Unit 1 Introduction to System Software

Short answer questions:

1. Language processor is which type of software?
2. Compare and contrast De- translator and pre- processor.
3. Give example of system utility software.
4. Which are the two main components of computer?
5. Give example of system programming language.
6. Give example of application programming language.
7. What is a disadvantage of machine language?
8. Differentiate application software and system software.
9. Compare compiler and interpreter.
10. State difference between machine language and high level language.
11. What is specification gap?
12. What is execution gap?
13. List different types of user interface.
14. List different functions of operating system.
15. Why language migrator is used?
16. List different types of language processor.
17. List four types of application software.
18. What is system software?
19. Which software is used to convert C++ program into C program?
20. Which software is used to convert program written into one programming language into another programming language? Why?

Long answer questions:

1. Which are the components of computer? Explain each in brief.
2. Discuss system software.
3. Explain process of system software.
4. Write a note on Types of Software.
5. Write a difference between Application Software and System Software.
6. Write a note on computer programming and programming language.
7. Differentiate machine language and assembly language.
8. Write a note on Language Processors.
9. Explain Programming Development Cycle.
10. Write a difference between compiler and interpreter.
11. Explain specification and execution gap.
12. What is Language Processing? Explain spectrum of language processor.
13. Explain semantic gap.
14. What is system software? Write typical functionalities of system software.
15. Explain pre-processor and translator with example.
16. Which software is used to convert program written into one programming language into another programming language? Why?
17. Write four benefits of programming language domain.

Multiple Choice Questions:

1. For application based on 3D graphics which type of computer should be used?
   a. Mainframe Computers
   b. Minicomputers
   c. Workstations
   d. All of the above can be used
2. "Inventory Management Software" is which type of software?
   a. System Software
   b. Packages
   c. Customized Software
   d. None of the above
3. PL stands for
   a. Procedure Language
   b. Programming Language
   c. Periodic Language
   d. None of the above
4. Java is an example of _____ language.
   a. compiled
   b. interpreted
   c. hybrid
   d. script
5. The mother tongue of a computer is
   a. Assembly Language
   b. Machine Language
   c. BASIC Language
   d. None of the above
6. Mnemonics are used in:
   a. C Language
   b. Machine Language
   c. BASIC Language
   d. Assembly Language
7. Which is not a computer translator?
   a. Interpreter
   b. Compiler
   c. Word Processor
   d. Assembler
8. In _____ a low level language can be used:
   a. game
   b. simulator
   c. robots
   d. all of the above
9. The source code refers to:
   a. Original program code

b. Translated code
c. Instructions in mnemonic
d. Instructions in binary code

10. Which of these translates assembly code into machine code?
    a. Compiler
    b. Interpreter
    c. Editor
    d. Assembler
11. The program that combines the output of compiler with various library functions to produce an executable image is called –
    a. Loader
    b. Linker
    c. Assembler
    d. Debugger
12. What is true for the compiler?
    a. A compiler does a conversion line by line as the program is run.
    b. A compiler converts the whole of a higher level program code into machine code in one step.
    c. A compiler is general purpose language providing very efficient execution
    d. All of the above
13. What is not true for the interpreter?
    a. An interpreter executes the instructions line by line
    b. An interpreter will find the errors
    c. An interpreter will generate the object code
    d. None of the above
14. Interpreter is a:
    a. Language processor
    b. Compiler development phase
    c. Hardware
    d. All of the above
15. Gap between application domain and programming language domain is known as:
    a. Execution gap
    b. Specification gap
    c. Semantic gap
    d. Communication gap

State True or False with justification:

1. Output device is a part of hardware component.
2. MS-Office is system software.
3. Editor is used to find logical errors.
4. We always use workstations at our home.
5. Macro processor handles repeated code of program.
6. High level programming language is slower in execution than machine language.
7. We can directly design a solution without analysing the problem.

8. Task of program execution is performed in CPU.
9. Linker produces executable after linking process.
10. C++ is a higher level programming language.
11. Semantics is the study of meaning.
12. Execution of the program is done with the help of a language processor.
13. Execution gap is the semantic gap between two specifications of the same task.
14. A pre-processor is a language processor which bridges an execution gap but is not a language translator.
15. Coding is a first phase of program development.

