

UNIVERSITY COLLEGE LONDON

University of London

EXAMINATION FOR INTERNAL STUDENTS

For The Following Qualification:-

B.Sc.

Information Studs. B2: Programming 2

COURSE CODE : INSTB002

UNIT VALUE : 0.50

DATE : 03-MAY-06

TIME : 14.30

TIME ALLOWED : 2 Hours 30 Minutes

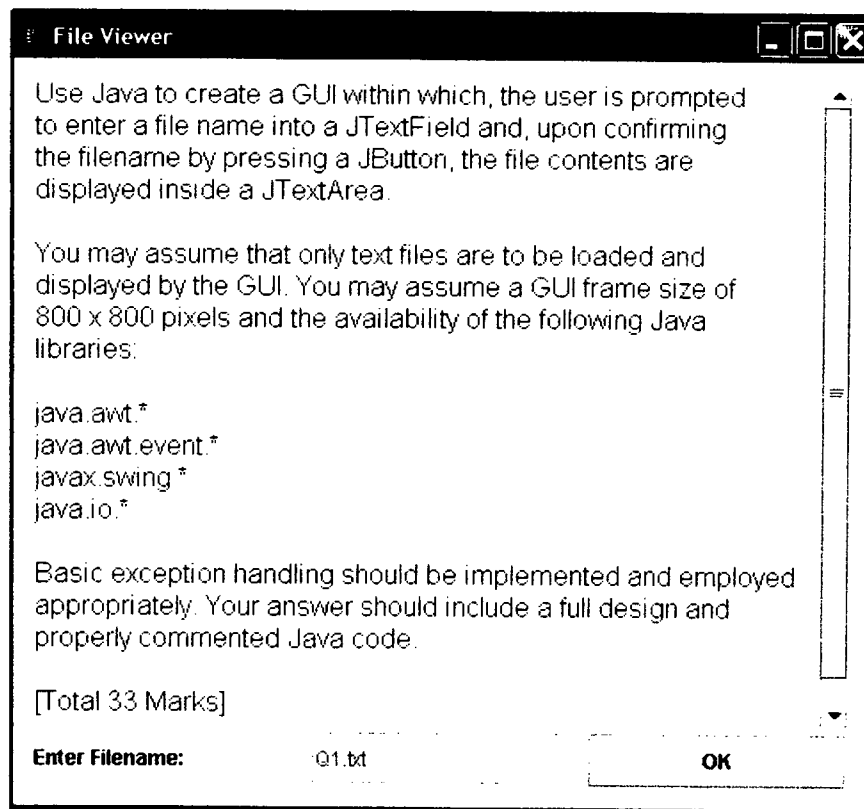
INSTB002: Java II, May 2006

Answer any THREE questions, including question 1 or question 2 (or both)

Marks for each part of each question are indicated in square brackets.

Calculators are not permitted.

1. Use Java to create a *GUI* within which the user is prompted to enter a file name into a *JTextField* and, upon confirming the filename by pressing a *JButton*, the file contents are displayed inside a *JTextArea* (see example below).



You may assume that only text files are to be loaded and displayed by the GUI. You may assume a GUI frame size of 800 x 800 pixels and the availability of the following Java libraries:

```
java.awt.*           java.awt.event.*
javax.swing.*       java.io.*
```

Basic *exception handling* should be implemented and employed appropriately. Your answer should include a full design and properly commented Java code.

[Total 33 Marks]

2. Use Java to create a class called `bookList` that is able to store simple information regarding a collection of books at the *nodes* of a *bi-directional linked list*. Each node in the list should be able to store:
- the book title (String)
 - the author name (String)
 - the recommended retail price (double)
 - the year of publication (int)

Both the list and node classes should possess an appropriate constructor to initialise the list and the information stored at each node of the list respectively.

The `bookList` (list) class should possess an `add` method that adds new nodes to a specified position within the list (if such a position exists).

Your answer should include a full design and properly commented Java code.

[Total 33 Marks]

- 3a. Describe and compare *static* data structures, e.g. *arrays*, and *dynamic* data structures, e.g. *linked lists*. Your answer should include illustrative examples of Java code and/or appropriate pseudo-code.

[13 marks]

- 3b. The class `twoInts`, is shown below:

```
public class twoInts
{
    private int A, B;
    public twoInts(int newA, int newB)
    {
        A = newA;
        B = newB;
    }
    public int intSum()
    {
        return(A + B);
    }
}
```

Implement the `BubbleSort` (otherwise known as `ExchangeSort`) algorithm to sort an `ArrayList` of `twoInts` objects in ascending order of their summed member variables, i.e. in ascending order of $(A+B)$ for each `twoInts` object.

[20 marks]

[Total 33 Marks]

4. Write an essay on the implementation of search and sort algorithms in Java. Your answer should detail both the manner in which popular search and sort algorithms are implemented in Java and the benefits provided by their employment over linear search and sort procedures. Your answer must include detailed illustrative examples written in Java and should also cover the concept of recursion and its use in algorithms such as *Binary search* and *QuickSort*.

[Total 33 Marks]

- 5a. Briefly describe each of the following features of Java and explain how they are used, including illustrative examples of Java code: *the System.err stream*, *the StringTokenizer class*, *serialization*, *exception propagation* and *the throws clause*.

[25 marks]

- 5b. Describe conversion of the standard `System.in` I/O stream to receive and capture buffered keyboard input. Your answer should include an illustrative example of Java code and/or appropriate pseudo-code.

[8 marks]

[Total 33 Marks]

END OF PAPER