

Midterm Exam 2-1 Solutions

```
1. public void paintComponent(Graphics g)
   {
       g2 = (Graphics2D) g;

       // Write one or more loops calling the drawRectangle method
       // to produce the desired pattern
       for (int row = 0; row < ROWS; row++)
           for (int col = 0; col < COLUMNS; col++)
               {
                   if (row < 3 || row > ROWS - 1 - 3)
                       drawRectangle(row, col, Color.RED);
                   else
                       drawRectangle(row, col, Color.BLUE);
               }
           }
   }
```

Rubric:

Loop over rows and columns: Attempt - 1. Correct - 3

Determine horizontal strip: Attempt - 2. Correct - 4

Draw rectangles: Attempt: 1. Correct - 3

2. (a) cities should be created in the constructor, otherwise it is initialized to null
in reverse, cities should be set to the reversed list after it is filled
toString completely ignores the cities data

```
(b) import java.util.ArrayList;
    /**
     * An itinerary holds a collection of city names.
     */
    public class Itinerary
    {
        /**
         * Constructs an empty itinerary.
         */
        public Itinerary()
        {
            cities = new ArrayList<String>(); /******ADDED******/
        }

        /**
         * Add a city to the itinerary.
         * @param cityName the city to add
         */
        public void addCity(String cityName)
```

```

    {
        cities.add(cityName);
    }

    /**
     Returns a string describing the object.
     @return a string in the format "Itinerary[cityName1,cityName2,...]"
    */
    public String toString()
    {
        /**METHOD OVERHAUL***/
        String str = "Itinerary[";
        for (String city : cities)
            str = str + city + ",";
        if (cities.size() > 0) // If there is at least one city
            str = str.substring(0, str.length() - 1); //This drops extra comma
        str = str + "];";
        return str;
    }

    /**
     Reverses the elements in the itinerary.
    */
    public void reverse()
    {
        ArrayList<String> reversed = new ArrayList<String>();
        for(int i = cities.size() - 1; i >= 0; i--)
            reversed.add(cities.get(i));
        cities = reversed;  /**ADDED***/
    }

    private ArrayList<String> cities;
}

```

Rubric:

- a) Grade the first three answers, no matter how many are given
2 points for each that correctly identifies an error
- b) Correct solution: 2 points for toString, 1 for the others.

3. I have put the following into a static method, but that was not required.

```

public static void moveToExpressLine (ArrayList<Integer> checkOutLine,
                                     ArrayList<Integer> expressLine)
{
    int index = 0;
    while (index < checkOutLine.size())
    {
        if(checkOutLine.get(index) <= 15)

```

```

        {
            expressLine.add(checkOutLine.get(index));
            checkOutLine.remove(index);
            index--;
        }
        index++;
    }
}

```

Rubric:

Copy/paste unrelated problem: 0

Loop through the elements. Attempted: 1, Correct: 2

If statement to identify elements to be moved. Attempted: 1, Correct: 2

Use array list methods, not array brackets. 1

Remove element: Attempt: 1, Correct: 3 (Including index adjust)

Insert in correct position. Attempt: 1, Correct: 2

```

4. public class Person
{
    public Person(String aName)
    {
        name = aName;
        population++;
    }

    public String getName()
    {
        return name;
    }

    public static int getPopulation()
    {
        return population;
    }

    private String name;
    private static int population;
}

```

Rubric:

Add a getPopulation method. 1

Make it static. 3

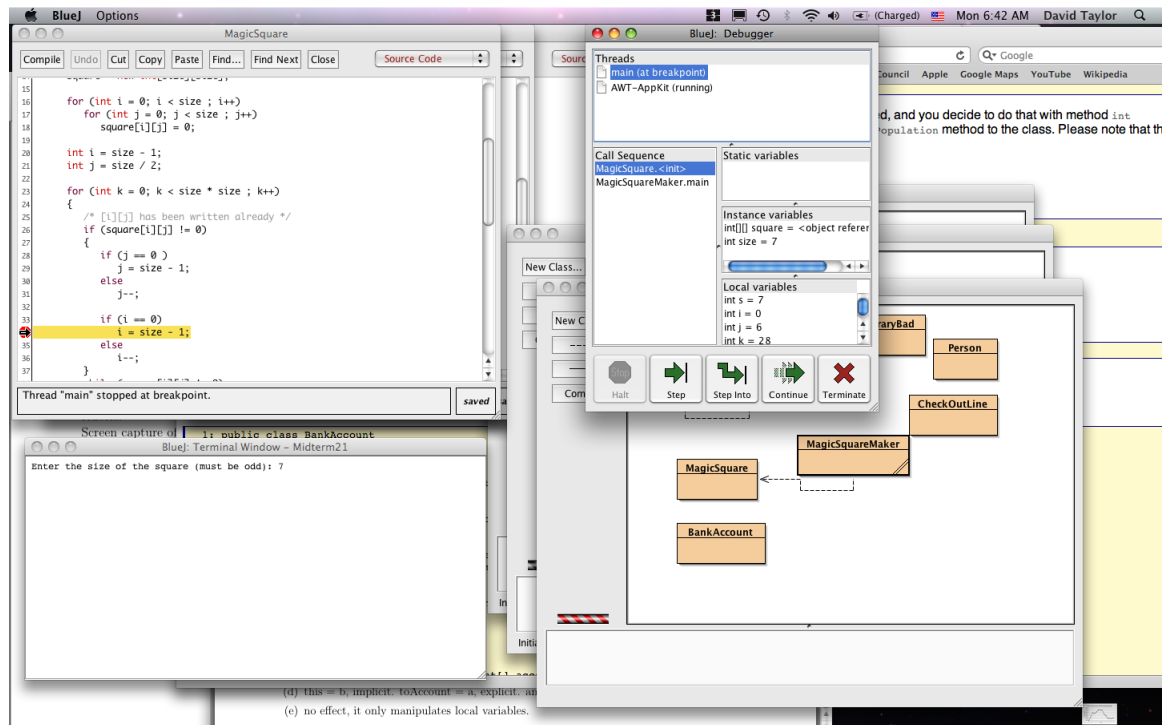
Add the field. 1

Make it static. 2

Increment it. Attempt: 1, Correct: 2

In the constructor: 1

5. (a) Screenshot:



(b) $i = 5, j = 6, k = 28$

Rubric:

Screen capture of breakpoint hit: 5 (or 4 if you forgot it)

Values of variables: 2, 3, or 5 points for 1, 2, or 3 correct. (or 2 each if you forgot screenshot)

6. (a) toBalance: lines 11-13. balance: instance variable, valid throughout all methods of the class.
- (b) On line 27, double amount overlaps in scope with the parameter variable amount
- (c) this.balance = balance;
- (d) this = b, implicit. toAccount = a, explicit. amount = 10, explicit.
- (e) no effect, it only manipulates local variables.

Rubric:

For each part, 1 point for partially correct answer, 3 points for correct answer