

# Compiler Design ISE-I

tejassaindane99@gmail.com [Switch accounts](#)

Draft saved

## Question

\_\_\_\_\_ is a program that takes as input a program written in one language (source language) and produces as output a program in another language (object language). 1 point

- A. translator
- B. assembler
- C. compiler
- D. interpreter

[Clear selection](#)

If the source language is high-level language and the object language is a low-level language(assembly or machine), then such a translator is called as a \_\_\_\_\_. 1 point

- A. translator
- B. assembler
- C. compiler
- D. interpreter

[Clear selection](#)

An interpreter is a program that directly executes an \_\_\_\_\_ code 1 point

- A. source
- B. object
- C. intermediate
- D. subject

[Clear selection](#)

The compilation process is partitioned into a series of sub processes called \_\_\_\_\_ 1 point

- A. phases
- B. sub program
- C. module
- D. subsets

[Clear selection](#)

The first phase of the compiler is also called as \_\_\_\_\_. 1 point

- A. scanner
- B. parser
- C. tokens
- D. macro

[Clear selection](#)

\_\_\_\_\_ breaks source program into pieces and creates an intermediate representation. 1 point

- A. Analysis
- B. Synthesis
- C. Code generator
- D. Optimizer

[Clear selection](#)

Users write the programs in which language?

1 point

- Low-level Language
- High-Level Language
- Decimal-Format
- Middle-Level Language

[Clear selection](#)

Which computer program accepts the high-level language and converts it into assembly language?

1 point

- Interpreter
- Linker
- Assembler
- Compiler

[Clear selection](#)

Does the compiler program translate the whole source code in one step?

1 point

- No
- Depends on the Compiler
- Don't Know
- Yes

[Clear selection](#)

Which of the following file is an output of the assembler?

1 point

- Program file
- Object file
- Data File
- Task File

[Clear selection](#)

Which tool is used for grouping of characters in tokens in the compiler?

1 point

- Parser
- Code optimizer
- code generator
- Scanner

[Clear selection](#)

Parsing is categorized into how many types?

1 point

- three types
- four types
- two types
- five types

[Clear selection](#)

In which parsing, the parser constructs the parse tree from the start symbol and transforms it into the input symbol?

1 point

- Bottom-up parsing
- Top-down parsing

^

None of the above

Both a and b

[Clear selection](#)

Which derivation is generated by the top-down parser?

1 point

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

[Clear selection](#)

Which derivation is generated by the bottom-up parser?

1 point

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

[Clear selection](#)

Which phase of the compiler checks the grammar of the programming?

1 point

- Code Optimization
- Semantic Analysis
- Code Generation
- Syntax Analys

[Clear selection](#)

Which of the following component is important for semantic analysis?

1 point

- Yacc
- Lex
- Symbol Table
- Type Checking

[Clear selection](#)

Which phase of the compiler is also known as Scanner?

1 point

- Syntax Analysis
- Lexical Analysis
- Semantic Analysis
- Code generation

[Clear selection](#)

Which phase of the compiler is also known as Parser?

1 point

- Code Optimization
- Semantic Analysis
- Syntax Analysis
- Lexical Analysis

[Clear selection](#)

Which of the following parser is a top-down parser?

1 point

- An LALR parser
- A LR parser

operator precedence parser

Recursive descent parser

[Clear selection](#)

Keywords are recognized in a compiler during -

1 point

the code generation

the data flow analysis

the lexical analysis of the program

the program parsing

[Clear selection](#)

Leaf nodes in a parse tree indicate?

1 point

sub-terminals

half-terminals

non-terminals

terminals

[Clear selection](#)

Which graph describes the basic block and successor relationship?

1 point

control graph

DAG

Flow graph

Hamilton graph

[Clear selection](#)

Option 1

[Clear selection](#)

Why compiler needs multiple passes?