Fill in the blanks:

1. _____ type of computer is specially made for engineering or scientific use.
2. ALU stands for _____.
3. _____ is widely used input device.
4. _____ handles input/output in computer.
5. _____ and _____ are two main type of software.
6. Compiler is a _____ type of system software.
7. _____ Provides features to have more direct access with computer's hardware.
8. The programming language that consists of a set of instructions represented as a series of 1s and 0s is _____.
9. _____ is a fastest type of programming language.
10. _____ is a machine dependent programming language.
11. _____ produces object code.
12. _____ translates the assembly language program into machine code.
13. _____ is a software which bridges a specification or execution gap.
14. _____ is a collection of programs that facilitate execution of programs and use of resources in a computer system.
15. Program produces _____ or _____ due to logical bugs.

Unit 2 Compiler

Short answer questions:

1. When can you call a compiler as a multi pass compiler?
2. Which phases of compiler are included in front end phase of a complier?
3. List the phases included in the back end phase of a compiler.
4. What is the input and output of syntax analysis phase of a compiler?
5. Which phases must be performed before semantic analysis phase of a complier?
6. Which phase generates a parse tree?
7. Which data are stored in symbol table?
8. What is the functionality of front end phase of a compiler?
9. What is a full form of DFA and NFA?
10. What is selected as root in top down approach of developing parse tree?
11. What is the purpose of LEX?
12. What is a use of symbol table in code generation phase of compiler?
13. Which are the four basic components of grammar?

14. What are terminal symbols?
15. What are non terminal symbols?
16. Differentiate front end of compiler and back end of compiler.
17. Differentiate between terminal and non terminal symbol.
18. List different parsing techniques.
19. Write drawback of recursive decent parsing techniques.
20. In which language Yacc is written?

Long answer questions:

1. What is compiler? Explain structure of compiler.
2. Write a note on analysis phase of compiler development.
3. Write a note on intermediate code generation of compiler development.
4. Write a note on code optimization of compiler development.
5. In which phase compiler can find an error? Explain each of such phases.
6. Explain how compiler manages symbol table.
7. Why compilers use intermediate representation? What does intermediate code look like? How is it different form final translated code?
8. Explain the different approaches to develop a compiler.
9. Write a note on: Types of Compiler.
10. Explain scanning.
11. Explain parsing.
12. Explain semantic analysis.
13. Explain the different notations used in grammar.
14. What is parsing? Explain the Recursive Descent techniques for parsing.
15. Write a note on Complier Development Tools.
16. Write a note on interpreter.

Multiple Choice Questions:

1. In which from the following phases, the compiler will not find the errors?
   a. Lexical Analysis
   b. Semantic Analysis
   c. Intermediate Representation
   d. None of the above
2. Which one is not true for lexical analysis?
   a. It recognizes the character stream and check the validity of it.
   b. It shows the error of use invalid datatype.
   c. It shows the error of invalid name of variable.
   d. All of the above are true.
3. Which data structure from the following is used in the compilation process?
   a. Symbol Table
   b. Parse Tree
   c. Transition Table
   d. Both a and b
   e. All a, b and c

4. What is true from following for intermediate representation?
   a. Intermediate Representation is same as binary code.
   b. Intermediate Representation makes the compilation process fast.
   c. In case of multi-platform compiler, intermediate representation is very useful.
   d. All are true
5. Symbol table stores
   a. Name of symbol used
   b. Information of constants
   c. Temporary variable's information
   d. All of the above
6. Select the correct regular expression for the language L = {ab^n / n >= 0}
   a. (ab)*
   b. (ab)+
   c. ab*
   d. ab+
7. Pattern matching of inputted words is done using –
   a. Regular Expression
   b. Grammar
   c. Both of the above
   d. None of the above
8. Which symbol specifies the set of inputs in DFA?
   a. Sigma
   b. Epsilon
   c. S
   d. Delta
9. Which is not a component of a grammar?
   a. States
   b. Terminals
   c. Production
   d. None of the above
10. Which is not a component of DFA?
    a. States
    b. Nodes
    c. Transitions
    d. Non-terminals
11. If the production is a -> b, then which type of grammar allow |a| = 1 only?
    a. Unrestricted Grammar
    b. Context Sensitive Grammar
    c. Context Free Grammar
    d. Regular Grammar
12. Which from the following is used for developing parser?
    a. Yacc
    b. Lex

c. Flex
d. Tree

13. Which from the following is used for developing lexical analyzer?
    a. Lex
    b. Yacc
    c. C language
    d. Both a. and b.
14. Which program converts one program into equivalent another program?
    a. Compiler
    b. Software package
    c. Machine level language
    d. Higher level language
15. Which symbols in a grammar stands for 'is defined as':
    a. :
    b. ::
    c. ::=
    d. ::==

