Program: BE Computer Engineering

Curriculum Scheme: Revised 2016

Examination: Third Year Semester VI

Course Code: CSC602 and Course Name: SPCC

|  |  |
| --- | --- |
| Q1. | Which of the following grammar is unambiguous? |
| Option A: | S-> aS |Sa| Є |
| Option B: | E-> E +E | E\*E| id |
| Option C: | A -> AA | (A) | a |
| Option D: | S -> AA , A -> aA , A -> b |
|  |  |
| Q2. | What converts High Level Language to output of low  level language i.e. machine or assembly language. |
| Option A: | Compiler |
| Option B: | Assembler |
| Option C: | Interpreter |
| Option D: | Source Program |
|  |  |
| Q3. | What is “**;**” in Lexical Analysis? |
| Option A: | Keywords |
| Option B: | Identifiers |
| Option C: | Seperators |
| Option D: | Operators |
|  |  |
| Q4. | S->aBDh  B->cC  C->bC  D->EF  E->g| Є  F->f| Є  What is FIRST(C)? |
| Option A: | {g, Є } |
| Option B: | {b, Є) |
| Option C: | {f, Є } |
| Option D: | {b} |
|  |  |
| Q5. | S->aBDh  B->cC  C->bC  D->EF  E->g| Є  F->f| Є  What is FIRST(D)? |
| Option A: | {g, Є } |
| Option B: | {b, Є) |
| Option C: | {g,f, Є } |
| Option D: | {b,f,Є} |
|  |  |
| Q6. | S->aBDh  B->cC  C->bC  D->EF  E->g| Є  F->f| Є  What is FOLLOW(C)? |
| Option A: | ){g,f,h} |
| Option B: | {g,f} |
| Option C: | {f,h} |
| Option D: | {f,h,e} |
|  |  |
| Q7. | S->aBDh  B->cC  C->bC  D->EF  E->g| Є  F->f| Є  What is FOLLOW(E)? |
| Option A: | ){g,f,h} |
| Option B: | {g,f} |
| Option C: | {f,h} |
| Option D: | {f,h,g} |
|  |  |
| Q8. | Assembler is a program that |
| Option A: | places programs into memory and prepares then for execution |
| Option B: | automates the translation of assemble language into machine language |
| Option C: | accepts a program written in a high level language and produces an object program |
| Option D: | appears to execute a resource as if it were machine language |
|  |  |
| Q9. | which of the following are language processors |
| Option A: | assembler |
| Option B: | compilers |
| Option C: | interpreter |
| Option D: | Linker and Loader |
|  |  |
| Q10. | An assembly program contains |
| Option A: | Imperative statements and assembler directives |
| Option B: | imperative and declarative statement |
| Option C: | imperative and declarative statement as well as assembler directive |
| Option D: | declarative statements and assembler directive |
|  |  |
| Q11. | At what phase of compiler grammar of the programming  is checked? |
| Option A: | Code generation |
| Option B: | Syntax Analysis |
| Option C: | Code Optimization |
| Option D: | Semantic Analysis |
|  |  |
| Q12. | Which is not a phase of front end compiler? |
| Option A: | Generation of intermediate code |
| Option B: | Semantic Analysis |
| Option C: | Code Optimization |
| Option D: | creation of symbol table |
|  |  |
| Q13. | A sequence of characters in the source program that is  matched by the pattern of tokens is known as? |
| Option A: | Tokens |
| Option B: | Lexeme |
| Option C: | Pattern |
| Option D: | Syntax |
|  |  |
| Q14. | Compiler can check which type of errors? |
| Option A: | Logical |
| Option B: | Syntax |
| Option C: | context |
| Option D: | Content |
|  |  |
| Q15. | List of Tokens is the output of \_\_\_\_\_\_\_\_? |
| Option A: | Interpreter |
| Option B: | Compiler |
| Option C: | Syntax Analysis |
| Option D: | Lexical Analysis |
|  |  |
| Q16. | The linker ? |
| Option A: | is same as the loader |
| Option B: | is required to create a load module |
| Option C: | is always used before programs are executed |
| Option D: | executed program of loader |
|  |  |
| Q17. | A system program that combines the separately compiled modules of a program into a form suitable for execution  ? |
| Option A: | Assembler |
| Option B: | Linking loader |
| Option C: | Cross compiler |
| Option D: | Load and Go |
|  |  |
| Q18. | Loading process can be divided into two separate programs, to solve some problems. The first is binder the other is ? |
| Option A: | Linkage editor |
| Option B: | Relocator |
| Option C: | Module Loader |
| Option D: | Cross compiler |
|  |  |
| Q19. | Load address for the first word of the program is called |
| Option A: | Load address origin |
| Option B: | Linker address origin |
| Option C: | Phase library |
| Option D: | Absolute library |
|  |  |
| Q20. | A linker program |
| Option A: | places the program in the memory for the purpose of execution. |
| Option B: | relocates the program to execute from the specific memory area allocated to it. |
| Option C: | Links the program with other programs needed for its execution. |
| Option D: | interfaces the program with the entities generating its input data. |
|  |  |
| Q21. | A grammar that produces more than one parse tree for some sentence is called \_\_\_\_\_\_\_\_\_\_ |
| Option A: | Ambiguous |
| Option B: | Unambiguous |
| Option C: | Regular |
| Option D: | Shift reduce |
|  |  |
| Q22. | Shift reduce parsers are \_\_\_\_\_\_\_\_. |
| Option A: | Top down Parser |
| Option B: | Bottom Up parser |
| Option C: | May be top down or bottom up |
| Option D: | Regular |
|  |  |
| Q23. | What is the output of lexical analyzer? |
| Option A: | A parse tree |
| Option B: | Intermediate code |
| Option C: | A list of tokens |
| Option D: | Machine code |
|  |  |
| Q24. | \_\_\_\_\_\_\_\_\_\_\_is a graph representation of a derivation. |
| Option A: | The parse tree |
| Option B: | The oct tree |
| Option C: | The binary tree |
| Option D: | Syntax tree |
|  |  |
| Q25. | In compiler, Source program is read by\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. |
| Option A: | parser |
| Option B: | lexical analyzer |
| Option C: | developer |
| Option D: | Analyst |