SET – 1

1. **Define Compiler.**

A. A compiler is a program that translates code written in a high-level programming language into machine code or an intermediate code, so it can be executed by a computer.

2. **Define Source Program.**

A. A source program is the original code written by a programmer in a high-level programming language before it is translated by a compiler into machine code.

3. **Define Lexical Analyzer.**

A. A lexical analyzer is the first compiler phase that converts source code into a sequence of tokens like keywords and identifiers.

4. **Define Syntax Tree.**

A. A syntax tree is a hierarchical structure representing the syntactic structure of a program based on its grammar.

5. **Define Semantic Analysis.**

A. Semantic analysis is a compiler phase that checks for semantic errors and ensures correct use of types, variables, and functions.

6. **Define Syntax Directed Definition.**

A. A Syntax Directed Definition (SDD) is a context-free grammar with attributes and rules used to define the semantics of a language.

7. **Define Dependency Graph.**

A. A dependency graph is a directed graph showing how attributes depend on each other in a syntax-directed definition.

8. **Define Intermediate Code.**

A. Intermediate code is a machine-independent code generated by the compiler between source code and target code during translation.

9. **Define Blank Symbol in TM.**

A. A blank symbol in a Turing Machine (TM) represents an empty cell on the tape. It is usually denoted by '□' or 'B' and indicates the absence of any input symbol.

10. **Define Decidability.**

A. Decidability refers to the property of a problem for which an algorithm exists that can provide a yes or no answer for all inputs in a finite amount of time.

SET – 2

1. **Define First Symbol.**

A. In a Context-Free Grammar (CFG), the first symbol is the start symbol, which is a special nonterminal symbol. The start symbol is used to derive strings in the language of the CFG.

2. **Define Pre-Processor.**

A. A preprocessor is a program that processes input data to create output that is used as input for another program, such as a compiler. It is a key component of a compiler that generates input for it.

3. **Define Bottom Up Parsing.**

A. Bottom-up parsing is a parsing technique that constructs the parse tree from the leaves (input symbols) to the root (start symbol) by reducing substrings to non-terminals using grammar rules.

4. **Define Syntax Analysis Phase.**

A. Syntax analysis phase is the second phase of a compiler that checks the source code for correct syntax using a context-free grammar and generates a parse tree.

5. **Define Semantic Rules.**

A. Semantic rules are rules in a syntax-directed definition used to compute attribute values and define the meaning of programming constructs.

6. **Define Attribute.**

A. An attribute is a value associated with a grammar symbol in a syntax-directed definition, used to store semantic information during compilation.

7. **Define Actual Parameters.**

A. Actual parameters are the values or expressions passed to a function or procedure during a function call. They provide input to the function for processing.

8. **Define Storage Allocation.**

A. Storage allocation is the process in a compiler of assigning memory locations to variables, constants, and data structures during program execution.

9. **Define Tape in Turing Machine.**

A. Tape in a Turing Machine is an infinite memory strip divided into cells, each holding a symbol. It serves as both input and working memory for the machine.

10. **Define Undecidabilty.**

A. Undecidability refers to problems for which no algorithm can be made that always gives a correct yes or no answer for all inputs in finite time.

SET – 3

1. **Define Compiler Process.**

A. Compiler process is the sequence of phases through which a source program is translated into machine code. It includes lexical analysis, syntax analysis, semantic analysis, intermediate code generation, optimization, and code generation.

2. **Define an Assembler.**

A. An assembler is a type of computer program that takes in basic instructions and converts them into a pattern of bits that the computer's processor can use to perform basic operations.

3. **Define Input Buffering.**

A. Input buffering is a technique used in lexical analysis to speed up reading of input characters by using buffers to minimize the number of I/O operations.

4. **Define Follow Symbol.**

A. A Follow symbol is a terminal that can appear immediately after a nonterminal in some derivation of a context-free grammar. It is part of the Follow set used in parsing.

5. **Define Grammar.**

A. A grammar is a set of rules that defines the syntactic structure of a language. It consists of terminals, non-terminals, a start symbol, and production rules.

6. **Define Directed Acyclic Graph.**

A. A Directed Acyclic Graph (DAG) is a finite directed graph with no cycles, used in compilers to represent expressions and optimize code by eliminating common subexpressions.

7. **Define Formal Parameters.**

A. Formal parameters are the variables declared in a function or procedure definition that receive values from actual parameters when the function is called.

8. **Define Heap Allocation.**

A. Heap allocation is a memory allocation technique where blocks of memory are dynamically allocated and deallocated at runtime from a region called the heap.

9. **Define Finite Control in Turing Machine.**

A. Finite control in a Turing Machine is the control unit that manages states and applies transition rules based on the current input.

10. **Define Recursive Language.**

A. A recursive language is a language for which there exists a Turing Machine that always halts and accepts or rejects any input string.