State True or False with justification:

1. Compiler can have only single pass.
2. Front-end phase of a compiler is dependent on a source language.
3. Code optimization phase is always machine dependent.
4. Undefined variable is identified at lexical analysis.
5. The output of semantic analysis is a parse tree.
6. Optimization phase use symbol table to check unused variables.
7. Intermediate code is in binary language.
8. Native code compiler is machine dependent.
9. Binary search tree can be used for implementing symbol table.
10. Grammar is used for string comparison.
11. Final state is represented by circle in finite automata.
12. There can be only one starting state in DFA.
13. There can be multiple starting states in NFA.
14. DFA can have only one state transition for every symbol of the alphabet.
15. Context sensitive grammar can have single non-terminal on left side of production rule.

Fill in the blanks:

1. _____ phase is dependent of source language.
2. Syntax error is identified in _____ phase.
3. _____ phase generates parse tree.
4. _____ is an input to the machine code generation phase of compiler.
5. _____ compiler is designed to compile a source code for different platforms.
6. Length of string X in grammar is denoted by _____ .
7. More than zero occurrences in string r can be represented by _____ in regular expression.

8. In finite automata states are represented by _____ shape.
9. _____ grammar can have a single non-terminal of left side of production rule.
10. Yacc stands for _____.
11. _____ data structure is used by interpreter.
12. Noun and verb are considered as _____ symbols in programming language grammar.
13. A _____is a compiler which runs on one machine and generates a code for another machine.
14. Early compilers were mostly coded in _____language.
15. _____ and _____ are widely used language processor development tool.

Unit 3 Assembler and Macro Processor

Short answer questions:

1. What is a main functionality of execution unit of 8086?
2. Write a purpose of Bus Interface Unit of 8086.
3. What is a format of assembly language statement?
4. Which are the three main types of mnemonic?
5. What is the functionality of pass 1 of assembler?
6. What does pass 2 of assembler do?
7. What is a use of location counter?
8. List tasks which are involved in macro expansion.
9. List steps to arrive at a design specification of macro processor
10. List the various assembler directives.
11. Write definition of assembler.
12. What do you mean by Mnemonic in assembly language?
13. What is an imperative statement in assembly language?
14. What is declarative statement in assembly language?
15. What is a function of macro processor?
16. List different types of field available in symbol table?

Long answer questions:

1. Explain the architecture of 8086.
2. Explain functionality of each flag register in 8086.
3. What is assembly language? Explain the three basics facilities of assembly language.
4. Write a short note on design of assembler.
5. Write a note on design of macro pre-processor.
6. Write a note on data structure of the macro pre-processor.
7. Explain one pass macro processor algorithm.
8. Describe Pass I of the assembler.
9. Discuss processing of declaration and assembler directives.
10. What are assembler directives? Write benefit of assembly language.
11. Write a note on assembly language statements.
12. Differentiate between forward reference and cross reference.
13. Discuss design overview and procedure to arrive at design specification.

Multiple Choice Questions:

1. 8086 is a _____ Microprocessor
   a. 8 –bit
   b. 16 bit
   c. 32bit
   d. 24 bit
2. The 8086 is a _____ bit microprocessor.
   a. 16
   b. 20
   c. 32
   d. 64
3. BIU stands for:
   a. Bus interface unit
   b. Bess interface unit
   c. A and B
   d. None of these
4. EU stands for:
   a. Execution unit
   b. Execute unit
   c. Exchange unit
   d. None of these
5. Which are the part of architecture of 8086:
   a. The bus interface unit
   b. The execution unit
   c. Both A and B
   d. None of these
6. CS Stand for:
   a. Code segment
   b. Coot segment
   c. Cost segment
   d. Counter segment
7. DS Stands for:
   a. Data segment
   b. Direct segment
   c. Declare segment
   d. Divide segment
8. Which are the segments in 8086 microprocessor?
   a. CS: Code segment
   b. DS: data segment
   c. SS: Stack segment
   d. ES:extra segment
   e. All of these
9. How many types of addressing are in memory?

