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→ int+E
   b. E→E+E|int
   c. E→E+int
   d. None of the above
8. 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 9-12 are related to the following grammar**

**E→E+T|T**
**T→ T*F | F**
**F→ (E) | id**

9. What is the follow set of E?
   a. { + }
   b. { $, +, ) }
   c. { +, ) }
   d. { $, + , ), ε }
10. What is the follow set of F?
   a. { + }
   b. { $, +, ),*}
   c. { +, ) }
   d. { $, + , ), ε }
11. What is first set of E?
   a. { ( }
   b. { ( , id }
   c. {id}
   d. { (, id, ε }
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→T+F|TF.**

**Assume also that first(T)={id, ε}, fist(F)= { * , ε }, follow(E)={id,$} and 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→T.+F**
**T→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. follow(T)={+}
    b. follow(E)={+,$}
    c. 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

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

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

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

**King Saud University**
**College of Computer and Information Sciences**
**Computer Science Department**
**CSC 340 Mid-1**

Key

mid term 2016 - 2017

Student Name: Student Number:

Q1) (14 marks) Put a circle around the symbol of the best answer for each of the following

1) Which of the following features must be available in a programming language to be suitable for writing embedded systems
   a) Provide constructs for low-level control over hardware.
   b) Use floating point representation
   c) Have features to analyze data
   d) All of the above.
2) An orthogonal programming language
   a. Has a relatively small set of primitive constructs that can be combined in a relatively small number of ways to get the desired results.
   b. Has a fewer number of exceptions because every possible combination is legal.
   c. Is good with respect to reliability
   d) All of the above
3) Which of the following features a programming language need to have in order to be good with respect to reliability
   a. Data types
   b. Support for abstraction
   c. Exception handling
   d) all of the above
4) Having a small number of manageable features and constructs, makes a programing language good with respect to
   a. Readability
   b. Writability
   c. Reliability
   d) All of the above
5) A language with too many operators and special symbols is
   a. good with respect to writability
   b. bad with respect to reliability
   c. bad with respect to readability
   d) all of the above
6) Java is partly compiled and partly interpreted (hybrid implementation) makes is
   a) Good for writing portable programs
   b. Good for writing efficient programs
   c. Good for reducing the cost of training
   d. All of the above
7) Classes, inheritance and polymorphism designed in programming languages as a result of
   a. The Von-Neumann architecture
   b) Programming methodologies
   c. People preferences
   d. None of the above