American University in Cairo *Date: March 22, 2015*
Computer Science and Engineering Department

CSCE 3104- Concepts of Programming Languages-Midterm Exam

Name:_____ ***ID: _____***

Question	Full Mark	Grade
1	25	
2	10	
3	10	
4	10	
5	10	
Total	65	

Name: _____ ID: _____

Question 1: Select one correct answer from the following:

1. Syntax of programming languages is the study of:
    a. Vocabulary.
    b. Grammar.
    c. Meaning.
    d. (a) and (b).
    e. All of the above.
2. Semantics of programming languages is the study of:
    a. Meaning.
    b. Meaning and Grammar.
    c. Vocabulary.
    d. All of the above.
    e. None of the above
3. Regular expressions can be used to specify:
    a. Grammatical structure.
    b. Vocabulary.
    c. Natural language statements.
    d. Context sensitive grammar.
    e. All the above.
4. Regular expressions are composed mainly of :
    a. Concatenation.
    b. Selection.
    c. Repetition.
    d. Conditions.
    e. (a) and (b).
    f. (a) (b) and (c)
5. A grammar is said to be ambiguous if:
    a. Two production rules have the same meaning.
    b. There are two ways of writing a given production rule.
    c. A string in the language described by the grammar has two parse trees.
    d. There are two non-terminals on the left hand side of the production.
    e. None of the above.
6. Using natural language grammars in programming language can:
    a. Eliminate ambiguity.
    b. Create ambiguity.
    c. Limit expressiveness.
    d. Limit extensibility.
    e. None of the above.
7. A programming language is:
    a. A notational system.
    b. Describes computation in machine readable format.
    c. Describes computation in human readable format.
    d. Part of its specification indicates whether it is compiled or interpreted.
    e. All of the above.
    f. (a) (b) and (c).
8. A GOTO statement can be catastrophic because:
    a. It can create multiple exit points in blocks.
    b. It can create multiple entry points in blocks.
    c. Increases maintainability.
    d. All the above.
    e. (a) and (b).

Name: _____ ID: _____

9. Logic programming languages are considered:
    a. High level.
    b. Low level.
    c. Close to assembly.
    d. None of the above.

10. Interpreted programming languages are typically more efficient than compiled languages:
    a. True.
    b. False.
    c. Doesn't make a difference.

11. Interpreted programming languages typically exhibit more of:
    a. Static features.
    b. Dynamic features.
    c. Polymorphic features.
    d. Low level features.

12. Describe, in English, the language defined by the following grammar (non-terminals are in italics):

$S \rightarrow$ m A B C d
$A \rightarrow$ a A | a
$B \rightarrow$ b B | b
$C \rightarrow$ c C | c

    a. One or more m's followed by zero or more a's, then zero or more b's then zero or more c's.
    b. One m followed by zero or more a's, then zero or more b's then zero or more c's.
    c. One m followed by one or more a's, then one or more b's then one or more c's followed by a d.
    d. One or more m's followed by zero or more a's, then zero or more b's then zero or more c's.
    e. None of the above.

13. Which string proves that the following grammar is ambiguous:

$S \rightarrow A$
$A \rightarrow A$ - A | id
$id \rightarrow letter\ id$ | $letter$
$letter \rightarrow$ a | b

    a. ab
    b. aaaabbb
    c. a+b
    d. a-b-a
    e. a*a*b
    f. None of the above.

14. The following code is written in LISP using the functional paradigm. What does the following LISP code achieve (given N>=1)?

```
(define xyz (N)
        (if (= N 1) 1
                (* N (xyz (- N 1)))))
```

    a. Factorial (N)
    b. Zero
    c. N - (N-1) - (N-2) - … - 0
    d. N + (N-1) + (N-2) + … + 0
    e. Average (N)
    f. None of the above.

15. What is the associativity of addition and multiplication in the following grammar:

    expr -> term + expr | term
    term -> factor * term | factor
    factor -> (expr) | number
    number -> number digit | digit
    digit -> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

    a. Right, left.
    b. Right, right.
    c. Left, right.
    d. Left, left.

16. Why is the following grammar considered to be context sensitive, and not context free:

    $S \rightarrow$ abc | aSBc
    c$B \rightarrow B$c
    b$B \rightarrow$ bb

    a. Upper case S in the grammar.
    b. Upper case S and B in the grammar.
    c. No single non-terminal on the left side.
    d. No single terminal on the right side.
    e. None of the above.