a. Logical address
b. Physical address
c. Both A and B
d. None of these

10. PA stands for:
    a. Project address
    b. Physical address
    c. Pin address
    d. Pointer address
11. EA stands for:
    a. Effective address
    b. Electrical address
    c. Effect address
    d. None of these
12. SI stands for:
    a. Stand index
    b. Source index
    c. Segment index
    d. Simple index
13. PC stands for:
    a. program counter
    b. project counter
    c. protect counter
    d. planning counter
14. Which are the general registers?
    a. AX: Accumulator
    b. BX: Base
    c. CX: Count
    d. DX: Data
    e. All of these
15. Symbol table has _____ primary fields.
    a. mnemonic and opcode
    b. name and address
    c. mnemonic and name
    d. name and opcode
16. Mnemonic has _____ primary fields.
    a. mnemonic and opcode
    b. name and address
    c. mnemonic and name
    d. name and opcode
17. The primary function performed by the analysis phase is the building of the _____.
    a. mnemonic table

b. analysis table
c. synthesis table
d. symbol table

18. The function of fixing the address of all program elements is known as _____.
    a. memory allocation
    b. memory relocation
    c. memory management
    d. paging
19. _____ is a data structure to implement memory allocation.
    a. Location table
    b. Location count
    c. Location counter
    d. Location area
20. _____ table is fixed table which is merely accessed by the analysis and synthesis phase.
    a. Mnemonic table
    b. Analysis table
    c. Synthesis table
    d. Symbol table
21. _____ table is constructed during the analysis and used during synthesis.
    a. Mnemonic table
    b. name table
    c. Synthesis table
    d. Symbol table
22. Which of the following is not a task of pass-I assembler?
    a. Build the symbol table
    b. Perform LC processing
    c. Synthesize the target program
    d. Construct intermediate representation
23. To update the contents of LC,_____ needs to know lengths of different instruction.
    a. synthesis phase
    b. analysis phase
    c. none of the above
    d. all of the above
24. An assembly language is a machine dependent, _____ level programming language which is specific to a certain computer system.
    a. high
    b. low
    c. machine
    d. none of these
25. The _____ performs memory bindings to symbolic names.

a. assembler
b. compiler
c. translator
d. analyzer

26. In a simple assembly language, the first operand is always:
    a. Memory Word
    b. Register
    c. Assembly Mnemonic
    d. None of these
27. In a simple assembly language, the second operand refers to_____.
    a. memory word
    b. register
    c. data declarations
    d. all of the above
28. The DS stands for:
    a. Data Store
    b. Date Storage
    c. Declare Storage
    d. Declare Statement
29. Which assembler directive indicates that the first word of the target program generated by the assembler should be placed in the memory word with address.
    a. LABEL
    b. END
    c. START
    d. STOP
30. Which assembler directive indicates that the end of the source program.
    a. EXIT
    b. END
    c. START
    d. LABEL END

State True or False with justification:

1. Accumulator register is a 16 bit register in 8086.
2. Pointer register is used to access data in stack segment.
3. Instruction Queue is a part of EU which is having queue of instructions to be executed.
4. Assembly language is machine independent.
5. Assembler never provides error in code.
6. Assembly language statement can have maximum two operands.
7. Pass 1 of assembler is responsible for assigning addresses to all statements in program.
8. LOCCTR is always initialized to zero.
9. Forward reference can be solved with the use of multi pass assembler.
10. Macros are used in assembly language only.

11. Macros can have any number of parameters.
12. Macro code increases the length of actual code.
13. Semantic expansions are done according to requirement of macro call.
14. Keyword parameters are stored in KPD Table.
15. EVNTAB is filled when definition of any EV is encountered in macro body.
16. MNT is a main table in which body of macro is stored.
17. You must declare all the macros before you call them

Fill in the blanks:

1. _____ register contains value to control the number of times a loop is repeated.
2. _____ and _____ is used to access data in stack segment.
3. _____ number of flags in 16-bit flag register are not used.
4. Output code of assembler is in _____ language.
5. _____ mnemonics are not converted into machine language.
6. LOCCTR is initialized to the beginning address specified by the _____ mnemonic.
7. _____ table is a static table which contains mnemonic operation codes.
8. _____ table includes the name and value(address) for each label in the source program.
9. The reference created for a symbol table for any undefined variable is called _____.
10. A unit which represents a commonly used group of statements and which replaces that group of statement statements when it is called is called _____.
11. Macro definition is placed between _____ and _____ statements.
12. _____ expansion includes replacement of character strings by another character string.
13. In _____ parameter type parameters are replaced on the basis of name given to them.
14. In _____ type of macro expansion few statements are never visited.
15. _____ table holds the name of all the parameters appear in the macros.

