final
1st term Form A

King Saud University
College of Computer and Information Sciences
Computer Science Department

Final Exam

Academic Year: 2016/2017
First Semester

Master Program
Course Name/No. : Programming languages and Compilation / CSC 340

Exam Date: 18/1/2017:
Exam Time: 180 minutes: From 8:00 To 11:00 am.

Total Number of Pages: 6 pages (including this cover page)

Student Name	
Student ID.	

Exercise No.	Full Mark	Student Mark
1.	**30**	
2.	**10**	
Total	**40**	

Q1 Answer: Please copy your answer to question 1 to this table.

1.	2.	3.	4.	5.	6.	7.	8.	9.	10
b	c	d	c	c	a	b	a	d	b

11.	12.	13.	14.	15.	16.	17.	18.	19.	20
b	d	c	~	a	d	d	d	a	b

21.	22.	23.	24.	25.	26.	27.	28.	29.	30
b	a	d a or c	b	d	c	a	c	d	a

Q1) Put a circle around the best answer for each of the following then copy your answer to the table on the first page (only the table will be graded). (30 grades)

1. The main advantage of interpreters over compliers is
   a. efficiency
   b. portability of programs
   c. reliability
   d. a+b
2. Java programs are partly compiled and partly interpreted to achieve
   a. portability
   b. efficiency
   c. portability and efficiency
   d. reliability
3. Which of the following programming language features affect the reliability of a programming language
   a. Data types
   b. Exception handling
   c. Orthogonality
   d. All of the above
4. Which of the following programming language features affects the writability of the programming language
   a. Data types
   b. Exception handling
   c. Orthogonality
   d. All of the above
5. Which of the following regular expressions describe the language over the alphabet {a,b} that consists of all strings that contain at least one 'a'.
   a. $b^*aa^*b^*$
   b. $b^*ab^*a^*b^*$
   c. $b^*a(a^*b^*)^*$
   d. none of the above
6. Which of the following regular expressions describe the language over the alphabet {a,b} that consists of all strings that begin and end with the same letter.
   a. a(b*a)*+b(a*b)*
   b. ab*a+ba*b
   c. ab*a*b*a+ba*b*a*b
   d. none of the above
7. Converting an NFA that consists of n states to an equivalent DFA would result in a DFA that contains
   a. exactly 2^n states
   b. at most 2^n states
   c. at least 2^n states
   d. n! states
8. A table that represents an NFA of 3 states and deals with 2 terminal symbols consists of
   a. 3 rows and 3 columns
   b. 3 rows and 2 columns
   c. 2 columns and 3 rows
   d. 5 rows and 2 columns

9. Which of the following methods require eliminating left recursion
   a. Recursive descent parsing
   b. LL(1) parsing
   c. Bottom up parsing
   d. a+b
10. Which of the following methods may require left factoring a grammar
   a. Recursive descent parsing
   b. LL(1) parsing
   c. Bottom up parsing
   d. a+b
11. Which of the following productions assigns left associativity for +
   a. E→ int+E
   b. E→E+int
   c. E→E+E|int
   d. None of the above
12. Which of the following grammars is not ambiguous
   a. E→ id+E | id*E | (E) | id
   b. E→E+E | E * E | (E) | id
   c. E→E+T|T
      T→ T*F | F
      F→ (E) | id
   d. a+c

Question 13-16 are related to the following grammar

E→E+T|T
T→ T*F | F
F→ (E) | id

13. What is follow set of E?
   a. { + }
   b. { +,) }
   c. { $, +,) }
   d. { $, + ,), ε }
14. What is follow set of F?
   a. { + }
   b. { +,) }
   c. { $, +,) }
   d. { $, + ,), ε }
15. What is first set of E?
   a. { (, id }
   b. { (}
   c. {id}
   d. { (, id, ε }
16. In the LL(1) table for the above grammar, the entry that contains E+T is
   a. row E column (
   b. row E column id
   c. row E column $
   d. a+b

The following assumptions are related to questions 17-19

Assume that a grammar contains the production E→T+F|TF,
Assume also that first(T)={id, ε}, fist(F)= { * , ε }, follow(E)={id,$} and follow(T)={+, $}

