School of Computer Science

COMP20010: Data Structures and Algorithms I
Semester I

Practical 1
On: 23/10/2018
Due: Not For Assessment

Q1: (5points)

What does the following code do?

```java
int[] a = new int[5];
System.out.println(a);
```

Q2: (5points)

Write your own `toString(int [] a)` method which returns a more useful string representation of an array, and test it works.

```java
public static String toString(int [] a) {
    // TODO
}
int[] a = new int[5];
System.out.println(toString(a));
```

Q3: (10points)

Create a class `ArrayEquals` which checks two 1 dimensional arrays for equality. The class should have the following methods:

```java
public class ArrayEquals {

    // return true if two integer arrays have same length and all
    // corresponding pairs of integers are equal
    public static boolean eq(int[] a, int[] b) {
        // TODO
    }


    // test client
    public static void main(String[] args) {
        int[] a = { 3, 1, 4, 1, 5 };
        int[] b = { 3, 1, 4, 1 };
        int[] c = { 3, 1, 4, 1, 5 };
        int[] d = { 2, 7, 1, 8, 2 };

        System.out.println(eq(a, a));
        System.out.println(eq(a, b));
        System.out.println(eq(a, c));
        System.out.println(eq(a, d));
    }
}
```

Q4: **(5points)**

Write a function which copies an array by iteration and returns the copy:

```java
public class Exercise {
 public static int[] copyArray(int [] a) {
    // TODO
 }

 public static void main(String[] args) {
   int[] a = {56, 14, -46, 15, 36, 99, 77, 18, 29, 49};

   int[] b = copyArray(a);
   // check its not a clone
   a[0] = -1;
   System.out.println(toString(b)); // print a string representation of the array
  }
 }
```

Q5: **(5points)**

Write a Java statement which creates an array with 3 rows and 5 columns?

Q6: **(5points)**

Write a Java function which prints a representation of a 2 dimensional array?

```java
public class Exercise {
    public static String toString(int [][] a) {
        // TODO
    }
    public static void main(String [] args) {
        int [][] a = new int[5][5];
        System.out.println(toString(a));
    }
}
```

Q7: **(5points)**

Is Java pass-by-reference or pass-by-value? What does the following code do? Is it what you expect?

```
public class ExerciseArrayIncrement {
    public static void increment(int [] a) {
        for (int i = 0; i < a.length ; i++) {
            a[i] += 1;
        }

    }
    public static void main(String[] args) {
        int N = 10;
        int[] a = new int[N];
        Random random = new Random();
        for(int i = 0; i < N; i++) {
            a[i] = random.nextInt(100);
        }
        System.out.println("Original array : " + Arrays.toString(a));
        increment(a);
        System.out.println("Incremented array : " + Arrays.toString(a)↩
           );
    }
}
```

Q8: **(5points)**

Write a function which takes an array as an argument and reverses the order of the elements of the array.

```
public class ExerciseArrayIncrement {
    public static void reverse(int [] a) {
        // TODO
    }
    public static void main(String[] args) {
        int N = 10;
        int[] a = new int[N];
        Random random = new Random();
        for(int i = 0; i < N; i++) {
            a[i] = random.nextInt(100);
        }
        System.out.println("Original array : " + Arrays.toString(a));
        reverse(a);
        System.out.println("Reversed array : " + Arrays.toString(a));
    }
}
```

Q9: **(10points)**

Write a Java program to move all 0's to the end of an array. Maintain the relative order of the other (non-zero) array elements:

```java
 public class Exercise {
    public static void moveZeros(int [] a) {
        // TODO
    }

     public static void main(String[] args) throws Exception {
        int[] a = {0,0,12,0,2,0,0,0,5,0,8};
         int i = 0;
        System.out.print("Original array: " + Arrays.toString(a));

        moveZeros(a);

        System.out.print("After moving 0: " + Arrays.toString(a));
    }
}
```

```
Original array: [0, 0, 12, 0, 2, 0, 0, 0, 5, 0, 8]
After moving 0: [12, 2, 5, 8, 0, 0, 0, 0, 0, 0, 0]
```