UNIVERSITY OF BIRMINGHAM

School of Computer Science

First Year – BSc Artificial Intelligence and Computer Science
First Year – BSc Natural Sciences
First Year – BA Computer Studies/French
First Year – BSc Computer Science
First Year – BSc Computer Science with Industrial Year
First Year – BSc Computer Science with Business Management
First Year – MEng Computer Science/Software Engineering
First Year –BSc Mathematics and Computer Science
First Year – MSci Pure Mathematics and Computer Science

06 18190

Software Workshop 1

Summer Examination 2013

Time allowed: 3 hours

[Answer ALL Questions]

No calculator

SECTION A.

[Use a Separate Answer Book for THIS Section]

1 Answer the following questions about the Java programming language.

(a) What is the value of `x` after this line is executed? [1%]

```
double x = 7 * 3 / 2 - 0.5;
```

(b) The following **for** loop has three parts missing from its first line. Write down that first line with the three parts filled in so that when the **for** loop runs it will print out the 1st, 3rd, 9th, and 27th elements of a double array `myArray`. [3%]

```
for (??? ; ??? ; ??? ) {
    System.out.println(myArray[k]);
}
```

(c) We have a class `C` and another class `D` that extends `C`. Each class has a constructor with no parameters. For each of the Java statements below, say whether it is legal. Explain your answers. [4%]

(i) `C x = new D();`
(ii) `D x = new C();`

2 The following method is intended to calculate the maximum value in an integer array (assumed non-empty).

```
static int max(int[] a) {
    int i = 0;
    double m = 0.0;

    while (i <= a.length) {
        if (a[i] > m);
            m = a[i];
        i = i + 1;
    }
    return m;
}
```

The method contains **four mistakes** that prevent it from calculating the correct result. Find those mistakes and state how they could be corrected. Note that the declaration of `m` as double is not sensible but OK. It should NOT be considered as a mistake.

[12%]

No calculator

3. (a) Arrays are indexed lists of elements. [3%]

(i) Explain the difference between the index and the value of an array element.

(ii) If the type of the array is `String[]`, what are the types of the indexes and of the values?

(b) A static method `myArray` is required, that constructs, initializes and returns an array. It has one parameter n, which will be the length of the array to be returned. The type of the returned result is an array of integers. Its elements are initialized as {0, 1, 4, 9, 16, 25 …, $(n-1)^2$}, i.e. squares of 0,1,2,3,…, $n-1$:

(i) Complete this definition of `myArray`: [9%]

```
  /*
 requires: n >= 0
*/
  public ______  _______  myArray (____ n) {

      ______ res = _____;

      for (____ ; ____ ; ____) {
         res[k] = ____;
      }
      return res;
  }
```

(ii) We can assign to an integer array variable `arr` the element values {0, 1, 4, 9, 16} using e.g. statement "`arr = {0, 1, 4, 9, 16};`", provided that `arr` has already been declared. Instead, write Java code that declares `arr`, and then uses a call of `myArray` to assign those element values to `arr`. [2%]

(c) (i) What is the meaning of this line "`requires: n >= 0`" in the *non-defensive* header comment for `myArray`? [1%]

(ii) Write a complete *defensive* header comment for `myArray`, to specify that it throws an illegal argument exception if `n` < 0. (You will gain an extra mark if you use correct Javadoc.) What would you need to include in the Java code to make this happen? [5%]

4 Suppose someone has already written a Java class `Triangle` whose instances represent triangles, described by the lengths of their three sides. Its constructor takes the three side lengths as parameters. It has various methods, including a method `area` that returns the area of the triangle.

(a) Write a *static* method `printArea` that takes three doubles as parameters, representing the side lengths for a triangle, and prints out (to `System.out`) the area of the triangle. It should work by creating an instance of `Triangle`, and calling its given `area` method. [4%]

(b) Write a Java definition for a class `Equilateral` that extends `Triangle` and represents equilateral triangles (all three sides are equal in length). It should have its own private, final field `side`, of type `double`, which should be equal to the three sides stored for `Triangle`. Most of the methods of `Triangle` will be inherited, but you should override the `area` method so that it uses the formula

```
Math.sqrt(3)/4*side*side
```

The constructor of `Equilateral` takes only one double parameter `side`. Include full Javadoc, and also an appropriate invariant condition for `side`.[7%]

(c) An interface `DataItem` is defined as follows.

```
public interface DataItem {
    public double value();
}
```

Write a Java definition for a class `AreaData` that implements `DataItem`. It should have a private, final field `theTriangle` of type `Triangle`. Its constructor should initialize that field from a parameter. Its `value` method calls `area` on `theTriangle`. [4%]

No calculator

5. The sine function can be computed from its Taylor series expansion:

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots \qquad \text{for all } x$$

To compute this series, one needs to compute terms $\frac{x^n}{n!}$ for positive n.