17. In the corresponding LL(1) table, which entry contains TF
    a. row E, column id
    b. row E, column +
    c. row E, column *
    d. a and b
18. which entry in the LL(1) table contains ε
    a. row E, column id
    b. row E, column +
    c. a and b
    d. none of the above
19. The grammar is not LL(1) grammar because the entry at
    a. row E column id is multiply defined
    b. row E column $ is multiply defined
    c. row E column + is multiply defined
    d. the grammar is LL(1) because no entry is multiply defined
20. Bottom-up parsers are
    a. More efficient than top-down parsers
    b. More general than top-down parsers
    c. Less general than top-down parsers
    d. a and b

The following assumptions are related to questions 21-...

Assuming that the DFA for recognizing the viable prefixes of a grammar contains the items
E→T.+F
T→F.

21. Which types of conflicts does the above grammar contain
    a. Reduce-reduce conflict
    b. Shift reduce conflict
    c. No conflicts
    d. A+b
22. The grammar is not SLR(1) grammar if
    a. follow(E)={+,$}
    b. follow(T)={+}
    c. first(F)={+}
    d. none of the above
23. If the top of the stack contains + and the next input token is +, then the SLR parsing algorithm
    a. pops + from the stack
    b. reports and error
    c. moves to the next input token
    d. a+b

a + c

24. If the top of the stack contains + and the next input token is *, then the SLR parsing algorithm
    a. pops + from the stack
    b. reports and error
    c. moves to the next input token
    d. a+b
25. Which of the following errors are not detected by a parser
    a. Undeclared identifier
    b. Unexpected data type
    c. Identifier declared more than once
    d. All of the above

Consider the following rule to answer questions 27- 29

$$\frac{\begin{array}{l} f \text{ is an identifier.} \\ f \text{ is a non-member function in scope S.} \\ f \text{ has type } (T_1, \ldots, T_n) \rightarrow U \\ S \vdash e_i : R_i \text{ for } 1 \le i \le n \\ R_i \le ?? \text{ for } 1 \le i \le n \end{array}}{S \vdash f(e_1, \ldots, e_n) : ?}$$

26. What should be written in place of the double question marks (??) in the rule to make correct?
    a. U
    b. f
    c. T_i
    d. None of the above
27. What should be written in place of the question mark (?) in the rule to make correct?
    a. U
    b. f
    c. T_i
    d. None of the above
28. Assuming that a primitive type is convertible to itself, then the above inference rule is applicable for
    a. Referenced types
    b. Primitive types
    c. a and b
    d. None of the above
29. In a stack machine, the heap section of the memory is dedicated for
    a. Local variables
    b. Actual arguments
    c. a and b
    d. dynamically allocated objects
30. The memory section allocated to global variables is allocated
    a. Statically
    b. Dynamically
    c. At execution time
    d. None of the above

Q2) a) Write the code generator segment that deals with the if statement (4 grades)

cgen(if e1=e2 then e3 else e4)

```
cgen(if e1=e2 then e3 else e4)
    cgen(e1)
    sw $a0 0($sp)
    addiu $sp $sp -4
    cgen(e2)
    lw $t1 4($sp)
    addiu $sp $sp 4
    beq $a0 $t1 true_branch
false_branch:
    cgen(e4)
    b end_if
true_branch:
    cgen(e3)
end_if:
```

b) What code will be generated for the following procedure? i.e. what would be produced by the compiler? (6 grades)

def f(x,y) =
if x=y then x else 0

```
f_entry:
    move $fp $sp
    sw $ra 0($sp)
    addiu $sp $sp -4
    # cgen(...)
    lw $a0 4($fp)   # x
    sw $a0 0($sp)   } push x
    addiu $sp $sp -4
    lw $a0 8($fp)   # y
    lw $t1 4($sp)   } pop x
    addiu $sp $sp 4
    beq $a0 $t1 true_branch
false_branch:
    li $a0 0
    b end_if
true_branch:
    lw $a0 4($fp)
end_if:
    lw $ra 4($sp)
    addiu $sp $sp 16
    lw $fp 0($sp)
    jr $ra
```