PRINCE OF SONGKLA UNIVERSITY

FACULTY OF ENGINEERING

Department of Computer Engineering

Final Examination: Semester 2 **Academic Year**: 2005-2006

Date: 25th February, 2006 **Time**: 13:30 – 16:30 (**3 hours**)

Subject Number: 240-321 **Room**: R 300

Subject Title: Advanced Computer Programming Techniques

Lecturer: Aj. Andrew Davison

Exam Duration: 3 hours

This paper has 3 pages.

Authorised Materials:

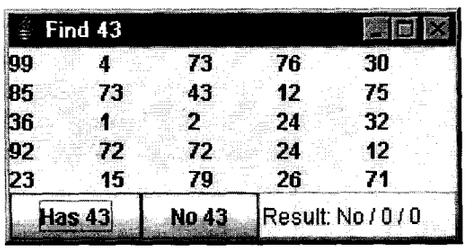
- Writing instruments (e.g. pens, pencils).
- Books (e.g. dictionaries) and calculators are **not** permitted.

Instructions to Students:

- *Answer questions in English.* Perfect English is **not** required.
- Attempt all questions.
- Write your answers in an answer book.
- Start your answer to each question on a new page
- Clearly number your answers.
- Any unreadable parts will be considered wrong.
- When writing programs, use good layout, and short comments; marks will not be deducted for minor syntax errors.
- The marks for each part of a question are given in brackets (...).

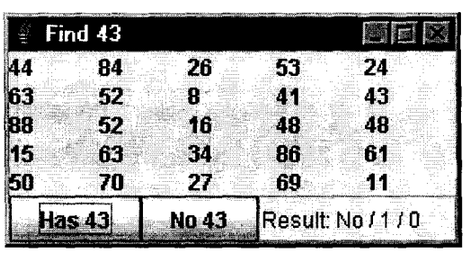
Question 1 (75 marks; 75 minutes)

Implement the Find43 application. It's GUI interface has the form:



A grid of 25 random integers between 0 and 99 are shown in the top half of the window. The player must press one of two buttons depending on whether "43" is in the grid. In this example, the user pressed the "No 43" button, which is incorrect since there is a "43" in the numbers grid.

The textfield on the right is then updated with the game number and the score, and a new grid of random integers is displayed:



If the user had pressed the correct button ("Has 43"), then the text field would show "Result: Yes / 1 / 5". The first number is the number of games played, the second is the total score. The score depends on the amount of time that has passed since the grid was generated and when the user pressed on a button.

The user can keep playing until he presses the window's close box.

Some statistics are printed to standard output as the game exits:

```
No. of Games: 2
Total Score: 0
Average Score: 0
```

Hints

The GUI should use a BorderLayout, GridLayout, and JPanels for organizing the components. You may need other layout managers as well.

The grid of random numbers should be represented by an array of JLabels.

Random integers in the range 0 to 99 can be generated using a Random object and its nextInt() method:

```
Random rand = new Random();
int num0 = rand.nextInt(100);   // random integer
int num1 = rand.nextInt(100);   // another random integer
```

An ActionListener should be attached to both buttons.

printed.

Your application should **not** have a GUI interface. Input should be read from standard input (the keyboard) using a suitable input reader class, and output should use `System.out`.

--- End of Examination ---

Use System.currentTimeMillis() to get the current time in milliseconds.

The close box should be monitored by a WindowListener, which prints statistics before making the application exit.

Question 2

(55 marks; 55 minutes)

Explain the meaning of the following Java thread issues:

a) race condition; (15)
b) mutual exclusion; (20)
c) condition synchronization. (20)

Each answer should include small code fragments and diagrams if they are useful.

Question 3

(50 marks; 50 minutes)

Write a Java application which reads in a calculation from the keyboard, prints the result of evaluating it, then exits. A typical input calculation would be:

23 / 4.7

The output would be: 4.8936

You may assume that a calculation occurs on a single line, and has the form:

<double> <operator> <double>

The operator should be +, -, *, or /. The doubles and the operator will always be separated by spaces.

Your code should deal with the possibility that:

- the operands are not doubles
- the operator is not +, -, *, or /
- division by zero may occur. The division by zero problem must be handled by a **new exception class**, created by you.

The output should be a double, but formatted so that only at most 4 decimal places are printed.

Your application should **not** have a GUI interface. Input should be read from standard input (the keyboard) using a suitable input reader class, and output should use `System.out`.

--- End of Examination ---