5COSC001W - Solutions to Tutorial 5 Exercises

1 Static Methods

```
class CalcManager {
    public static boolean isEven(int n) {
        if (n % 2 == 0)
            return true;
        else
            return false;
    }

    public static int cube(int n) {
        return n*n*n;
    }

    public static double add(double...x) {
        double sum = 0.0;

        for (double e : x)
            sum = sum + e;

        return sum;
    }
}

public class CalcManagerTest {
    public static void main(String[] args) {
        double sum1 = CalcManager.add(2.1, 5.6);
        System.out.println("sum1 = " + sum1);

        double sum2 = CalcManager.add(7.9, 1.3);
        System.out.println("sum2 = " + sum2);

        System.out.println("isEven(4): " + CalcManager.isEven(4));
        System.out.println("isEven(7): " + CalcManager.isEven(7));

        System.out.println("cube(3): " + CalcManager.cube(3));
        System.out.println("cube(10): " + CalcManager.cube(10));
    }
}
```

The program displays:

```
sum1 = 7.699999999999999
sum2 = 9.200000000000001
isEven(4): true
isEven(7): false
cube(3): 27
cube(10): 1000
```

2 Static Fields

1. The output of the program is:

```
e1 contains: Name: John, 77 Exhibition Road London
e2 contains: Name: George, 77 Exhibition Road London
e3 contains: Name: Helen, 77 Exhibition Road London

After e1.company_address = a2
e1 contains: Name: John, 5 Guilford Road York
e2 contains: Name: George, 5 Guilford Road York
e3 contains: Name: Helen, 5 Guilford Road York

Memory location for e1.company_address object: Address@15ff48b
Memory location for e2.company_address object: Address@15ff48b
Memory location for e3.company_address object: Address@15ff48b

e1.numberOfEmployees: 3
e2.numberOfEmployees: 3
e3.numberOfEmployees: 3
```

3 Since `company_address` is static in `Employee` it is shared among all objects of the class. Therefore it can only have one value for all objects, and employees working for different companies cannot be created in this program.

3 Using super() - Implementing the toString() method

```
class Book {
    private int pages;  // number of pages in the book

    Book(int pages) {
        this.pages = pages;
    }

    public String toString() {
        return ""+pages;  // convert pages to String before returning
    }
}
```

```
class Dictionary extends Book {
    private int words;  // number of words in the dictionary

    Dictionary(int words, int pages) {
        super(pages);
        this.words = words;
    }

    public String toString() {
        String s = "Number of pages: " + super.toString();
        s = s + ", Number of words: " + words;

        return s;
    }
}

public class BookTest {
    public static void main(String[] args) {
        Dictionary d1 = new Dictionary(100000, 500);
        System.out.println(d1);
    }
}
```

4 Final Classes and Methods

1. A `final` class cannot be extended. Therefore `X2` cannot extend `X1`. Either `X1` should not become final, or `X2` should not attempt to inherit from it.

2. `final` method cannot be overridden. Therefore, `foo` cannot be overridden in `X3`. The `final` keyword should be removed from the `foo` version of `X2`.

5 Construction of Objects

1. ```
   Cell constructor called
   TinyCell constructor called
   MicroscopicCell constructor called
   ```

2. The `TinyCell2` constructor will call the default constructor of its parent class `Cell2`. As `Cell2` defines a constructor with arguments, the compiler will not synthesise a no-arguments constructor for `Cell2` and the compiler will report an error.

   Adding the line `super(5)` at the indicated point will force the constructor of `TinyCell2` to call the `Cell2(int)` constructor as opposed to the default constructor. The program will then compile and run.
   ```

6 Using the instanceof operator

```
Treenode-> data: 100
Vertex-> x: 500, y: 400
```

Make sure that you understand why this output was obtained.

7 Challenge: A Clock Program: Implementing Class Hierarchies - Inheritance

This is an optional challenge exercise. If you attempt this and if you have any doubts about your solution, you could show this to your tutor.