Duration: ***50 minutes*** **(10:10am–11:00am)**

Aids Allowed: ***The Java API: An Introduction for Students*****, July 2000 edition.**

Student Number: ____________________

Last Name: ____________________

First Name: ____________________

Tutor's Name: ____________________

Do ***not*** *turn this page until you have received the signal to start.*
(In the meantime, please fill out the identification section above,
and read the instructions below *carefully*.)

This test consists of 3 questions on 5 pages (including this one). *When you receive the signal to start, please make sure that your copy of the test is complete.* Answer each question directly on the test paper, in the space provided, and *use the reverse side of the pages for rough work.* (If you need more space for one of your solutions, use the reverse side of the page and indicate **clearly** which part of your work should be marked.)

Be aware that concise, well thought-out answers will be rewarded over long rambling ones. Also, unreadable answers will be given zero (0) so write legibly.

General Hint: We were careful to leave ample space on the test paper to answer each question, so if you find yourself using much more room than what is available, you're probably missing something. Also, remember that hints are just hints: you are not required to follow them!

1: ______/17

2: ______/12

3: ______/14

TOTAL: ______/43

Good Luck!

Question 1. [17 MARKS]

Below is a partially completed class modelling cars. Read it carefully. For this question, you will write code calling methods of the class, and also write additional methods for the class.

```
public class Car {
    private String maker;            // Chrysler, Ford, Toyota, etc.

    private double kmPerLitre;      // the fuel efficiency: how far
                                    // it can go on one litre of gas

    private double litresOfGas;     // gas currently in the tank
    private int tankSize;           // the maximum the tank can hold

    public Car(String m, double k, int t) {
        maker = m;
        kmPerLitre = k;
        tankSize = t;
    }
} // class Car
```

Part (a) [3 MARKS]

In a *single* java statement: construct an object representing a Toyota that has a 50 litre gas tank and gets 15.5km per litre, and declare a reference `c1` to this object.

Part (b) [3 MARKS]

Write a method `fillErUp()` for the `Car` class, which sets the amount of fuel currently in the tank to the maximum amount.

Part (c) [3 MARKS]

Write a method `fuelInTank()` that could be called as demonstrated in the line of code

```
double litres = c1.fuelInTank();
```

and that does what its name suggests.

Question 1. (CONTINUED)

The next 2 parts are about a method `drive()`. The parameter for this method indicates the number of kilometers the user hopes to drive. If there is enough gas for the trip, the method removes that much gas and returns `true`. If there is not enough gas, the method uses up all the gas and returns `false`.
Here is an example of a call to the method. In this example, `c1` is a `Car` reference to an existing `Car` with some gas.

```
if (c1.drive(140.2)) {
    System.out.println("We made it to London!");
} else {
    System.out.println("We ran out of gas!");
}
```

Part (d) [2 MARKS]
Write the prototype for `drive()`.

Part (e) [6 MARKS]
Write the body of `drive()`.