Unit 4 Loader and Linker

Short answer questions:

1. What is linker?
2. What is loader?
3. List steps for execution of program written in programming language.
4. Why origin of a program may have to be changed by the linker or loader?
5. Write meaning of translation time, linked address and load time.
6. What is static linking?
7. What is dynamic linking?
8. Write one limitation of absolute loader.
9. Which information are stored in binary program of absolute loader?
10. Write one function of bootstrap loader.
11. What is relocating loader?

12. Which information are stored in binary program of relocating loader.
13. What is object file?
14. List components of object module.
15. List data structure needed for linking loader.
16. Which tasks are performed in first pass of algorithm for linking loader?
17. Which tasks are performed in second pass of algorithm for linking loader?
18. Write one advantage of reference number mechanism in linking loader algorithm.
19. When it is better to use linkage editor?
20. What is a dis-advantage of linkage editor?

Long answer questions:

1. Discuss various steps for execution of program written in a programming language.
2. Write a note on dynamic linking.
3. What is loader? List types of loader. Explain each loader in brief.
4. Write a note on absolute loader.
5. Write a note on bootstrap loader.
6. Write a note on relocating loader.
7. Explain algorithm for an absolute loader.
8. Write a note on object file.
9. Write algorithm for pass I of linking loader.
10. Write algorithm for pass II of linking loader.
11. Differentiate linking loader and linkage editor.
12. Write a note on linkage editor.
13. Describe schematic of a program's execution.
14. Differentiate between static linking and dynamic linking.
15. Explain processing of an object program using linking editor and linkage editor.

Multiple Choice Questions:

1. The _____ of a program is the address of the instruction from which its execution must begin.
    a. execution address
    b. address
    c. execution start address
    d. all of above
2. The start address specified by the translator is the _____ of the program.
    a. translated start address
    b. linked start address
    c. load address
    d. none of these
3. Address of the origin assumed by the translator is known as _____.
    a. linked origin
    b. load origin
    c. translated origin
    d. origin

4. Address of the origin assigned by the linker while producing a binary program is called _____.
    a. load origin
    b. linked origin
    c. translated origin
    d. none of these
5. Address of the origin assigned by the loader while loading the program for execution is known as _____.
    a. translated origin
    b. linked origin
    c. origin
    d. load origin
6. _____ is the process of modifying the addresses used in the address sensitive instructions of a program.
    a. Program relocation
    b. Program
    c. Program location
    d. None of above
7. In performing relocation, _____ can be positive, negative or zero.
    a. relocation_factorp
    b. l_originp
    c. t_originp
    d. none of these
8. _____ is the process of binding an external reference to the correct link time address.
    a. Translation
    b. Linking
    c. Loading
    d. Assembler
9. A _____ performs relocation while loading a program for execution.
    a. relocating linker
    b. relocating translator
    c. relocating loader
    d. relocating assembler
10. Which of the following system software reside in main memory always?
    a. Text editor
    b. Assembler
    c. Linker
    d. Loader
11. The liker is:
    a. same as the loader
    b. required to create a load module
    c. is always used before programs are executed

d. none of the above

12. Which of the following loader is executed when a system is first turned on or restarted?
    a. Boot loader
    b. Compile and Go loader
    c. Bootstrap loader
    d. Relating loader
13. A linker program
    a. places the program in the memory for the purpose of execution.
    b. relocates the program to execute from the specific memory area allocated to it.
    c. links the program with other programs needed for it s execution.
    d. interfaces the program with the entities generating its input data.
14. Relocation bits used by relocating loader are specified by
    a. relocating loader itself
    b. assembler or translator
    c. macro processor
    d. both (a) and (b)
15. Resolution of externally defined symbols is performed by
    a. linker
    b. loader
    c. compiler
    d. editor
16. Relocatable programs
    a. cannot be used with fixed partitions
    b. can be loaded almost anywhere in memory
    c. do not need a linker
    d. can be loaded only at one specific location
17. _____ in system software resides in main memory always.
    a. Linker
    b. Loader
    c. Text editor
    d. Assembler
18. A system program that combines the separately compiled modules of a program into a form suitable for execution?
    a. Assembler
    b. Linking loader
    c. Cross compiler
    d. Load and Go
    e. None of above

State True or False with justification:

1. Linking translate a program into target program.
2. Relocation locates the program into different memory area other than the program

been coded.