1 point

A. Some issues raised early in a program may remain un-answered until later.

B. All issues are defined before-hand.

C. Have enough memory to do everything.

D. All of the above

[Clear selection](#)

Compilers can be ported to different language and machines

1 point

A. by reusing the front end

B. by re-doing the back end

C. by reusing the front end and re-doing the back end

D. None

[Clear selection](#)

Which of the following is true?

1 point

A. The analysis part is called the front end of the compiler and the synthesis part is the back end.

B. The analysis part is called the back end of the compiler and the synthesis part is the front end.

C. The analysis part and synthesis part are called the front end of the compiler.

D. The analysis part and synthesis part are called the back end of the compiler.

D. The analysis part and synthesis part are called the back end of the compiler.

[Clear selection](#)

Which of the following is false?

1 point

- A. Compiler is usually much faster than an interpreter.
- B. An interpreter is better error diagnostics than a compiler.
- C. It is easier to produce machine language as an output by a compiler.
- D. It is easier to produce Assembly language as an output by a compiler.

[Clear selection](#)

The output of the lexical analyzer is a stream of \_\_\_\_\_.

1 point

- A. instructions
- B. tokens
- C. values
- D. inputs

[Clear selection](#)

The main function of lexical analyzer is to read a \_\_\_\_\_.

1 point

- A. source program
- B. object program
- C. intermediate code
- D. sub program

[Clear selection](#)

Which is not a token?

1 point

- A. operator
- B. instructions
- C. keywords
- D. identifier

[Clear selection](#)

\_\_\_\_\_ is used to define a token.

1 point

- A. Lexical Analyzer
- B. Parser
- C. Regular Expression
- D. Identifier

[Clear selection](#)

\_\_\_\_\_ is a tool that automatically generating lexical analyzer.

1 point

- A. LEX
- B. HEX
- C. SLR
- D. CLR

[Clear selection](#)

Tokens are \_\_\_\_\_.

1 point

- A. terminals
- B. non-terminals
- C. symbols

D. digits

[Clear selection](#)

Which of the following is a task of Lexical Analyzer?

1 point

- A. stripping out comments and whitespace
- B. check Syntactic construct
- C. Type checking
- D. Debugging

[Clear selection](#)

In lexical analysis, which of the following is not a panic mode error recovery action from the remaining input?

1 point

- A. delete one character
- B. Insert a missing character
- C. Replace a character
- D. Interchange any characters

[Clear selection](#)

Sentinel is a special character that is \_\_\_\_\_.

1 point

- A. Symbol '\$'
- B. part of the source program
- C. not part of the source program
- D. None of the above

[Clear selection](#)

A LEX specification consists of \_\_\_\_\_ parts.

1 point

- A. Two
- B. Three
- C. Four
- D. Five

[Clear selection](#)

A parser for Grammar G is a program that takes as input string W and produces as output is \_\_\_\_\_ for W

1 point

- A. parse tree
- B. syntax tree
- C. error message
- D. string

[Clear selection](#)

In a top-down parser, the starting \_\_\_\_\_ is expanded to derive the given input string.

1 point

- A. terminal
- B. letter
- C. digit
- D. non-terminal

[Clear selection](#)

Predictive parser is one kind of \_\_\_\_\_ parser.

1 point

A. shift-reduce

B. recursive descent

C. bottom-up

D. top-down

[Clear selection](#)

\_\_\_\_\_ makes grammar suitable for parsing. 1 point

A. Factoring

B. Right Factoring

C. Left Factoring

D. Reverse Factoring

[Clear selection](#)

Viable-prefix property allows early detection of \_\_\_\_\_. 1 point

A. Lexical error

B. syntactic error

C. semantic error

D. logical error

[Clear selection](#)

Panic-Mode Error Recovery is skipping the input symbols until a \_\_\_\_\_ 1 point  
is found.

