REGULATIONS 2021 ACADEMIC YEAR 2023 - 2024

UNIT – I CLOUD ARCHITECTURE MODELS AND INFRASTRUCTURE

Cloud Architecture: System Models for Distributed and Cloud Computing – NIST Cloud Computing Reference

Architecture – Cloud deployment models – Cloud service models; Cloud Infrastructure: Architectural Design

of Compute and Storage Clouds – Design Challenges

PART – A

Q.N

o.

QUESTIONS

1.

What is Cloud Computing? BTL 1

Cloud is a parallel and distributed computing system consisting of a collection of inter-

connected and virtualized computers that are dynamically provisioned and presented as

one or more unified computing resources based on service-level agreements (SLA)

established through negotiation between the service provider and consumers.

2.

What are the computing

**CS3501-COMPILER DESIGN**

**UNIT 1 INTRODUCTION TO COMPILERS& LEXICAL ANALYSIS**

**PART-A**

**1. What is a Complier?**

A Complier is a program that reads a program written in one language-the source language-and translates it in to an equivalent program in another language-the target language .The compiler reports to its user the presence of errors in the source program.

**2. What are the main two parts of compilation? What are they performing?**

The two main parts are

• Analysis part breaks up the source program into constituent pieces and creates an intermediate representation of the source program.

• Synthesis part constructs the desired target program from the intermediate representation.

**3. How many phases does analysis consists?**

Analysis consists of three phases

i .Linear analysis

ii. Hierarchical analysis

iii. Semantic analysis

**4. What is a Symbol table?**

A Symbol table is a data structure containing a record for each identifier, with fields for the attributes of the identifier. The data structure allows us to find the record for each identifier quickly and to store or retrieve data from that record quickly.

**5. State the general phases of a compiler**

i. Lexical analysis

ii. Syntax analysis

iii. Semantic analysis

iv. Intermediate code generation

v. Code optimization

vi. Code generation

**6. What is the need for separating the analysis phase into lexical analysis and parsing? (Or)**

**What are the issues of lexical analyzer?**

• Simpler design is perhaps the most important consideration. The separation of lexical analysis

from syntax analysis often allows us to simplify one or the other of these phases.

• Compiler efficiency is improved.

• Compiler portability is enhanced.

**7. What is Lexical Analysis?**

The first phase of compiler is Lexical Analysis. This is also known as linear analysis in which thestream of characters making up the source program is read from left-toright and grouped into tokens that are sequences of characters having a collective meaning.

**8. What are the functions performed in analysis phase?**

* Lexical analysis or Linear analysis
* Syntax analysis or hierarchical analysis
* Semantic analysis

**9. What are the functions performed in synthesis phase?**

* Intermediate code generation
* Code generation
* Code optimization

**10. Define patterns/lexeme/tokens**

A set of strings in the input for which the same token is produced as output. This set of stringsdescribed by a rule called pattern associated with the token.

A lexeme is a sequence of characters in the source program that is matched by the pattern for a

token.

Token is a sequence of character that can be treated as a single logical entity.

**UNIT II SYNTAX ANALYSIS**

Part-A

**1. What is the output of syntax analysis phase? What are the three general types of parsers for grammars?**

Parser (or) parse tree is the output of syntax analysis phase.

General types of parsers:

1) Universal parsing

2) Top-down

3) Bottom-up

**2. What are the different strategies that a parser can employ to recover from a syntactic error?**

* Panic modePhrase level
* Error productions
* Global correction

**3. What are the goals of error handler in a parser?**

The error handler in a parser has simple-to-state goals:

* It should report the presence of errors clearly and accurately.
* It should recover from each error quickly enough to be able to detect
* subsequent errors.
* It should not significantly slow down the processing of correct programs.

**4. What is phrase level error recovery?**

On discovering an error, a parser may perform local correction on the remaining input;that is, it may replace a prefix of the remaining input by some string that allows theparser to continue. This is known as phrase level error recovery.

**5. How will you define a context free grammar?**

A context free grammar consists of terminals, non-terminals, a start symbol, and productions.

i. Terminals are the basic symbols from which strings are formed. “Token” isa

synonym for terminal. Ex: if, then, else.

ii. Nonterminals are syntactic variables that denote sets of strings, which help define

the language generated by the grammar. Ex: stmt, expr.

iii. Start symbol is one of the nonterminals in a grammar and the set of strings it

denotes is the language defined by the grammar. Ex: S.

iv. The productions of a grammar specify the manner in which the terminals and nonterminals can be combined to form strings Ex: expr id

**6. Define context free language. When will you say that two CFGs are equal?**

A language that can be generated by a grammar is said to be a context free language. Iftwo

grammars generate the same language, the grammars are said to be equivalent.

