseHits;
    private int rbis;    // runs batted in

    public Player(String name) {
        this.name= name;
        atBats= 0;
        baseHits= 0;
        rbis= 0;
    }

    public double battingAvg() {
        if (atBats == 0) {
            return 0.0;
        }
        return (double) baseHits / atBats;
    }

    public int getRbis() {
        return rbis;
    }
    public void setRbis(int rbis) {
        this.rbis= rbis;
    }
}
```

```

    public String toString() {
        return name+" "+rbis+"/"+battingAvg();
    }
}

```

Imagine that this class has been used in a variety of programs to represent players and it is working well. One concern is that this class does not support the a designation that is used only in evaluating the performance of pitchers: earned run average (ERA).

- (a) (4 pts) Create a new class called `Pitcher` that inherits from `Player` and adds the appropriate attribute to store a pitcher's ERA (which is a number with decimal places).
 - (b) (4 pts) In the new `Pitcher` class create a constructor that accepts the name of the player and invokes the constructor of it's super class. Also `era` should be 0.0.
 - (c) (6 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.
 - (d) (2 pts) In our current design would it make sense to declare the `Player` class as abstract or not? Briefly explain.
8. Assume that the `Player` and `Pitcher` classes explained in problem 7 have been correctly built. Write a section of code that will:
- (a) (4 pts) Declare an array of 20 references to players (some of which may be regular players and some of which may be pitchers).
 - (b) (3 pts) In the first slot of the array create a regular player object for a player named "Fred".
 - (c) (3 pts) In the second slot of the array create an object that holds a pitcher named "Ace".
 - (d) (4 pts) Suppose that the entire array has been initialized to have a full roster of 20 players. Write code to print a list of the entire team including each player's name, `rbis`, and batting average.