

# Dronacharya College of Engineering

Department of Electronics and Computers Engineering

## MULTIPLE CHOICE QUESTIONS

(Session 2014-2015)

**Subject with code: Compiler Design(EC-712 F)**

**Sem: VII /Branch: ECS**

**Q.1** A compiler for a high level language that runs on one machine and produce code for different machine is called

- Optimizing compiler
- One pass compiler
- Cross compiler
- Multipass compiler

**Q.2** The output of a lexical analyzer is

- Machine code
- Intermediate code
- A stream of tokens
- A parse tree

**Q.3** \_\_\_\_\_or scanning is the process where the stream of characters making up the source program is read from left to right and grouped into tokens.

- Lexical analysis
- Diversion
- Modeling
- None of the above

**Q.4** Task of the lexical analysis

- To parse the source program into the basic elements or tokens of the language
- To build a literal table and an identifier table
- To build a uniform symbol table
- All of these

**Q.5** The linker

- is similar to interpreter
- uses source code as its input
- is required to create a load module
- none of the above

**Q.6** Type checking is normally done during

- Lexical analysis
- Syntax analysis
- Syntax directed translation
- Code optimization

**Q.7** The lexical analyzer takes \_\_\_\_\_ as input and produces a stream of \_\_\_\_\_ as output.

- Source program,tokens
- Token,source program
- Either A and B
- None of the above

**Q.8** Macro-processors are \_\_\_\_\_

- Hardware
- Compiler
- Registers
- None of the above

**Q.9** In which way(s) a macroprocessor for assembly language can be implemented ?

- Independent two-pass processor
- Independent one-pass processor
- Expand macrocalls and substitute arguments
- All of the above

**Q.10** Macro' in an assembly level program is \_\_\_\_\_.

- sub program
- a complete program
- a hardware portion
- relative coding

**Q.11** Grammar of the programming is checked at \_\_\_\_\_ phase of compiler.

- semantic analysis
- code generation
- syntax analysis
- code optimization

## **Section :-B**

**Q.1** The action of parsing the source program into proper syntactic classes is called

- Syntax analysis
- Lexical analysis
- Interpretation analysis
- General syntax analysis

**Q.2 Which of the following is used for grouping of characters into tokens?**

- Parser
- Code optimization
- Code generator
- Lexical analyzer

**Q.3 A grammar is meaningless**

- If terminal set and non terminal set are not disjoint
- If left hand side of a production is a single terminal
- If left hand side of a production has no non terminal
- All of these

**Q.4 Which of the following is not an intermediate code form?**

- Postfix notation
- Syntax trees
- Three address codes
- Quadruples

**Q.5 LR stands for**

- Left to right
- Left to right reduction
- Right to left
- Left to right and right most derivation in reverse

**Q.6 Inherited attribute is a natural choice in**

- Keeping track of variable declaration
- Checking for the correct use of L values and R values
- Both A and B
- None of these

**Q.7 Which of the following parser is most powerful?**

- Operator precedence
- Canonical LR
- LALR
- SLR

**Q.8 Syntax directed translation scheme is desirable because**

- It is based on the syntax
- Its description is independent of any implementation
- It is easy to modify
- All of these

**Q.9 A top down parser generates**

- Right most derivation
- Right most derivation in reverse
- Left most derivation
- Left most derivation in reverse

**Q.10 Type checking is normally done during**

- Lexical analysis
- Syntax analysis
- Syntax directed translation
- Code optimization

**Q.11 Whether a given pattern constitutes a token or not depends on the**

- Source language
- Target language
- Compiler
- All of these

**Q.12 A grammar that produces more than one parse tree for some sentence is called**

- Ambiguous
- Unambiguous
- Regular
- None of these

**Q.13 Shift reduce parsers are**

- Top down parser
- Bottom up parser
- May be top down or bottom up parser
- None of the above

**Q.14 Synthesized attribute can be easily simulated by a**

- LL grammar
- Ambiguous grammar
- LR grammar
- None of the above

## **Section :-C**

**Q.1 Three address code involves**

- Exactly 3 address
- At most most 3 address
- No unary operators
- None of these

**Q.2 Relocating bits used by relocating loader are specified by**

- Relocating loader itself
- Linker
- Assembler
- Macro processor

**Q.3 Running time of a program depends on**

- The way the registers and addressing modes are used
- The order in which computations are performed
- The usage of machine idioms
- All of these

**Q.4 Any description error can be repaired by**

- Insertion alone
- Deletion alone
- Insertion and deletion alone
- Replacement alone

**Q.5 A grammar that produces more than one parse tree for some sentence is called**

- Ambiguous
- Unambiguous
- Regular
- None of these

**Q.6 Which of the following does not interrupt a running process?**

- A device
- Timer
- Scheduler
- Power failure

**Q.7 A parser with the valid prefix property is advantageous because it**

- Detects error as soon as possible
- Detects errors as and when they occur
- Limits the amount of erroneous output passed to the text phase
- All of these

**Q.8 \_\_\_\_\_ is a graph representation of a derivation.**

- The parse tree
- The oct tree

- The binary tree
- None of the above

**Q.9 Which of the following can be accessed by transfer vector approach of linking?**

- External data segments
- External subroutines
- Data located in other procedure
- All of these

**Q.10 YACC builds up**

- SLR parsing table
- Canonical LR parsing table
- LALR parsing table
- None of the above

**Q.11 Intermediate code generation phase gets input from**

- Lexical analyzer
- Syntax analyzer
- Semantic analyzer
- Error handling

**Q.12 In operator precedence parsing , precedence relations are defoned**

- For all pair of non terminals
- For all pair of terminals
- To delimit the handle
- Only for a certain pair of terminals



## **Section :-D**

### **Q.1 Pee hole optimization**

- Loop optimization
- Local optimization
- Constant folding
- Data flow analysis

### **Q.2 An optimizer compiler**

- Is optimized to occupy less space
- Is optimized to take less time for execution
- Optimizes the code
- None of these

### **Q.3 Local and loop optimization in turn provide motivation for**

- Data flow analysis
- Constant folding
- Pee hole optimization
- DFA and constant folding

### **Q.4 Concept which can be used to identify loops is**

- Dominators
- Reducible graphs
- Depth first ordering
- All of these

### **Q.5 The optimization technique which is typically applied on loops is**

- Removal of invariant computation

- Peephole optimization
- Constant folding
- All of these

**Q.6 The optimization which avoids test at every iteration is**

- Loop unrolling
- Loop jamming
- Constant folding
- None of these

**Q.7 Code can be optimized at**

- Source from user
- Target code
- Intermediate code
- All of the above

**Q.8 We can optimize code by**

- Dead code elimination
- Common subprograms
- Copy intermediate loop
- Loop declaration

**Q.9 In an absolute loading scheme which loader function is accomplished by assembler ?**

- re-allocation
- allocation
- Linking
- Loading

**Q.10 Reduction in strength means**

- Replacing run time computation by compile time computation
- Removing loop invariant computation
- Removing common sub expression
- Replacing a costly operation by a relatively cheaper one