A. end of File

B. end of statement

C. designated token

D. synchronizing token

[Clear selection](#)

Which of the following statements is false? 1 point

a) Left as well as right most derivations can be in Unambiguous grammar

b) An LL (1) parser is a top-down parser

c) LALR is more powerful than SLR

d) Ambiguous grammar can't be LR (k)

[Clear selection](#)

Which of the following suffices to convert an arbitrary CFG to an LL(1) 1 point  
grammar?

a) Removing left recursion only

b) Factoring the grammar alone

c) Factoring & left recursion removal

d) None of the mentioned

[Clear selection](#)

Assume that the SLR parser for a grammar G has n1 states and the LALR 1 point  
parser for G has n2 states.

a) n1 is necessarily less than n2

b) n1 is necessarily equal to n2

c) n1 is necessarily greater than n2

d) none of the mentioned

[Clear selection](#)

Which of the following derivations does a top-down parser use while parsing an input string? 1 point

- a) Leftmost derivation
- b) Leftmost derivation in reverse
- c) Rightmost derivation
- d) Rightmost derivation in reverse

[Clear selection](#)

The process of assigning load addresses to the various parts of the program and adjusting the code and data in the program to reflect the assigned addresses is called? 1 point

- a) Assembly
- b) Parsing
- c) Relocation
- d) Symbol resolute

[Clear selection](#)

Which of the following statements is false? 1 point

- a) Left as well as right most derivations can be in Unambiguous grammar
- b) An LL (1) parser is a top-down parser
- c) LALR is more powerful than SLR
- d) Ambiguous grammar can't be LR (k)

[Clear selection](#)

Which one of the following is a top-down parser? 1 point

- a) Recursive descent parser
- b) Operator precedence parser
- c) An LR(k) parser
- d) An LALR(k) parser

[Clear selection](#)

Which of the following is TRUE? 1 point

- a) Both P and Q are true
- b) P is true and Q is false
- c) P is false and Q is true
- d) Both P and Q are false

[Clear selection](#)

Which one of the following is TRUE? 1 point

- a) + is left associative, while \* is right associative
- b) + is right associative, while \* is left associative
- c) Both + and \* are right associative
- d) Both + and \* are left associative

[Clear selection](#)

The grammar  $A \rightarrow AA \mid (A) \mid e$  is not suitable for predictive-parsing because the grammar is? 1 point

- a) Ambiguous
- b) Left recursive
- c) Right recursive

d) An operator grammar

[Clear selection](#)

Which of the following suffices to convert an arbitrary CFG to an LL(1) grammar? 1 point

- a) Removing left Recursive alone
- b) Factoring the grammar alone
- c) Along with removing left recursion we also perform the factoring of the grammar
- d) None of the mentioned

[Clear selection](#)

How many parts of compiler are there? 1 point

- A. 1
- B. 2
- C. 4
- D. 8

[Clear selection](#)

Grammar of the programming is checked at \_\_\_\_\_ phase of compiler. 1 point

- A. Semantic analysis
- B. Syntax analysis
- C. Code optimization
- D. Code generation

[Clear selection](#)

\_\_\_\_\_ is a process of finding a parse tree for a string of tokens. 1 point

- A. Parsing
- B. Analysing
- C. Recognizing
- D. Tokenizing

[Clear selection](#)

What is the action of parsing the source program into proper syntactic classes? 1 point

- A. Lexical analysis
- B. Syntax analysis
- C. General syntax analysis
- D. Interpretation analysis

[Clear selection](#)

Compiler can check \_\_\_\_\_ error. 1 point

- A. Logical
- B. Syntax
- C. Content
- D. Both A and B

[Clear selection](#)

A grammar that produces more than one parse tree for some sentence is called as 1 point

A. Ambiguous

B. Unambiguous

C. Regular

D. All of these

[Clear selection](#)

[Back](#)

[Submit](#)

Page 2 of 2

[Clear form](#)

Never submit passwords through Google Forms.

This form was created inside College of Engineering and Technology. [Report Abuse](#)

Google Forms