3. Object file is used for storage of object code and related data.
4. Object files are produced by an assembler, compiler, or other language translator.
5. Object file works as an input for assembler.
6. Dynamic loading means loading the library (or any other binary for that matter) into the memory during load or run-time.
7. Dynamic linker is a compile time program that loads and binds all of the dynamic dependencies of a program before starting to execute that program.
8. Dynamic loading is handled by operating system.
9. Dynamically linked shared libraries are easier to create than static linked shared libraries.
10. Absolute loaders link a program for execution.
11. Bootstrap loader loads operating system into ROM when a computer's power is on.
12. Relocating loader loads a program in a designed area of memory.
13. The algorithm for a linking loader is considerably simple than absolute loader algorithm.
14. External Symbol Table is a main data structure for linking loader.
15. A linkage editor produces linked version of program which is written to a file or library for later execution.

Fill in the blanks:

1. Address assigned by a translator is called _____.
2. If translation time address is not same as the linked time address, address relocation is performed by _____.
3. Loader loads the program in _____.
4. _____ is responsible for relocation when object program is loaded into memory.
5. _____ loader converts the source code into machine instructions line by line and put them at already known address for execution.
6. _____ loader does not perform linking or relocation.
7. Bootstrap loader is used when _____.
8. _____ means replacing a block of instructions or data with other.
9. _____ is a beginning address in memory where the linked program is to be loaded.
10. Value of length of control section is found in _____.
11. _____ data structure is used to maintain external symbol table.
12. Binding several logical spaces into one composite logical space is called _____.
13. _____ performs all linking and relocation operations.
14. _____ produces a lined version of the program.
15. _____ brings an object program into memory and start its execution.

Unit 5 Software Tools

Short answer questions:

1. What is a software tool?

2. List down fundamental steps in program development.
3. Which categories are used in program design and coding?
4. Define: Program generator
5. Define: Program Environment
6. List down steps for program testing and debugging.
7. List down components of program environment.
8. What is a user interfaces?
9. What is a command dialog design?
10. List out principles of command dialog design.
11. Define: Profile Monitor
12. What is Command Language?
13. What is Command Menu?
14. What is Direct Manipulation?
15. What is Presentation of Data?
16. Write difference between Hypertext, Hyper card and Manually
17. What is program design and coding?
18. What is program entry and Editing?
19. What is test data?
20. What is the purpose of test data generator?
21. Efficiency of programs depends on which two factors?
22. What is the main difference between line editor and stream editor?
23. Give example of screen editor.
24. Give example of word processor.
25. Give example of structured editor.
26. What is the basic difference between viewing and displaying?
27. What is a full form of IDE?
28. What is program documentation aids?
29. Draw a diagram explaining structure of an editor.
30. Write definition of Editor.
31. How programming environment is useful to the user?
32. List different components of IDE

Long answer questions:

1. Explain Program Testing.
2. Discuss way to implement command dialogs.
3. Explain user interface management system.
4. Define Software Tools. Discuss three software tools for programming development.
5. Briefly discuss the principles of command dialog design.
6. Define Editor. Explain five different types of editors.
7. Write a note on structured editor. Explain by giving example.
8. Differentiate between screen editor and word processor.
9. Which are the four fundamental functions performed in editor? Explain each in brief.
10. Write a note on structure of editor.

11. Write a note on design of editor.
12. Explain step by step procedure followed by any editor when user has given any command.
13. Write a note on user interface of an editor.
14. Explain two basic components of user interface of editor.
15. Write a note on programming environment.
16. Which are the four basic categories in which programming environment can be classified? Explain each in brief.
17. Explain features of IDE.
18. Explain program documentation, program pre-processing and instrumentation, program interpretation and program generation in brief.

