

# Computer Science 141 -- Exam 1

October 10, 2003

**General Directions.** This is an open-book, open-notes, open-computer test. However, you may not communicate with any person, except me, during the test. You have the full class period (50 minutes) in which to do the test. Put your answer to each question in the space provided (use the backs of pages if you need more space). Be sure to **show your work!** I give partial credit for incorrect answers if you show correct steps leading up to them; conversely, I do not give full credit even for correct answers if it is not clear that you understand where those answers come from. Good luck.

This test contains 4 questions on 4 pages.

**Question 1** (15 Points). Write (pseudocode is fine) a recursive algorithm that an object from a subclass of our robots can use to make itself turn to the left  $n$  times, where  $n$  is a parameter to the algorithm. You may assume that  $n$  is a natural number.

**Question 2** (15 Points). Here is an algorithm that prints certain strings of characters:

```
void printChars( int n ) {  
    if ( n > 0 ) {  
        System.out.print( "a" );  
        this.printChars( n-1 );  
        System.out.print( "b" );  
    }  
}
```

Prove that this algorithm prints strings consisting of  $n$  "a's" followed by  $n$  "b's", for every natural number  $n$ .

**Question 3** (10 Points). Here is a class that extends our robots with the ability to paint a tile and then move forward off of it. The class also includes a `main` method that exercises the `paintAndMove` message:

```
class ExtendedRobot extends Robot {  
  
    public void paintAndMove( java.awt.Color c ) {  
        this.paint( c );  
        this.move();  
    }  
  
    public static void main( String[] args ) {  
        ExtendedRobot r = new ExtendedRobot();  
        r.paintAndMove( java.awt.Color.blue );  
    }  
}
```

Explain how you would change the above code in order to measure how long it takes to handle one `paintAndMove` message. (Your explanation may consist of showing in the above code where you would make changes and what those changes would be.)

**Question 4** (10 Points). Here is a class that gives our robots a few new behaviors:

```
class BetterRobot extends Robot {
    public BetterRobot() {
        super();
    }
    public BetterRobot( int w, int h, int d, RobotRoom rm ) {
        super( w, h, d, rm );
    }
    public void safeMove() {
        if ( this.okToMove() ) {
            this.move();
        }
    }
    public void paintAndMove( java.awt.Color c ) {
        this.paint( c );
        this.move();
    }
}
```

Suppose you wanted another kind of robot, which provides the features that `BetterRobot` does, except that it also provides a `turnAround` message that makes a robot turn 180 degrees, and it handles `safeMove` messages by moving if possible, but printing an error message if it's not possible to move. Write (pseudocode is OK) a class with these features.