**Sample Questions**

Computer Engineering

**Subject Name:** System Programming and Compiler Construction **Semester: VI**

Multiple Choice Questions

|  |  |
| --- | --- |
|  | **Choose the correct option for following questions. All the Questions carry equal marks** |
| 1. | Which of the following is designed to control the operations of a computer? |
| Option A: | Application Software |
| Option B: | **System Software** |
| Option C: | Utility Software |
| Option D: | User |
|  |  |
| 2. | A person who designs the programs in a software package is called : |
| Option A: | User |
| Option B: | Software Manager |
| Option C: | System Developer |
| Option D: | **System Programmer** |
|  |  |
| 3. | Assembler is used as a translator for? |
| Option A: | **Low level language** |
| Option B: | High Level Language |
| Option C: | COBOL |
| Option D: | C |
|  |  |
| 4. | They normally interact with the system via the user interface provided by the application software. |
| Option A: | Programmers |
| Option B: | Developers |
| Option C: | **Users** |
| Option D: | Testers |
|  |  |
| 5. | Storage mapping is done by\_\_\_\_\_\_\_\_\_\_ |
| Option A: | Linker |
| Option B: | **Compiler** |
| Option C: | Loader |
| Option D: | Operating system |
|  |  |
| 6. | Interpreter is used as a translator for \_\_\_\_\_\_\_\_\_\_ |
| Option A: | Low level language |
| Option B: | **High Level Language** |
| Option C: | COBOL |
| Option D: | C |
|  |  |
| 7. | A system program that set up an executable program in main memory ready for execution is |
| Option A: | **Loader** |
| Option B: | Linker |
| Option C: | Assembler |
| Option D: | load and go |
|  |  |
| 8. | The \_\_ of a system includes the program s or instructions. |
| Option A: | Icon |
| Option B: | **Software** |
| Option C: | Hardware |
| Option D: | Information |
|  |  |
| 9. | Instructions which won’t appear in the object program are called as \_\_\_\_\_ |
| Option A: | Redundant instructions |
| Option B: | Exceptions |
| Option C: | Mnemonic opcode |
| Option D: | **Assembler Directives** |
|  |  |
| 10. | The last statement of the assembly program should be \_\_\_\_\_\_\_ |
| Option A: | STOP |
| Option B: | RETURN |
| Option C: | TERMINATE |
| Option D: | **END** |
|  |  |
| **11.** | Translator for low level programming language were termed as |
| Option A: | **Assembler** |
| Option B: | Compiler |
| Option C: | Linker |
| Option D: | Loader |
|  |  |
| **12.** | The Macro processor is also called as \_\_\_\_\_\_\_ |
| Option A: | **Preprocessor** |
| Option B: | Postprocessor |
| Option C: | Debugger |
| Option D: | Translator |
|  |  |
| **13.** | In parameterised macro , the parameter is mapped using\_\_\_\_ |
| Option A: | **by position** |
| Option B: | by keyword |
| Option C: | by reference |
| Option D: | by string |
|  |  |
| **14.** | The linker is a software that is used for\_\_\_\_\_\_ |
| Option A: | **Creating signle executable load module** |
| Option B: | Excecuting the program |
| Option C: | Creating link between program and data |
| Option D: | Helping loader to load program in memory |
|  |  |
| **15.** | Which is not a function of a loader |
| Option A: | Allocation |
| Option B: | **Translation** |
| Option C: | Relocation |
| Option D: | Loading |
|  |  |
| **16.** | Which of the following software always resides in main memory? |
| Option A: | Text editor |
| Option B: | Assembler |
| Option C: | Linker |
| Option D: | **Loader** |
|  |  |
| **17.** | **What type of data structure is used by shift reduce parser** |
| Option A: | linked list |
| Option B: | **Stack** |
| Option C: | Queue |
| Option D: | Pointer |
|  |  |
| **18.** | We can optimize code by |
| Option A: | **Dead code elimination** |
| Option B: | Common subprogram |
| Option C: | Copy intermediate loop |
| Option D: | Loop declaration |
|  |  |
| **19.** | Local and loop optimization in turn provide motivation for |
| Option A: | **Data flow analysis** |
| Option B: | Constant folding |
| Option C: | Pee hole optimization |
| Option D: | DFA |
|  |  |
| **20.** | Compiler can check \_\_\_\_\_\_\_\_ error |
| Option A: | Logical |
| Option B: | **Syntax** |
| Option C: | both a and b |
| Option D: | Content |

Descriptive Questions

|  |
| --- |
| What is the forward reference problem? Explain single pass assembler with flowchart. |
| Explain multi pass assembler in detail |
| Show machine code generated for following assembly level program along with data structures entries |
| Explain single pass macro processor |
| Explain the working of macro processor along with the data structures used in it |
| Explain the working of DLL loader in detail. |
| Draw and Explain the various phases of compilers with suitable example. |
| Modify the given grammar and construct a Predictive parser table explaining each step.  E->E+T|T T->T\*V|V V-> id. |
| For a given grammar below, Construct operator precdence relation matrix, assuming \*, + are binary operators and 'id' is terminal symbol, and E as Non terminal. E->E+E E->E\*E E->id Apply operator precedence parsing algorithm for the statement ' id + id \* id' |
| Consider the following grammar: S --> aSbS | bSaS | Epcillon.  1. Frame the transition table and action / goto table of the given grammar.  2. Demonstrate if the grammar is LR(0) or not. |
| Explain the working of shift reduce parser along with suitable example |
| Explain the different forms of intermediate codes used by Compiler. |
| What is code optimization? Explain machine dependent code optimisation techniques with suitable example |
| Explain machine independent code optimization techniques with suitable example |
| Discuss various issues that occur in the code generation phase of the compiler. |
| Explain the difference between Compiler and Interpreter |
| Define the various system softwares used in compilers |
| What is the need of system softwares? |
| Explain various data structures used in assembler design |
| What is the need of an assembler to be multi pass? |
| Explain various types of statements used in assembler design |
| What are the different functions performed by macroprocessor? |
| Explain Parameterized macro with suitable example |
| Explain conditional macro with suitable example. |
| What are the different functions performed by loader |
| Enlist different types of noodles and explain compile and go loader in detail |
| Explain the working of absolute loader. |
| What do you mean by relocation? Explain relocating loader in detail. |
| Explain the difference between linking loader and linkage editor. |
| Explain the working of compiler phases for following expression Position = initial + rate \* 60. |
| Explain the role of finite automata in lexical analysis |
| Design DFA for given finite automata. (a+b)\*abb |
| Differentiate between top down and bottom up parser. |
| Define synthesized and inherited attributes used in Syntactic analysis of compiler. |
| Generate three address code for the following logical expression. If a<b then 1 else 0 |
| Design quadruple and triple for following expression a=(b+c)\*(d+e) |
| Design DAG representation for given expression. a=(a+b)\*(a-c) |
| Explain flow graphs and basic blocks in detail. |
| Write a short note on LEX and YACC. |