CMPS12A – Practice Final
Fall 2017

1.) Write a static method `findIndex()` that accepts an array of String `ar`, and a String `s`. The array is traversed, and the index of the *first* occurrence of `s` is returned. If no occurrence of `s` is found, `-1` is returned.

2.) Given the following program, complete the methods below:

```
class Question1{
   public static void main(String[] args){
      int[][] twoD = {{3,6,4,7}, {2,3}, {8,5,6}};
      int a = findHighest(twoD);
      int b = findSum(twoD);
      System.out.println(a + " " + b);
   }
   // this method should find the highest value in the
   // 2d array.
   public static int findHighest(int[][] z){









   }
   // this method should find the sum of elements in
   // the 2d array.
   public static int findSum(int[][] y){









   }
}
```

3.) Complete the recursive method `fib()` below such that the method computes values in the Fibonacci sequence. The Fibonacci sequence is defined by:
fib(0) = 1, fib(1) = 1
integer x > 1: fib(x) = fib(x - 1) + fib(x - 2)

```
int fib(int x){









}
```

4.) Given the Node class shown here, write a method that takes a Node and returns the sum of all elements in the linked list starting with that element. Complete this method (a) recursively, and (b) iteratively.

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

(a)

```
// recursive
int sum(Node n){









}
```

(b)

```
// iterative
int sum(Node n){









}
```

5.) Determine the output of the program below:

```
class Question4{
   public static void main(String[] args){
      Tree red = new Tree("Redwood", 300);
      Tree seq = new Tree("Sequoia", 200);
      Tree euc = new Tree("Eucalyptus", 75);

      Tree a = red;
      Tree b = seq;
      Tree c = b;

      grow(a);
      grow(b);
      grow(c);

      System.out.println(a.toString());
      System.out.println(b.toString());
      System.out.println(c.toString());

   }

   public static void rain(Tree t){
      Tree a = t;
      t.height += 10;
      a.height += 10;
   }

   public static void grow(Tree t){
      t.height += 5;
      rain(t);
   }
}

class Tree{
   String species;
   int height;
   Tree(String s, int h){
      species = s;
      height = h;
   }
   public String toString(){
      return species + " " + height + " ft"; "
   }
}
```

Output:

6.) The following program should display a single button that when clicked prints "Submitted." Fill in the blanks to complete the program. Everything you need to fill the blanks is either in the following list of fragments or is a variable in the incomplete program (e.g. frame, display, submit, etc.). Some items in the list may be used to fill more than one blank. Every item is used at least once.

```
ActionEvent
ActionListener
addActionListener
actionPerformed
frame
getContentPane
new Submit()
```

```
import java.awt.event.*;

class Submit implements ____________________________{
  public void ___________________ (_____________________ e) {
    System.out.println("Submitted");
  }
}
```

```
import javax.swing.*;
import java.awt.*;

class SimpleButton {
  public static void main(String[] args) {
    JFrame frame = new JFrame("Simple Button");
    Container display = ___________________._____________ ();
    JButton submit = new JButton("Submit");
    ____________________________.add(_____________________________);
    submit. _____________________ (_____________________________);
    _____________________________.pack();
    _____________________________.setVisible(true);
  }
}
```

```
// The following classes are for the next two questions
class ClassOne {
   int data = 123;
   public int get() {
      return data;
   }
   public int mystery() {
      return get()*10;
   }
}
class ClassTwo extends ClassOne {
   int data = 234;
   public int get() {
      return data;
   }
}
```

7.) What does the following program print?

```
class Question6 {
   public static void main(String[] args) {
      someFunc(new ClassOne());
      someFunc(new ClassTwo());
   }
   static void someFunc(ClassOne one) {
      System.out.println(one.mystery());
      System.out.println(one.get());
   }
}
```

8.) What does the following program print?

```
class Question7 {
   public static void main(String[] args) {
      foo(new ClassOne());
      foo(new ClassTwo());
   }
   static void foo(ClassOne one) {
      someFunc(one);
   }
   static void someFunc(ClassOne one) {
      System.out.println("one " + one.get());
   }
   static void someFunc(ClassTwo two) {
      System.out.println("two " + two.get());
   }
}
```

9.) Determine the output of the following program:

```
import java.util.Scanner;
import java.io.*;
class Problem1{
   public static void main( String[] args ){
      int a=5, b=6, c=1;
      double x=0.5, y=1.0, z=1.5;
      c = fcn1(a, b);
      y = fcn2(y, a);
      b = fcn3(x, y);
      z = fcn3(c, b);
      System.out.println("a="+a+", b="+b+", c="+c);
      System.out.println("x="+x+", y="+y+", z="+z);
   }
   static int fcn1(int i, int j){
      int k = i-j;
      k++;
      return (k);
   }
   static double fcn2(double t, int n){
      return (t*n);
   }
   static int fcn3(double u, double v){
      return fcn1((int)(u*v), 2);
   }
   static double fcn3(int r, int s){
      return fcn2(r,s);
   }
}
```

10.) The following program should display a single button that when clicked prints "Submitted." Fill in the blanks to complete the program.

```
import java.awt.event.*;
import javax.swing.*;
import java.awt.*;

class Sub implements ActionListener {
   public void actionPerformed ( ActionEvent e) {
      System.out.println("Submitted");
   }
}

class SimpleButton {
   public static void main(String[] args) {
      JFrame window = new JFrame("Simple Button");
      Container content = window.getContentPane();
      JButton theButton = new JButton("Submit");

      _____________.add(____________________);
      _____________.addActionListener(____________________);
      window.pack();
      window.setVisible(true);
   }
}
```

11.) Write a method `howMany()` that takes two parameters, a `String, line` and a `char, symbol`. The method should return an integer that is the number of times `symbol` appears in `line`. For example, if passed "This is a test." and 't' the result would be 2 (case sensitive). Or if passed "This is a test." and 's' the result would be 3. Hint: Given a `String line`, `line.charAt(n)` returns the character stored a position n in the String.