Write a static method `itemN` with input `x` (of type double) and `n` (of type integer), returning $\frac{x^n}{n!}$. Inside `itemN`, you need to call two static methods called `powN` and `fact` that calculate x^n and $n! = 1 * 2 * ... * n$, respectively, in a recursive manner. Write a Java code for

(a) `itemN` [2]

(b) `powN` [5]

(c) `fact`. [5]

No calculator

SECTION B

[Use a Separate Answer Book for THIS Section]

6. (a) The heading for the class `HashMap` in the Java library package `java.util` includes the following elements:

```
public class HashMap<K,V> ...
```

The method `put` defined in the class `HashMap` starts with:

```
    public V put(K key, V value) ...
```

Explain the use of the types `K` and `V` in the above code fragments.

[3%]

(b) Write a declaration for a `HashMap` variable and construct a `HashMap` object that could be used to store a simple dictionary for mapping a word in one language (String) to the corresponding word in another language (String).

[3%]

(c) Write a class `Dictionary` that can be used to represent an English-French dictionary. Your dictionary class should use two `HashMap` objects, one for mapping an English word to a corresponding French word and the other for mapping a French word to a corresponding English word. Your class should include the following methods

- `add` for adding a given pair of words to the dictionary (the pair should be added to both mappings);
- `getFrench` returns the French word corresponding to a given English word
- `getEnglish` returns the English word corresponding to a given French word

Your answer should include a main method that tests your class by adding three pairs of words to the dictionary and calling the `getFrench` and `getEnglish` methods.

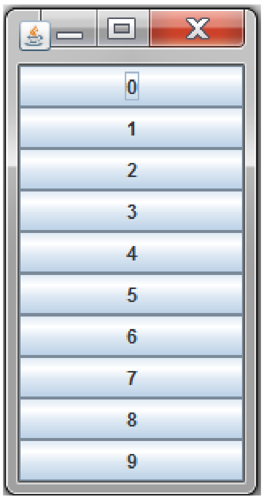
Hint: Word pairs could be: (dog, chien), (cat, chat), (tree, arbre) [6%]

(d) What computational complexity (using Big-O notation - e.g. constant: O(1), linear: O(n), logarithmic: O(log n), etc.) does a `HashMap` provide for inserting pairs (`put`) and finding paris (`get`)? What alternative implementation of a Map in Java do you know and what is the computational complexity of it?

[4%]

7. The window to the right is displayed by a program that is going to be developed to dial numbers. The window contains a panel described by the following Java class:

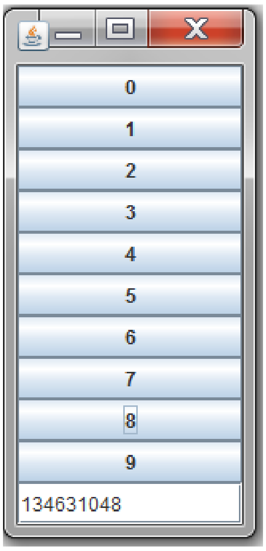
```
public class DialPanel  {

     public JPanel createDialPanel() {

        JPanel p = new JPanel();
        p.setLayout(new GridLayout(10,0));
        for(int i =0;i<10;i++) {
              JButton b = new JButton();
              b.setText(""+i);
              p.add(b);
        }


       return p;
     }
}
```



(a) Write a Java class `FrameDial` that can be used to create windows like the one illustrated above, and write a main method to display one such window. [5%]

(b) Now modify the `DialPanel` class so that if numbers are selected they could be displayed in a text field below the buttons. This means you need to add a text field. [5%]

(c) Now modify your `DialPanel` class so that the number(s) selected via the buttons are shown in the text field. Your `DialPanel` class should be modified to implement `ActionListener` so that it can respond to the clicks on the buttons.



134631048

For your information, the following is a listing of the interface class `ActionListener` from the Java API:

```
package java.awt.event;

import java.util.EventListener;

public interface ActionListener extends
EventListener {

    public void actionPerformed(ActionEvent e);

}
```

[7%]