17. A regular expression for a language composed of **one or more** of the following: even number of 0's followed by an odd number of 1's.
    a. ((00)* 1 (11)*)*
    b. ((00)+1 (11)*)+
    c. (00)+ 1 (11)+
    d. (00) 1 (11)*
    e. None of the above

18. A typical example of a delimiter used in free form languages is:
    a. #
    b. //
    c. spaces
    d. semicolons
    e. All of the above.

19. To eliminate ambiguity of precedence, the following can be used:
    a. Prefix notation.
    b. Postfix notation.
    c. Full parenthesis of expressions.
    d. In order notation.
    e. All of the above.
    f. (a), (b), and (c).

20. Dynamic type binding is done when
    a. Variables are allocated by a programming language primitive
    b. Variables are specified through an assignment statement
    c. Variables are declared in functions or procedures
    d. Variables are bound to storage during runtime time

21. The lifetime of a variable is the time during which
    a. The variable is bound to a specific type
    b. The variable is bound to a specific name
    c. The variable is bound to a specific value
    d. None of the above

22. An implicit declaration is a default mechanism for:
    a. Specifying types of variables
    b. Allocating a variable implicitly
    c. Specifying a default value to a variable
    d. None of the above

23. A binding is dynamic if
    a. It occurs before run time and remains unchanged throughout program execution
    b. It occurs before run time and changes throughout program execution
    c. It occurs during runtime and remains unchanged throughout program execution
    d. It occurs during runtime and changes through program execution

24. A named constant is
    a. a variable that can change its value only during compilation
    b. a variable that is bound to a value only when it is bound to storage
    c. a variable that cannot change its location
    d. all of the above

25. Dynamic scoping means:
    a. The variables are allocated statically
    b. The variables are visible according to the location of calling a function
    c. The variables are visible according to the lexical structure of the program
    d. The variables are allocated dynamically

Name: _____ ID: _____

Question 2:

a. Write the formal definition in BNF for the variable(s) declaration that is defined informally as a list of variables preceded by the reserved word **var,** and followed by a colon (:), a type, and semicolon (;). The type is one of the reserved words **char,** or **bool.** The list of variables is defined as a set of variables separated by comma (,). The variable is an identifier. For example this a valid string driven from the above grammar: **var** a,b,c: **char;** [4 marks]

b. Convert the grammar written in (a) to EBNF notation [3 marks]

c. Draw the syntax diagram for the above grammar [3 marks]

Name: _____ ID: _____

Question 3:

Given the following grammar:

assign → *id* := *expr*
id → A | B | C
expr → *expr* + *term* | *term*
term → *term* * *factor* | *factor*
factor → (*expr*) | *id*

a. Using the above grammar, show a **leftmost derivation (first five steps)** for the following assignment statement: A:= ((A * B) + C) [3 marks]

b. Using the above grammar, show a **rightmost derivation (first five steps)** for the following assignment statement: A:=A+(B+C)+A [3 marks]

c. Draw the abstract syntax tree for each of the above statements [4 marks]

Name: _____ ID: _____

Question 4:

Given the following program (in some hypothetical programming language):

```
Program main;
    var x, z : integer;

    procedure sub1;
        var a, b, z: integer;
                                    ---------->

    begin {sub 1}
        writeln('The value of x is : ', x);
        …
    end {sub 1}

    procedure sub 2;
        var a, y, z: integer;

        begin {sub 2}
        sub 1;    …
        end {sub 2}

    procedure sub3;
        var a, x, y: integer;
                                    ---------->
    begin {sub 3}
         x := 5;
        sub 2;
        …
    end {sub 3}

begin {main}
    x:= 30;
    sub 3;    …
end {main}
```

a. Show the content of the symbol table using static scoping rules at the indicated locations [4 marks]

b. Assume the above program was compiled and executed using static scoping rules. What value of x is printed in procedure sub1? [3 marks]

c. What if dynamic scoping rules are used in the program above, what value of x is printed in procedure sub1 [3 marks]

Name: _____ ID: _____

Question 5:

Given the following skeletal program where BIGSUB is the main program

	Memory Stack at Position 1	Memory Stack at Position 1
Main BIGSUB; **var** a,b: **integer;** **Procedure** C; **var** x,y **:real;** **Procedure** D; **var** a,x,w**: char;** /* start of D */ …….. …… ← 1 **End;** {D} /* start of C */……. **call** D; **End;** {C} **Procedure** A(f: **boolean**); **var** x: **boolean**; **Procedure** B; **var** b**: real;**……. /* start of B */ A(false) ……← 2 **End**; {B} /*start of A */…….. **If** f **then** B **else** C **End**; {A} /* start of BIGSUB */ …… **call** A(true); **End**; {BIGSUB}		

Show the stack with all activation record instances, when execution reaches position 1and position 2 in the above skeletal program [10 marks]