2B11 Exam 2001

2.5 Hours

Contact:
Dr. Graham Roberts
Department of Computer Science
x3711, room 217
G.Roberts@cs.ucl.ac.uk

Answer ALL questions from Part I and TWO questions from Part II

Part I

1. a) An ordered collection stores objects in order, and uses an ordering relation to compare object values to determine their order (less than, equal, or greater than). Write a Java interface declaration for an ordering relation type.

[3 marks]

b) Write a Java interface declaration for ordered collections that support the operations get, insert, remove, find and size. Include exception information where necessary.

[6 marks]

c) Write, using pseudocode, an ordered list class that implements your ordered collection interface and stores objects of classes that implement your ordering relation interface. Each method body should show how the ordered list is manipulated. Methods should throw exceptions where required.

[15 marks]

d) Outline a JUnit style test class for your ordered list class.

[6 marks]

[Total 30 marks]

2. **This question is concerned with the generation of code by a recursive-descent compiler for a simple stack-based machine.**

a) A commonly used form of the for-loop in Java is

```
for (assignment-exp; boolean-exp; assignment-exp)
        statement
```

Using the VSCC methodology,

i) create an annotated syntax diagram for this construct, and

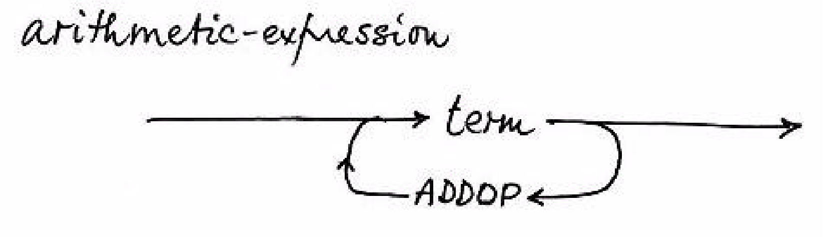
ii) implement the annotated diagram as a Java method.

Also

iii) explain how the requirement to generate code ‘on the fly’ influences your implementation.

[12 marks]

b) Explain why the following syntax diagram is not suitable as a basis for generating code for the target machine.



[4 marks]

c) Give two examples of where more than one token of lookahead might be needed in parsing Java programs, and explain why.

[4 marks]

[Total 20 marks]

Part II

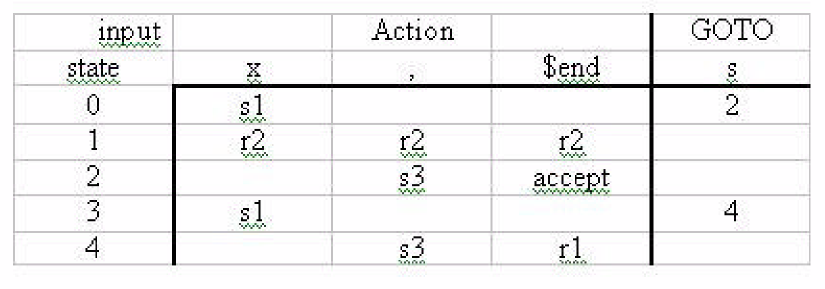
Answer Two questions

3. a) Give an example of the dangling-else ambiguity. Discuss the approaches that have been taken to resolve it in different programming languages, and how these approaches are made apparent to users of the languages.

[6 marks]

b) The following S-R table was derived from output produced by yacc when given the input:

```
%%
s : s ',' s
s : 'x'
```



input		Action		GOTO
state	x	,	$end	s
0	s1			2
1	r2	r2	r2	
2		s3	accept	
3	s1			4
4		s3	r1	

i) Explain how the table entries s1 and r2 are to be interpreted. Also explain how a blank entry is to be interpreted.

ii) Show the moves which the parser would make when parsing the sentence

x_1 , x_2 , x_3

[Hint: remember to initialise the stack with $ 0, corresponding to state 0.]

iii) Construct the corresponding parse tree, labelling each node with the number of the move from the parse.

[10 marks]

During the processing of the grammar, yacc reported a conflict.

iv) Explain the nature of this conflict.

v) Show how the grammar could be modified to remove the conflict, without affecting the structure of the parse tree.

[9 marks]

[Total 25 marks]

TURN OVER

4. a) Explain each of the following:
   thread, synchronized method, finally, event object, event listener, layout manager

[3 marks each]

b) Explain what a design pattern is and how patterns relate to software architecture.

[7 marks]

[Total 25 marks]

5. a) Explain the purpose of a package.

[5 marks]

b) Consider the design of a comprehensive system to support the management and running of an airport. It needs to provide many services, including flight information, baggage handling, passenger check-in, security, flight information display, personnel management, payroll, stock control for supplies and so on. Some services need to integrate with airline computer systems, for example to obtain information about flights and times.

Propose a design for such a system, clearly identifying an overall architecture, and the use of packages, interfaces and components, along with their roles and behaviours. Include a diagram to show the overall structure.

[20 marks]

[Total 25 marks]

END OF PAPER