

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

Final Exam
Academic Year: 2017/2018
First Semester
Master Program
Course Name/No. : Programming languages and Compilation / CSC 340
Exam Date: 3/1/2018:
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.	

<u>Exercise No.</u>	<u>Full Mark</u>	<u>Student Mark</u>
1.	<u>30</u>	
2.	<u>10</u>	
<u>Total</u>	40	

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

[illegible]

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. 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. $b^*a(a^*b^*)^*$
 - c. $b^*ab^*a^*b^*$
 - d. none of the above
2. 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. ab^*a+ba^*b
 - b. $a(b^*a)^*+b(a^*b)^*$
 - c. $ab^*a^*b^*a+ba^*b^*a^*b$
 - d. none of the above
3. 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 least 2^n states
 - c. at most 2^n states
 - d. $n!$ states
4. A table that represents an NFA of 3 states and deals with 2 terminal symbols consists of
 - a. 3 rows and 2 columns
 - b. 3 rows and 3 columns
 - c. 2 rows and 3 columns
 - d. 5 rows and 2 columns
5. Which of the following methods require eliminating left recursion
 - a. Recursive descent parsing
 - b. LL(1) parsing
 - c. $a+b$
 - d. Bottom up parsing
6. Which of the following methods may require left factoring a grammar
 - a. Recursive descent parsing
 - b. Bottom up parsing
 - c. LL(1) parsing
 - d. $a+b$
7. Which of the following productions assigns left associativity for $+$
 - a. $E \rightarrow \text{int}+E$
 - b. $E \rightarrow E+E \mid \text{int}$
 - c. $E \rightarrow E+\text{int}$
 - d. None of the above
8. Which of the following grammars is not ambiguous
 - a. $E \rightarrow \text{id}+E \mid \text{id}^*E \mid (E) \mid \text{id}$
 - b. $E \rightarrow E+E \mid E^*E \mid (E) \mid \text{id}$
 - c. $E \rightarrow E+T \mid T$
 $T \rightarrow T^*F \mid F$
 $F \rightarrow (E) \mid \text{id}$
 - d. $a+c$

Question 9-12 are related to the following grammar

$E \rightarrow E+T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid \text{id}$

9. What is the follow set of E?
 - a. $\{+\}$
 - b. $\{\$, +,)\}$
 - c. $\{+,)\}$
 - d. $\{\$, +,), \epsilon\}$
10. What is the follow set of F?
 - a. $\{+\}$
 - b. $\{\$, +,), *\}$
 - c. $\{+,)\}$
 - d. $\{\$, +,), \epsilon\}$
11. What is first set of E?
 - a. $\{(\}$
 - b. $\{(\, \text{id}\}$
 - c. $\{\text{id}\}$
 - d. $\{(\, \text{id}, \epsilon\}$
12. 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 13-15

Assume that a grammar contains the production $E \rightarrow T+F \mid TF$.

Assume also that $\text{first}(T)=\{\text{id}, \epsilon\}$, $\text{first}(F)=\{*, \epsilon\}$, $\text{follow}(E)=\{\text{id}, \$\}$ and $\text{follow}(T)=\{+, \$\}$

13. 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**
14. which entry in the LL(1) table must contain ϵ
 - a. row E, column id
 - b. row E, column +
 - c. a and b
 - d. **none of the above**
15. The grammar is not LL(1) grammar because the entry at
 - a. row E column \$ is multiply defined
 - b. **row E column id is multiply defined**
 - c. row E column + is multiply defined
 - d. the grammar is LL(1) because no entry is multiply defined

The following assumptions are related to questions 17-20

Assuming that the DFA for recognizing the viable prefixes of a grammar contains the items

$E \rightarrow T. + F$

$T \rightarrow F.$

16. Which types of conflicts does the above grammar contain
 - a. Reduce-reduce conflict
 - b. No conflicts
 - c. Shift reduce conflict
 - d. A+b
17. The grammar is not SLR(1) grammar if
 - a. $\text{follow}(T) = \{+\}$
 - b. $\text{follow}(E) = \{+, \$\}$
 - c. $\text{first}(F) = \{+\}$
 - d. none of the above
18. 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+c
19. If the top of the stack contains + and the next input token is *, then the SLR parsing algorithm
 - a. pops + from the stack
 - b. moves to the next input token
 - c. reports and error
 - d. a+c
20. 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 22- 24

f is an identifier.
 f is a non-member function in scope S .
 f has type $(T_1, \dots, T_n) \rightarrow U$
 $S \vdash e_i : R_i$ for $1 \leq i \leq n$
 $R_i \leq ??$ for $1 \leq i \leq n$

 $S \vdash f(e_1, \dots, e_n) : ?$

21. What should be written in place of the double question marks (??) in the rule to make correct?
 - a. U
 - b. T_i
 - c. f
 - d. None of the above

22. What should be written in place of the question mark (?) in the rule to make correct?
- a. f
 - b. U
 - c. T_i
 - d. None of the above
23. Assuming that a primitive type is convertible to itself, then the above inference rule is applicable for
- a. Referenced types
 - b. Primitive types
 - c. None of the above
 - d. a and b
24. In a stack machine, the heap section of the memory is dedicated for
- a. Local variables
 - b. Actual arguments
 - c. dynamically allocated objects
 - d. a and b
25. The memory section allocated to global variables is allocated
- a. Dynamically
 - b. Statically
 - c. At execution time
 - d. None of the above
26. Bottom-up parsers are preferred because they are
- a. more efficient than top-down parsers.
 - b. more general than top-down parsers.
 - c. simpler than top-down parsers.
 - d. All of the above
27. Recursive descent parsers are not used in real compilers because they are
- a. too complicated
 - b. require left-factorization
 - c. too inefficient
 - d. all of the above
28. One of the following operations is not used on symbol tables
- a. push scope
 - b. insert symbol
 - c. pop symbol
 - d. look-up symbol
29. A type error is detected by
- a. The parser
 - b. The semantic analyzer
 - c. The lexical analyzer
 - d. The code generator
30. An SLR parser uses a DFA to determine
- a. If what it is on the stack is a viable prefix of the handle
 - b. If what is on the stack is the handle
 - c. What to action to execute next shift or reduce
 - d. a + c

Q2) Assuming that x is the first argument, write down the code that will be generated (by cgen) for the following expressions (i.e. what would be the output of the compiler?).

a) **$3+1$** (4 grades)

b) **if $x=1$ then x else $f(1,0)$** (6 grades)

