UNIVERSITY OF ABERDEEN **SESSION 2003–04**

DEGREE EXAMINATION

MX3015 Mathematical Computing

Monday 19 January 2004 (9am to 11am)

Only calculators approved by the Department of Mathematical Sciences may be used in this examination. Calculator memories must be clear at the start of the examination.

Marks may be deducted for answers that do not show clearly how the solution is reached.

Answer THREE questions. All questions carry equal weight.

1. A tank in the shape of a cuboid has its rectangular base placed horizontally. The tank can hold an amount, up to its own volume, of liquid. JAVA code for a `Tank` class to model this situation appears at the end of the paper. Read the `Tank` class and answer the following questions.

(a) Explain the purpose of the `package` statement.

(b) List the class variables and class methods and provide a one-line comment about each.

(c) Describe what happens when an attempt is made to add too much liquid to the tank.

(d) Write a method `double getDepth()` for the class `Tank` which returns the depth of liquid currently in the tank and a method `pourFrom(Tank t)` which transfers any liquid from `t` to `this` tank.

(e) Describe in detail what each statement in the `main` method does and say what you expect the output to be.

2. In this question, you are asked to write a JAVA class `Interval` which models closed intervals on the real line like $[1, 3]$ or $[0, \pi/2]$.

(a) Write a minimal `Interval` class, setting up private instance variables with accessor and mutator methods and an appropriate constructor.

(b) Extend your class by writing a method `double length()` which returns the length of the interval, and a method `boolean contains(double x)` which is true precisely when the interval contains x.

(c) Write a static method `Interval unit()` which returns the interval $[0, 1]$.

(d) Add a brief `main` method which constructs a unit interval and the interval $J = [0.5, 3]$. Use your methods to print out the length of J and whether or not J contains the number 2.5. Describe briefly the process necessary to run the `main` method, and the output you expect.

3. **(a)** Describe the secant method for estimating a root of the equation $f(x) = 0$, deriving the formula to obtain the next estimate from two previous ones. Illustrate your answer by obtaining *two* estimates, other than your starting values, for the unique root of the equation

$$4x^2 + x - 3 = 0$$

which lies in the interval $[0, 1]$.

(b) Give a brief description of how JAVA stores the `int` type. Your description should be full enough to explain how arithmetic is done and why the following main method stops after printing 32 lines.

```
public static void main(String argv[]) {
    int n = 0;
    int value = 1;
    while (value !=0) {
        value*=2;
        n++;
        System.out.print("The value is " + value);
        System.out.println(" doubling " + n + " times.");
    }
}
```

Give an indication of the next to the last line that is produced.

4. **(a)** Describe the trapezium rule for estimating a definite integral. Set up the notation carefully, explain the geometrical justification, and derive the formula you would use.

(b) Explain briefly what the following code fragment appears to do, and explain how its function enhances the basic trapezium rule.

```
double adaptive(double epsilon) {
    double estimate;
    double newEstimate;
    int steps = 1;
    double h;
    newEstimate = trapeziumRule(steps);
    do {
        estimate = newEstimate;
        steps *=2;
        h = (upperLimit - lowerLimit)/steps;
        double newOrdinates = 0;
        for (int i = 1; i<steps; i+=2) {
            newOrdinates +=f.at(lowerLimit + h*i);
        }
        newEstimate = 0.5*estimate + h*newOrdinates;
    } while (Math.abs(newEstimate - estimate) > epsilon);
    return newEstimate;
}
```

Assume the class containing this fragment and the `trapeziumRule` method has been set up with suitable instance variables and that `f.at(lowerLimit + h*i)` does what you might expect.

Here are the JAVA *classes referred to in Question 1. The public class is stored in a file named appropriately.*

```
package uk.ac.abdn.maths.mx3015.jan04;

public class Tank {
    private double length;
    private double breadth;
    private double height;
    private double volume;
    private double maxVolume;

    Tank(double l, double b, double h) {
        length = l;
        breadth = b;
        height = h;
        maxVolume = l*b*h;
        volume = 0.0;
    }
    public void addVolume(double v) throws TankOverflowException{
        if (v + volume > maxVolume) {
            throw new TankOverflowException(v,this);
        } else {
            volume +=v;
        }
    }
    public double getVolume(){
        return volume;
    }
    public String toString() {
        return "the tank with length " + length +
               ", breadth " + breadth + " and height " + height;
    }

    public static void main(String argv[]) {
        Tank t = new Tank(3.0,4.0,5.0);
        try {
            t.addVolume(12.0);
        } catch(TankOverflowException e) {
            // Do nothing.
        }
        try {
            t.addVolume(60.0);
            System.out.println("The tank contains " + getVolume());
        } catch(TankOverflowException e) {
            // Do nothing.
        }
    }
    class TankOverflowException extends Exception {
        TankOverflowException(double v, Tank t) {
            System.out.println("Adding volume " + v + " will cause "
                               + t + " to overflow.");
        }
    }
}
```