**7. Give the definition for leftmost and canonical derivations.**

Derivations in which only the leftmost nonterminal in any sentential form is replaced at each step are termed leftmost derivations

Derivations in which the rightmost nonterminal is replaced at each step are termed canonical derivations.

**8. What is a parse tree?**

A parse tree may be viewed as a graphical representation for a derivation that filters out the choice regarding replacement order. Each interior node of a parse tree is labeled by some nonterminal A and that the children of the node are labeled from left to right by symbols in the right side of the production by which this A was replaced in the derivation. The leaves of the parse tree are terminal symbols.

**9. What is an ambiguous grammar? Give an example.**

A grammar that produces more than one parse tree for some sentence is said to be ambiguous

An ambiguous grammar is one that produces more than one leftmost or rightmost derivation for the same sentence.

Ex:

E E+E / E\*E / id

**10. Why do we use regular expressions to define the lexical syntax of a language?**

i. The lexical rules of a language are frequently quite simple, and to describe them wedo

not need a notation as powerful as grammars.

ii. Regular expressions generally provide a more concise and easier to understand

notation for tokens than grammars.

iii. More efficient lexical analyzers can be constructed automatically from regular

expressions than from arbitrary grammars.

iv. Separating the syntactic structure of a language into lexical and non lexical parts provides a convenient way of modularizing the front end of a compiler into two manageable-sized components.

**UNIT III SYNTAX DIRECTED TRANSLATION &INTERMEDIATE CODE GENERATION**

PART-A

**1.What is syntax directed translation?**

A syntax directed definition specifies the values of attributes by associating semantic rules with the grammar productions

Production E->E1+T

Semantic Rule E.code=E1.code||T.code||’+’

**2. What is synthesized attributes?**

A synthesized attribute at node N is defined only in terms of attribute values of children of N and at N

**3. What is inherited attributes ?**

An inherited attribute at node N is defined only in terms of attribute values at N’s parent, N itself and N’s siblings

**4. List the three kinds of intermediate representation.**

The three kinds of intermediate representations are

i. Syntax trees

ii. Postfix notation

iii. Three address code

**5. What is postfix notation?**

A Postfix notation is a linearized representation of a syntax tree. It is a list of nodes of the tree in which a node appears immediately after its children.

**6. What is the usage of syntax directed definition.**

Syntax trees for assignment statement are produced by the syntax

directeddefinition.

**7. Why “Three address code” is named so?**

The reason for the term “Three address code” is that each usually

containsthree addresses, two for operands and one for the result.

**8. Define three-address code.**

Three-address code is a sequence of statements of the general

formx := y op z

where x, y and z are names, constants, or compiler-generated temporaries; op stands

for any operator, such as fixed or floating-point arithmetic operator, or a logical operator on boolean-valued data. Three-address code is a linearized representation of a syntax tree or a dag in which explicit names correspond to the interior nodes of the graph.

**9. State quadruple**

A quadruple is a record structure with four fields, which we call op, arg1,

arg2and result.

**10. What is called an abstract or syntax tree?**

A tree in which each leaf represents an operand and each interior node an operator is called as abstract or syntax tree.

**UNIT IV RUN-TIME ENVIRONMENT AND CODE GENERATION**

PART-A

**1. Define an attribute. Give the types of an attribute?**

An attribute may represent any quantity, with each grammar symbol, it associates a set of attributes and with each production, a set of semantic rules for computing values of the attributes associated with the symbols appearing in that production. Example: a type, a value, a memory location etc.,

1. Synthesized attributes.
2. ii) Inherited attributes.

**2. Mention the two rules for type checking.**

Type checker for a language is based on information about the syntactic constructs in the language, the notion of types, and the rules for assigning types to language constructs.

**3. When does dangling references occur?**

When there is a reference to storage that has been de-allocated, logical error occurs as it uses dangling reference where the value of de-allocated storage is undefined according to the semantics of most languages

**4. What is syntax directed translation?**

A syntax directed definition specifies the values of attributes by associating semantic rules with the grammar productions

Production E->E1+T

Semantic Rule E.code=E1.code||T.code||’+’

**5. What is synthesized attributes?**

A synthesized attribute at node N is defined only in terms of attribute values of children of N and at N

**6. What is inherited attributes ?**

An inherited attribute at node N is defined only in terms of attribute values at N’s parent, N itself and N’s siblings

**7. Define DAG.**

A DAG for a basic block is a directed acyclic graph with the following labels on nodes:

i) Leaves are labeled by unique identifiers, either variable names or constants.

ii) Interior nodes are labeled by an operator symbol.

iii)Nodes are also optionally given a sequence of identifiers for labels.