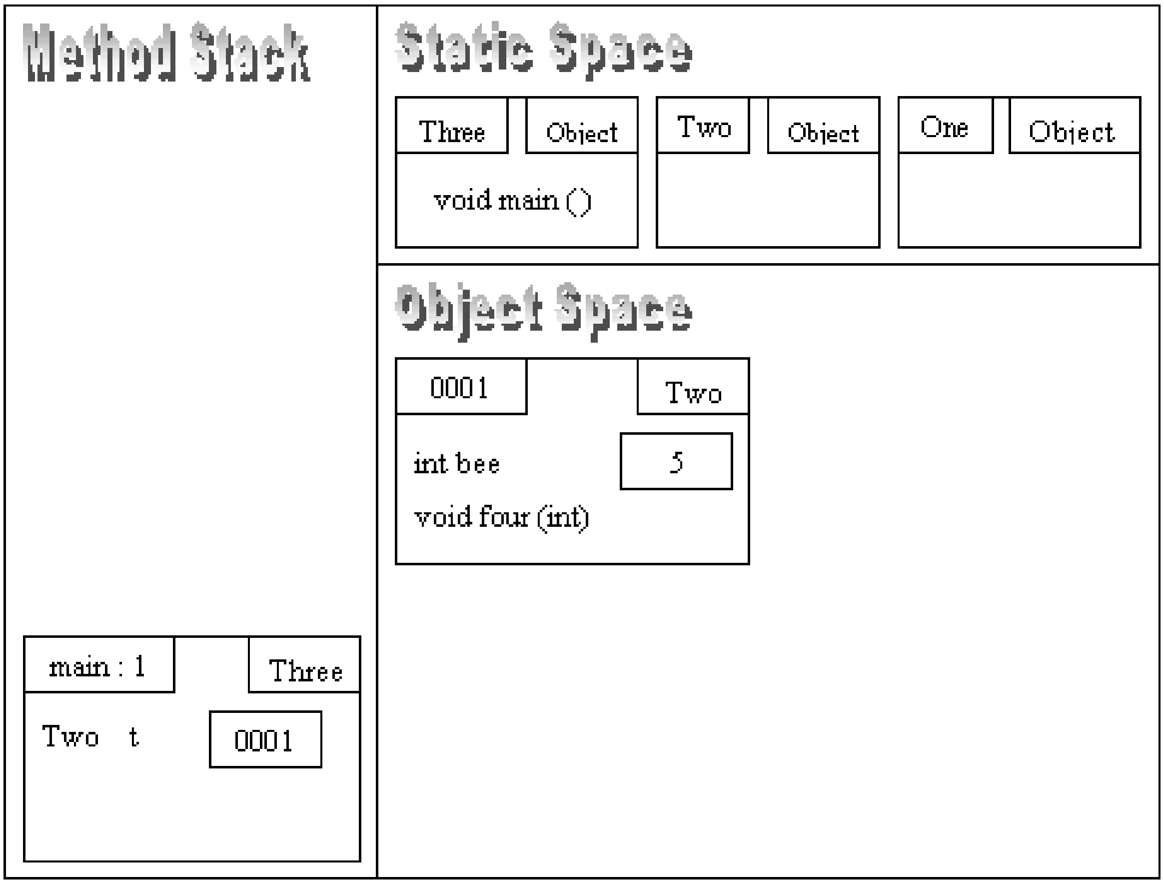
Question 2. [12 MARKS]

The following is a Java program and the memory model that results after executing statement 1 in the `main` method. Update the diagram to show the state of memory after statement 1 in method `four` executes for the first time (this is initiated by the call in statement 3 of method `main`.) As in Assignment 2, you may *cheat* by representing `String`s as text inside variable boxes.

```
class One {
    private String aye;
    private int bee;
    private Two sea;
    public One(String a, int b, Two c) {
        aye = a;
        bee = b;
        sea = c;
    }
    public void three(int d) {
        sea.four(d);        // stmt 1
    }
} // class One
```

```
class Two {
    private int bee;
    public Two(int d) {
        bee = d;
    }
    public void four(int f) {
        bee -= f;          // stmt 1
        int dee = 5;       // stmt 2
    }
} // class Two


public class Three {
    public static void main(String[] args) {
        Two t = new Two(5);               // stmt 1
        One o = new One("won",12,t);      // stmt 2
        o.three(3);                       // stmt 3
    }
} // class Three
```



Question 3. [14 MARKS]

Here is a Java program, which compiles and runs without error messages. Write the output from the program in the space below it.

```
public class Q3 {
    public static void main(String[] args) {
        String s = "Alan Turing";
        String t = s;
        System.out.println(s.substring(1, 3));
        s = s.toUpperCase() + " " + t.toLowerCase();
        System.out.println(t);
        mister(s);
        System.out.println(s);
        Person p1 = new Person("Alan Turing", "Student");
        Person p2 = new Person("Alan Turing", "Computer Scientist");
        Person p3 = p1;
        p3.changeOccupation("Secret Agent");
        p1.printOccupation();
        p2.printOccupation();
        p3.printOccupation();
    }
    public static void mister(String s) {
        s = "Mr. " + s;
        System.out.println(s);
    }
}

public class Person {
    private String name;
    private String occupation;
    public Person(String n, String o) {
        name = n;
        occupation = o;
    }
    public void changeOccupation(String o) {
        occupation = o;
    }
    public void printOccupation() {
        System.out.println(occupation);
    }
}
```