Quiz 1 — Data Struct. & More (T. III/21–22)

Name: Chandt Keratiyutwong
ID: 6380181

Directions:

- This quiz is paper-based. Answer all the questions in this booklet.
- No consultation with other people, notes, books, nor the Internet is permitted. Do **not** use an IDE or run Java code.
- This quiz is worth a total of 30 points. You have 50 minutes. Good luck!

4.6

Problem 1: Instant Questions (7 points)

Consider the following Java code snippet:

```
class Feline { int w; }
void update(int x, String y, int[] z, double d, Feline f) {
   x = 1; y = "hello"; z[0] = 3; d = 4.0; f.w = 5;
}
```

3

(i) **[3 points]** For each of the following variables, indicate whether it is a reference or primitive type as shown in the snippet:

x primitive, y reference, z reference, d primitive, f reference

(ii) **[4 points]** Suppose the main method below is run.

1.6

```
public static void main(String[] args) {
   int x = 10; String y = "ye", int[] z = {1,5,0,2};
   double d = 44.44; Feline f = new Feline();

   update(x, y, z, d, f);
}
```

Right before the method terminates, what is the value of the following?

x 1 (✗ 10), y ~~"hello"~~ (✗ "ye"), z[0] 3 (✓), d 4.0 (✗ 49.99), f.w 5 (✓)

Problem 2: What Will Java Do (8 points)

6

Carefully consider the following Unknown class.

```
public class Unknown {
   public static int w;
   private static int x;
   private int y;
   public int[] z;

   public Unknown() { w = 55; z = new int[3]; }

   public static void setX(int i) {x += i;}
   public void setZ(int i) {z[y] = i; y += 1;}

   public int getW() {return w;}
   public int getX() {return x;}
   public int getY() {return y;}
   public int getZ(int i) {return z[i];}
}
```

> global variables, always accessible.
> must create object of class Unknown to access

Each program below is a complete implementation that refers to the class Unknown (above). For each program, write down the output for each print statement, or if that line results in an error, write "ERROR" and a brief explanation.

```
1. import java.util.Arrays;
   public class P1 {
     public static void main(String[] args) {
       System.out.println(Unknown.x);      // ____
       System.out.println(Unknown.y);      // ____
       System.out.println(Unknown.w);      // ____
       System.out.println(Arrays.toString(Unknown.z)); // ____
     }
   }
```

private access (ERROR)

y and w are not static variables, (global)

must create object of class Unknown (new Unknown()) to access.

ex: Unknown u1 = new Unknown();

```
2. public class P2 {
     public static void main(String[] args) {
         Unknown u1 = new Unknown();
         u1.setX(7);
         System.out.println(u1.getW()); // ____

         Unknown u2 = new Unknown();
         u2.setX(11);
         System.out.println(u2.getX()); // ____
     }
   }
```

w=55, z = new int[3], y=0

x = 7

55

x = 7+11 = 18

18

```
3. public class P3 {
     public static void main(String[] args) {
         Unknown u1 = new Unknown();
         u1.setZ(5);
         System.out.println(u1.getZ(0)); // ____

         Unknown u2 = new Unknown();
         u2.setZ(4);
         System.out.println(u2.getZ(1)); // ____
     }
   }
```

w=55, z = new int[3], y=0

z[0] = 5, y = 1

5

w=55, z = new int[3], y=1

z[1] = 4, y = z

y=0

4

0

z[0] = 4, y = 1

z[1] = 0.

Problem 3: Fill in the Blanks (5 points)

5

Below is a method inside a class. The method takes as input an array of `String`s and returns an array of `int`s, where the *i*-th number in the output array is the number of vowels in the *i*-th `String` in the input array. In the English alphabet, the vowels are `a`, `i`, `o`, and `u`. For example, abusing the array notation for brevity, expect `numberOfVowels({"hwk", "el", "loila", "E,W", "Urla!"})` to return `{0,1,3,1,2}`. It is possible that the input is an empty array (i.e., array of size 0). Complete the code below by filling in the blanks.

```
public int[] numberOfVowels(String[] arr) {
   ____(1) counts = new int[arr.length];

   for (int i=0;i<arr.____(2);i++) {
      String st = arr[i].toLowerCase();

      for (int k=0;k<st.length();k++) {

         char ch = ____(3); // extract the k-th character

         if (____(4))
            counts[i]++;
      }
   }

   return counts;
}
```

(1) int[]

(2) length

(3) st.charAt(k)

(4) ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u'

Problem 4: Singly-Linked List With a Sentinel (2 + 2 + 6 points)

7.5

The class `SLList` below implements an `int` singly-linked list *with* a front sentinel, like was discussed in class. More precisely, the instance variable `sen` refers to the sentinel node and thus `sen.next` refers to the real first node. Our list is therefore defined by the linked list starting after the sentinel. For this problem, you'll add 3 extra methods to the `SLList` class:

-1 (1) a method **`public boolean isSingleton()`** that returns true if the list contains exactly one element and return false otherwise.

-0.5 (2) a method **`public void removeFirst()`** that removes the element at the front of the list. If the list is empty, this method will have no effects.

(3) a method **`public int sum()`** that returns the sum of the numbers in the list. Remember that the sentinel isn't part of the list. If the list is empty, the method should return `0`.

IMPORTANT: You cannot add new member variables to either class, nor can you modify existing methods/-constructors. Your methods can be iterative or recursive, or any combination of them that works.

```
public class SLList {
   private static class IntNode {
      int head;   // an int data item
      IntNode next; // ref to the next node

      public IntNode(int val, IntNode next) {
         this.head = val; this.next = next;
      }
   }

   private IntNode sen;

   public SLList() { sen = new IntNode(0, null); }

   public void addFirst(int x) { sen.next = new IntNode(x, sen.next); }
}
```

① 1

```
public boolean isSingleton() {
   if (sen.next.next == null) {
      return true;
   }
   return false;
}
```

or

what if sen.next == null? -1

② 0.5

```
public void removeFirst() {
   if (sen.next != null) {
      sen.next.next = null;
   }
}
```

everything is gone?

③ 6

```
public int sum() {
   IntNode p = sen;
   int sumN = 0;

   if (p.next == null) {
      return 0;
   }

   while (p.next != null) {
      p = p.next;
      sumN += p.head;
   }

   return sumN;
}
```

(1)
```
public boolean isSingleton() {
    if (sen.next == null)
        return false;

    else if (sen.next.next == null)
        return true;

    else return false;
```

(2)
```
public void removeFirst() {
    IntNode in = sen;
    sen = sen.next.next;
    in.next = null;
}
```