**8. What are the functions and properties of Memory Manager?**

Two basic functions: Allocation Deallocation

Properties of memory managers: Space efficiency

Program efficiency

Low overhead

**9. What is static checking?**

A compiler must check that the source program follows both syntactic and semantic conversions of the source language. This checking called static checking detects and reports programming errors.

**10. Give some examples of static checking?**

**Type checks:**

A compiler should report an error if an operator is applied to an incompatible operand.

**Flow of control checks:**

Statements that cause flow of control to leave a construct must have some place to which to transfer the flow of control.

**UNIT V CODE OPTIMIZATION**

PART-A

**1. What are basic blocks?**

A sequence of consecutive statements which may be entered only at the beginning and when

**2. What do you mean by copy propagation?**

After the assignment of one variable to another, a reference to one variable may be replaced with the value of the other variable.

If w := x appears in a block, all subsequent uses of w can be replaced with uses of x.

Before After

b := z + y b := z + y

a := b a := b

x := 2 \* b x := 2 \* a

**3. What is a flow graph?**

The basic block and their successor relationships shown by a directed graph is called a flow graph. The nodes of a flow graph are the basic blocks.

**4. Write the three address code sequence for the assignment statement.**

d:=(a-b)+(a-c)+(a-c) t1=a-b

t2=a-c t3=t1+t2 t4=t3+t2

d=t4

**5. What is meant by peephole optimization?**

Peephole optimization is a technique used in many compliers, in connection with the optimization of either intermediate or object code. It is really an attempt to overcome the difficulties encountered in syntax directed generation of code.

**6. What are the issues in the design of code generators?**

Input to the code generator Target programs Memory management Instruction selection Register allocation Choice of evaluation order Approaches to code generation

**7. What is register descriptor and address descriptor?**

A register descriptor keeps track of what is currently in each register.

An address descriptor keeps track of the location where the current value of the name can be found at run time.

**8. Define DAG.**

A DAG for a basic block is a directed acyclic graph with the following labels on nodes:

Leaves are labeled by unique identifiers, either variable names or constants.

Interior nodes are labeled by an operator symbol.

Nodes are also optionally given a sequence of identifiers for labels.

**9. Name the techniques in Loop optimization.**

Code Motion, Induction variable elimination, Reduction in strength

**10. Define local optimization.**

The optimization performed within a block of code is called a local optimization.

CCS335 CLOUD COMPUTING

Question Bank

UNIT 1

CLOUD ARCHITECTURE MODELS AND INFRASTRUCTURE

SYLLABUS:Cloud Architecture: System Models for Distributed and Cloud Computing – NIST

Cloud Computing Reference Architecture – Cloud deployment models – Cloud service models;

Cloud Infrastructure: Architectural Design of Compute and Storage Clouds – Design Challenges

PART A

2 Marks

1. What is Cloud Computing? BTL 1

Cloud Computing is defined as storing and accessing of data and computing services over the

internet. It doesn’t store any data on your personal computer. It is the on-demand availability of

computer services like servers, data

CCS335 CLOUD COMPUTING

Question Bank

UNIT 1

CLOUD ARCHITECTURE MODELS AND INFRASTRUCTURE

SYLLABUS:Cloud Architecture: System Models for Distributed and Cloud Computing – NIST

Cloud Computing Reference Architecture – Cloud deployment models – Cloud service models;

Cloud Infrastructure: Architectural Design of Compute and Storage Clouds – Design Challenges

PART A

2 Marks

1. What is Cloud Computing? BTL 1

Cloud Computing is defined as storing and accessing of data and computing services over the

internet. It doesn’t store any data on your personal computer. It is the on-demand availability of

computer services like servers, data

CCS335 CLOUD COMPUTING

Question Bank

UNIT 1

CLOUD ARCHITECTURE MODELS AND INFRASTRUCTURE

SYLLABUS:Cloud Architecture: System Models for Distributed and Cloud Computing – NIST

Cloud Computing Reference Architecture – Cloud deployment models – Cloud service models;

