

PRINCE OF SONGKLA UNIVERSITY
FACULTY OF ENGINEERING
Department of Computer Engineering

Final Examination: Semester 2

Academic Year: 2006-2007

Date: 22nd February, 2007

Time: 13.30 – 16.30 (3 hours)

Subject Number: 240-321

Room: A400

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.

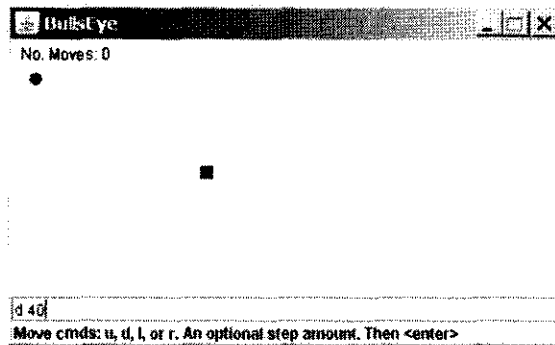
Instructions to Students:

- *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

(70 marks; 70 minutes)

Implement the BullsEye game:



The aim of the game is to move the circle until it touches the square.

A move command is entered in the text field at the bottom of the window. The move commands are 'u', 'd', 'l', and 'r', followed by a number, and <enter>. 'u' moves the circle up by the specified number of pixels, 'd' moves it down, 'l' moves it left, and 'r' sends it right.

In the screenshot, the user has just typed the command `d 40` which will move the circle downwards by 40 pixels.

The number of move commands entered by the user is displayed at the top-left of the panel, and is updated after each move has been carried out.

When the circle touches the square, a "FINISHED!!" message is displayed in the panel.

When the game starts, the circle is positioned at a random location in the game panel, and the square is placed near the center.

- a) Implement the top-level window as a class called `BullsEye`, which should be a subclass of `JFrame`. It should create the GUI and listen for user input. (35)
- b) Implement the game panel as a class called `BullsEyePanel`, which should be a subclass of `JPanel`. It should manage the drawing of the square, the circle, and the text messages about the number of moves and being finished. (35)

Note: `BullsEye` should communicate with the `BullsEyePanel` by calling its `movePlayer()` method, which moves the circle in a particular direction, by a specified number of pixels.

Question 2

(40 marks; 40 minutes)

Briefly describe the **four** main layout managers used in Java. Include **small** code fragments and diagrams if they are useful.

Question 3

(50 marks; 50 minutes)

The following code fragment searches through an array of integers, called `vals[]`, looking for the number stored in `searchNum`. The duration of the search is printed at the end.

```
int[] vals = new int[size];
// fill vals[] with random integers (code not shown)

long startTime = System.nanoTime();

for (int i=0; i < size; i++) {
    if (vals[i] == searchNum)
        System.out.println("Found at position " + i);
}

long duration = (System.nanoTime() - startTime)/1000000L;
System.out.println("Duration (ms): " + duration);
```

- a) Create a subclass of `Thread` called `Searcher` which searches the `vals[]` array in a specified index range. For example, you can call `Searcher` with the numbers 0 and 999 to make it search for `searchNum` between `vals[0]` and `vals[999]`. (15)
- b) Rewrite the code fragment above to generate multiple `Searcher` threads to search over different ranges of the `vals[]` array. For example, the first thread will search between `vals[0]` and `vals[999]`, the second one between `vals[1000]` and `vals[1999]`, and so on. *Note*: do not implement the code for initializing `vals[]`. (10)
- c) Will the threaded approach that you've implemented in parts (a) and (b) be faster or slower than the non-threaded approach in the original code fragment? Explain your answer in words. (10)
- d) Describe the *four* main kinds of threaded programming. Do **not** include any code fragments in your answer. (15)

Question 4

(20 marks; 20 minutes)

- a) Write a `main()` method that reads in a single integer from the *keyboard*. Do **not** use the `Console`, `EasyIn`, or `Scanner` classes. (10)
- b) Write a `main()` method that uses the `Scanner` class to read in floats from a *text file*. The floats should be added, and printed to standard output. (10)

--- End of Examination ---