FRQ 1 — Methods & Control Structures

The StringAnalyzer class stores a string and provides methods to analyze it.

```
public class StringAnalyzer
{
    private String str;

    public StringAnalyzer(String s)
    {
        str = s;
    }

    public int countVowels()
    {
        /* part (a) */
    }

    public boolean hasMoreVowels()
    {
        /* part (b) */
    }
}
```

FRQ 2 — Class Design

A class Assignment represents a student assignment. Each assignment stores a title, a maximum score, and a student's score. The student's score should never exceed the maximum score.

```
Assignment hw = new Assignment("Homework 1", 50);

hw.recordScore(40);
System.out.println(hw.getPercent());
System.out.println(hw.isPassing());
```

Part A: Write the constructor that initializes all instance variables and sets the student score to 0.

Part B: Write recordScore to update the student's score. If the score exceeds the maximum, set it equal to the maximum.

Part C: Write getPercent to return the student's percentage score as a double.

FRQ 3 — ArrayList Data Analysis

The Item class represents a product with a name and a price. A ShoppingCart stores a list of Item objects and provides methods to analyze and modify the list.

```
public class Item
{
    private String name;
    private int price;

    public Item(String n, int p)
    {
        name = n;
        price = p;
    }

    public String getName() { return name; }
    public int getPrice() { return price; }
}

import java.util.ArrayList;

public class ShoppingCart
```

```
{
    private ArrayList<Item> items;

    public ShoppingCart(ArrayList<Item> list)
    {
        items = list;
    }

    public int countCheapItems(int maxPrice)
    {
        /* part (a) */
    }

    public void removeExpensiveItems(int maxPrice)
    {
        /* part (b) */
    }

    public String getMostExpensiveName()
    {
        /* part (c) */
    }
}
```

Part A: Count how many items have a price less than maxPrice using the getPrice method.

Part B: Remove all items whose price is greater than maxPrice. Be careful when removing from an ArrayList while looping.

Part C: Return the name of the item with the highest price using getName and getPrice.

FRQ 4 — 2D Array

A SeatingChart represents a classroom seating arrangement using a 2D array of student names. Each position represents a seat. The string "empty" represents an unoccupied seat.

```
public class SeatingChart
{
    private String[][] seats;

    public SeatingChart(String[][] s)
    {
        seats = s;
    }

    public int countEmptySeats()
    {
        /* part (a) */
    }

    public boolean containsStudent(String name)
    {
        /* part (b) */
    }

    public void markAvailable()
    {
        /* part (c) */
    }
}
```

Part A: Count the number of seats labeled "empty".

Part B: Return true if a given student name appears anywhere in the seating chart.

Part C: Replace all "empty" seats with "available".