Cloud Infrastructure: Architectural Design of Compute and Storage Clouds – Design Challenges

PART A

2 Marks

1. What is Cloud Computing? BTL 1

Cloud Computing is defined as storing and accessing of data and computing services over the

internet. It doesn’t store any data on your personal computer. It is the on-demand availability of

computer services like servers, data

CCS335 CLOUD COMPUTING

Question Bank

UNIT 1

CLOUD ARCHITECTURE MODELS AND INFRASTRUCTURE

SYLLABUS:Cloud Architecture: System Models for Distributed and Cloud Computing – NIST

Cloud Computing Reference Architecture – Cloud deployment models – Cloud service models;

Cloud Infrastructure: Architectural Design of Compute and Storage Clouds – Design Challenges

PART A

2 Marks

1. What is Cloud Computing? BTL 1

Cloud Computing is defined as storing and accessing of data and computing services over the

internet. It doesn’t store any data on your personal computer. It is the on-demand availability of

computer services like servers, data

4. Lower Software Costs

5. Instant Software Updates

6. Increased Computing Power

6. Write down disadvantage of cloud computing?BTL1

1. Requires a constant Internet connection

CCS335 CLOUD COMPUTING

Question Bank

UNIT 1

CLOUD ARCHITECTURE MODELS AND INFRASTRUCTURE

SYLLABUS:Cloud Architecture: System Models for Distributed and Cloud Computing – NIST

Cloud Computing Reference Architecture – Cloud deployment models – Cloud service models;

Cloud Infrastructure: Architectural Design of Compute and Storage Clouds – Design Challenges

PART A

2 Marks

1. What is Cloud Computing? BTL 1

Cloud Computing is defined as storing and accessing of data and computing services over the

internet. It doesn’t store any data on your personal computer. It is the on-demand availability of

computer services like servers, data

CCS335 CLOUD COMPUTING

Question Bank

UNIT 1

CLOUD ARCHITECTURE MODELS AND INFRASTRUCTURE

SYLLABUS:Cloud Architecture: System Models for Distributed and Cloud Computing – NIST

Cloud Computing Reference Architecture – Cloud deployment models – Cloud service models;

Cloud Infrastructure: Architectural Design of Compute and Storage Clouds – Design Challenges

PART A

2 Marks

1. What is Cloud Computing? BTL 1

Cloud Computing is defined as storing and accessing of data and computing services over the

internet. It doesn’t store any data on your personal computer. It is the on-demand availability of

computer services like servers, data

CCS335 CLOUD COMPUTING

Question Bank

UNIT 1

CLOUD ARCHITECTURE MODELS AND INFRASTRUCTURE

SYLLABUS:Cloud Architecture: System Models for Distributed and Cloud Computing – NIST

Cloud Computing Reference Architecture – Cloud deployment models – Cloud service models;

Cloud Infrastructure: Architectural Design of Compute and Storage Clouds – Design Challenges

PART A

2 Marks

1. What is Cloud Computing? BTL 1

Cloud Computing is defined as storing and accessing of data and computing services over the

internet. It doesn’t store any data on your personal computer. It is the on-demand availability of

computer services like servers, data

CCS335 CLOUD COMPUTING

Question Bank

UNIT 1

CLOUD ARCHITECTURE MODELS AND INFRASTRUCTURE

SYLLABUS:Cloud Architecture: System Models for Distributed and Cloud Computing – NIST

Cloud Computing Reference Architecture – Cloud deployment models – Cloud service models;

Cloud Infrastructure: Architectural Design of Compute and Storage Clouds – Design Challenges

PART A

2 Marks

1. What is Cloud Computing? BTL 1

Cloud Computing is defined as storing and accessing of data and computing services over the

internet. It doesn’t store any data on your personal computer. It is the on-demand availability of

computer services like servers, data

CCS335 CLOUD COMPUTING

Question Bank

UNIT 1

CLOUD ARCHITECTURE MODELS AND INFRASTRUCTURE

SYLLABUS:Cloud Architecture: System Models for Distributed and Cloud Computing – NIST

Cloud Computing Reference Architecture – Cloud deployment models – Cloud service models;

Cloud Infrastructure: Architectural Design of Compute and Storage Clouds – Design Challenges

PART A

2 Marks

1. What is Cloud Computing? BTL 1

Cloud Computing is defined as storing and accessing of data and computing services over the

internet. It doesn’t store any data on your personal computer. It is the on-demand availability of

computer services like servers, data