Midterm 2 – A

10 Questions. While some questions may seem familiar to practice problems, there are likely to be subtle differences.

1.) Write a method `remove()` that accepts the following parameters: an array of integers `a`, and an integer index `n`. This method should return an array of integers with the element removed at the appropriate index, preserving the order of the remaining elements of the array.

```java
public class Question3{
    public static void main(String[] args){
        int[] x = {1, 1, 2, 3, 5, 8, 13};
        x = remove(x, 4); // results in array {1, 1, 2, 3, 8, 13}
    }
    // your method goes here
    // we’re assuming b is in the range of x’s size
    public static int[] remove(int[] a, int b){
        int[] z = new int[a.length - 1];
        for(int i = 0; i<b; i++){
            z[i] = a[i];
        }
        for(int i = b; i<z.length; i++){
            z[i] = a[i + 1];
        }
        return z;
    }
}
```

2.) Determine the output of the following program.

```java
public class Question4{
    public static void main(String[] args){
        Student a = new Student("Jeff", 22);
        Student b = new Student("Donny", 25);
        Student c = new Student("Walter", 30);
        Student d = new Student("Bunny", 35);
        System.out.println(a.numStudents);
        System.out.println(a.ID);
        System.out.println(d.numStudents);
        System.out.println(d.ID);
    }
}
```

Output:

```
4
1
4
4
```
3.) Write a recursive method `fact()`, that takes a single `int n` as input and returns the product of the positive integers from 1 to `n`. For example, calling `fact(4)` would return 24.

```java
public static int fact(int n) {
    if (n <= 1) {
        return 1;
    } else {
        return n * fact(n - 1);
    }
}
```

4.) Write a method, `concat()`, that accepts two integer arrays as parameters, and returns an array of the concatenated values of the two integer arrays accepted. The length of the returned array is the sum of the lengths of `x` and `y`, and the contents of the returned array are the values in `x` followed by the values in `y`. For example, for arrays `x:{1,3,5}` and `y:{2,4,6,8}`, a call to `concat(x,y)` will return array `{1,3,5,2,4,6,8}`.

```java
public int[] concat(int[] m, int[] n) {
    int[] z = new int[m.length + n.length];
    for (int i = 0; i < m.length; i++) {
        z[i] = m[i];
    }
    for (int i = 0; i < n.length; i++) {
        z[i + m.length] = n[i];
    }
    return z;
}
```

5.) Determine the output of the following program.

```java
public class Question7 {
    public static void main(String[] args) {
        Node list = null;
        for (int i = 0; i < 5; i++) {
            list = new Node(i, list);
        }
        mystery(list);
    }
    static void mystery(Node list) {
        if (list != null) {
            System.out.println(list.data);
            mystery(list.next);
        }
    }
}
```

```
public class Node{
    int data;
    Node next;
    Node(int d, Node n){
        data = d;
        next = n;
    }

    public class Node{
    int data;
    Node next;
    Node(int d, Node n){
        data = d;
        next = n;
    }
```
6.) Determine the output of the following program.

```java
public class Question1{
    public static void main(String[] args){
        Car a = new Car("yellow", 5000);
        Car b = new Car("orange", 10000);
        Car c = new Car("blue", 15000);
        Car d = new Car("red", 20000);

        a = b;
        a = c;
        roadTrip(c);
        roadTrip(a);
        roadTrip(b);
        d = a;
        System.out.println(a.mileage);
        System.out.println(d.color);
    }

    public static void roadTrip(Car c){
        Car a = c;
        a.drive();
        c.drive();
        c.drive();
    }
}

public class Car{
    String color;
    int mileage;
    Car(String color, int mileage){
        this.color = color;
        this.mileage = mileage;
    }
    void drive()
    {
        mileage++;
    }
}
```

Output:

```
15006
blue
```
7.) Write a class called RandomArray, that has one instance variable: an array of integers that holds 5 values. The class’s constructor should accept no parameters, and should initialize all elements in the array to random values between 1 and 50. The class should have the following instance methods:

highestIndex() – returns the index of the highest value in the array.
lowestIndex() – returns the index of the lowest value in the array.
average() – returns the average of the elements of the array.

```java
public class RandomArray {
    int[] a;
    RandomArray(){
        a = new int[5];
        for(int i = 0; i < a.length; i++){
            a[i] = (int)(Math.random() * 50 + 1);
        }
    }

    public int highestIndex() {
        int n = 0;
        for(int i = 0; i < a.length; i++) {
            if(a[n] < a[i]) {
                n = i;
            }
        }
        return n;
    }

    public int lowestIndex() {
        int n = 0;
        for(int i = 0; i < a.length; i++) {
            if(a[n] > a[i]) {
                n = i;
            }
        }
        return n;
    }

    public double average() {
        double sum = 0;
        for(int i = 0; i<a.length; i++){
            sum += a[i];
        }
        return sum/5;
    }
}
```
8.) Write a static method `equals()` that accepts two `Rational` objects as parameters, and returns true if the instance variables of the two `Rational` objects are the same, and false otherwise. Use the following `Rational` class for reference.

```java
public class Rational{
    int num;
    int denom;
    Rational(int num, int denom){
        this.num = num;
        this.denom = denom;
    }
}
```

// your static method goes below

```java
public static boolean equals(Rational r, Rational s){
    return(r.num == s.num && r.denom == s.denom);
}
```

9.) Write a class called `Point`. `Point` will consist of an x and a y – both fields are private, and will only store whole numbers. `Point` will also have a method called `distance()` that will accept a `Point` object as a parameter, determine the distance between its own x and y values and the x and y values of the `Point` parameter, and return that value as a double. (Hint: consider implementing Pythagorean theorem using methods provided by `Math` class. You are not allowed to use the `hypot` method).

```java
public class Point {
    private int x;
    private int y;
    int getX() {
        return x;
    }
    int getY() {
        return y;
    }
    void setX(int x) {
        this.x = x;
    }
    void setY(int y) {
        this.y = y;
    }
    double distance(Point p) {
        return(Math.sqrt(((p.getX() - x) * (p.getX() - x)) + ((p.getY() - y) * (p.getY() - y))));
    }
}
```
10.) Write a method `bubbleSort()` below such that an array of integers is accepted as a parameter, the contents are sorted in ascending order (you are not allowed to use Arrays class), and returns the sorted array. If you use a sorting algorithm other than Bubble Sort, say so in comments.

```java
public static int[] bubbleSort(int[] a){
    for(int i = 0; i<a.length; i++) {
        for(int j = 1; j<a.length - i; j++) {
            if(a[j-1] > a[j]) {
                int temp = a[j-1];
                a[j-1] = a[j];
                a[j] = temp;
            }
        }
    }
    return a;
}
```