Multiple Choice Questions:

1. A _____ views the entire text as a stream of characters.
    a. stream editor
    b. screen editor
    c. line editor
    d. structure editor
2. _____ support character, line and context oriented commands based on the current editing context indicated by the position of a text pointer.
    a. Screen editors
    b. Word processors
    c. Stream editors
    d. Structure editors
3. _____ and _____ editors maintain multiple representations of text.
    a. Screen, stream
    b. Line, stream
    c. Stream, structure
    d. Structure, screen
4. The _____ maintains an internal form which is used to perform the edit operations.
    a. editor
    b. loader
    c. linker
    d. assembler
5. Word processors are also called _____ editors.
    a. screen
    b. structure
    c. stream
    d. document
6. A special class of structure editors, called _____ editors, are used in programming environments.
    a. syntax
    b. syntax directed

c. line
d. stream

7. _____ generates a program which performs a set of function described in its specification.
   a. Software Tools
   b. Program Generator
   c. Program Environment
   d. Language Processor
8. _____ supports program coding by incorporating awareness of the program language syntax and semantics in the language editors.
   a. Program Generator
   b. Program Environment
   c. Software Tools
   d. Language Processors
9. _____ accepts user commands specifying the editing function to be performed.
   a. Data Mode
   b. Command Mode
   c. Language Processor
   d. Software Tools
10. _____ helps the user in selecting test data for his program.
   a. Program Environment
   b. Language Processor
   c. Software tools
   d. Test data generator
11. _____ is responsible for managing the user's screen and accepting data and presenting results.
   a. Debug Monitors
   b. Presentation Manager
   c. Dialog Manager
   d. All of Above
12. _____ is responsible for interpreting user command and implementing them by invoking different modules of the application code.
   a. Dialog Manager
   b. Debug Manager
   c. Presentation Manager
   d. All of Above
13. Which is not the fundamental step in program development?
   a. Program design, coding and documentation
   b. Preparation of programs in machine readable form.
   c. Program translation, linking and loading
   d. None of the above
14. Which tools are used in program design and coding?

a. Program generator
b. Program environments
c. Boat a and b
d. None of the above

15. A software environment consist of:
    a. A syntax directed editor
    b. A language processor
    c. Dialog monitor
    d. All of the above
16. Hypertext uses which protocol to fetch and transmit web pages when a user clicks on a hyperlink.
    a. HTTP
    b. HTTPS
    c. SSL
    d. POP
17. Command issued to an application can be implemented in which ways:
    a. Command languages
    b. Command menus
    c. Direct manipulation
    d. All of the above
18. Which is not a fundamental function in editing :
    a. Travelling
    b. Editing
    c. Viewing
    d. Correcting
19. A software tool is a :
    a. Editors
    b. Debug monitors
    c. Programming environment
    d. All of the above
20. A user can interact with the application through:
    a. Command
    b. Middle man
    c. Camera
    d. WWW

State True or False with justification:

1. Editor is a type of software tool.
2. Software tool is an application program.
3. Interpreter and compiler are type of language processor.
4. Command mode and what-you-see-is-what-you-get mode is used for program entry and editing.
5. Efficiency of program depends only on efficiency of the algorithm.
6. A compiler can improve the efficiency of the algorithm.

7. Program documentation consists of: flow charts, IO specification, cross references in formation etc.
8. A programming environment provides integrated facilities for program creation, editing, execution, testing and debugging.
9. A user interface provides interaction of a user with another user.
10. A source code management system helps to ensure consistency of a program during debugging and modification.
11. An infeasible path is simply a path that may not be traversed during execution of a program.
12. A source code management system is used to decide whether a modification is correct.
13. To handle queries of an ad hoc kind, one should develop a program generator rather than and interpreter.
14. Dynamic debugging is easier to implement in interpreter than in compiler.
15. Hypercard uses an interpretive schematic to implement a UI.

Fill in the blanks:

1. System program is an interface between program and _____.
2. _____ And _____ are two UIMS using the event description approach.
3. An optimizing compiler can improve the efficiency of _____.
4. A _____ is a software tool that uses special techniques making and doing of program modifications in a consistent manner.
5. Most of the programming projects suffer from lack of up-to-date_____.
6. Program pre-processing techniques are used to support _____ analysis of programs.
7. The undo function can be implemented by _____.
8. Automated test drivers help in _____.
9. _____ path is an execution path that cannot be traversed for any set of input values.
10. _____ promote user's interest to use an application.
11. Large volume of data can be effectively presented through _____.
12. The _____ is the most extensive example of hypertext.
13. Components of User Interface are _____ and _____.
14. _____ is an example of command dialog through direct manipulation.
15. _____ software collects information regarding execution time consumed by its module.