Program: BE ------------------------------- Engineering

Curriculum Scheme: Revised 2016/2012

Examination: Third Year Semester V/VI

Course Code: \_\_\_\_\_ and Course Name: \_\_\_\_

Time: 1 hour Max. Marks: 50

==============================================================================

NOTE to the Question Bank Generator:

1. The question bank consists of 25 MCQ questions with each question carrying a maximum of 2 marks. It should cover all the modules with appropriate weightages.

2. You need to check the questions and their answers for their correctness. There should not be any ambiguity in the questions and the options. Only one option should be the Correct Answer.

3. You must ensure that the same question is not repeated again in this question paper.

4. Among 25-questions, 13 questions can be under the ‘Simple’ category, 7-questions can be under the ‘Moderate’ category, and the remaining 5-questions can be under the ‘Difficult’ category.

5. Please do not reveal answer on this Question Paper.

6. Use another template provided to enter the correct answers.

7. Please save this file with file name as per the sample format given below:

File Name: “Date of Examination\_Scheme\_Program\_Semester\_Subject Code\_QP Set Number”

For example:

QP set number 1 of first core course of Mechanical Engineering Semester V for Rev2016 scheme and scheduled on 25/09/2020 has to have the file name as

**2509\_R16\_Mech\_V\_MEC501\_QP1**

QP set number 1 of Department Level Optional Course of Computer Engineering Semester VI for Rev2012 scheme and scheduled on 28/09/2020 has to have the file name as

**2809\_R12\_Comp\_VI\_CSDLO6021\_QP1**

=============================================================================

Note to the students:- All the Questions are compulsory and carry equal marks .

|  |  |
| --- | --- |
| Q1.  | When dealing with the branching code, the assembler |
| Option A: | Replaces the target with its address |
| Option B: | Does not replace until the test condition is satisfied |
| Option C: | Finds the Branch offset and replaces the Branch target with it |
| Option D:  | Replaces the target with the value specified by the DATAWORD directive |
|  |  |
| Q2. | In a two-pass assembler, the task of the Pass II is to |
| Option A: | separate the symbol, mnemonic opcode and operand fields. |
| Option B: | build the symbol table.  |
| Option C: | construct intermediate code. |
| Option D: | synthesize the target program.  |
|  |  |
| Q3. | Address symbol table is generated by the |
| Option A: | memory management software |
| Option B: | assembler |
| Option C: | match logic of associative memory |
| Option D: | generated by operating system |
|  |  |
| Q4. | Which assembler directive indicates that the first word of the target programgenerated by the assembler should be placed in the memory word with address. |
| Option A: |  LABEL |
| Option B: | END |
| Option C: | START |
| Option D: | STOP |
|  |  |
| Q5. | Assembly language programs are written using |
| option A: |  Hex code |
| Option B: | Mnemonics |
| Option C: | ASCII code |
| Option D:  | BCDDIC |
|  |  |
| Q6. | Assembly language program is called: |
| Option A: |  Object program |
| Option B: | Source program |
| Option C: |  Oriented program |
| Option D:  | Machine program  |
|  |  |
| Q7.  |  By whom address of external function in the assembly source file supplied by \_\_\_\_\_\_ when activated: |
| Option A: | Assembler |
| Option B: |  Linker |
| Option C: | Machine |
| Option D:  | Code |
|  |  |
| Q8.  | The assembler in first pass reads the program to collect symbols defined with offsets in a table\_\_\_\_\_\_\_: |
| Option A: |  Hash table |
| Option B: |  Symbol table |
| Option C: | Literal table |
| Option D:  | Machine Opcode table  |
|  |  |
| Q9. |  A Label is separated by \_\_\_\_\_ from the mnemonic |
| Option A: | semicolon  |
| Option B: |  colon  |
| Option C: | fullstop |
| Option D:  | space |
|  |  |
| Q10.  | Which one is NOT a branching operation? |
| Option A: | JUMP  |
| Option B: | CALL |
| Option C: | RETURN  |
| Option D:  | MOV |
|  |  |
| Q11.  | A linker is given object module for a set of programs that were compiled separately. What information need not be included in an object module |
| Option A: | Object code |
| Option B: | Relocation bits |
| Option C: | Absolute addresses of internal symbols |
| Option D:  | Names and locations of all external symbol |
|  |  |
| Q12.  | When a computer is first turned on or restarted, a special type of absolute loader, called a \_\_ is executed |
| Option A: | bootstrap loader |
| Option B: | loader |
| Option C: | linker |
| Option D: | Execution |
|  |  |
| Q13. | Loaders that allow for program relocation are called |
| Option A: | relative loaders |
| Option B: | relocate loaders |
| Option C: | boaster loader |
| Option D:  | header shooter |
|  |  |
| Q14.  | A relocate program form is the one which |
| Option A: | consists of a program and relevant information for its relocation |
| Option B: | can itself perform the relocation of its address-sensitive portions |
| Option C: | cannot be made to execute in any area of storage other than the one designated for it at the time of its coding or translation |
| Option D:  | Execution level |
|  |  |
| Q15. | A loader is |
| Option A: | a program that places programs into memory and prepares them for execution |
| Option B: | a program that automate the translation of assembly language into machine language |
| Option C: | a program that accepts a program written in a high level language and produces an object program |
| Option D:  | is a program that appears to execute a source program if it were machine language |
|  |  |
| Q16.  | A linker is given object module for a set ofprograms that were compiled separately. What information need not be included in an object module |
| Option A: | Object code |
| Option B: | Relocation bits |
| Option C: | Absolute addresses of internal symbols |
| Option D:  | Names and locations of all external symbols denied in the object module |
|  |  |
| Q17. | Which of the following functions is performed by loader ? |
| Option A: | Adjust all address dependent locations, such as address constants, to correspond to the allocated space |
| Option B: | Allocate space in memory for the programs and resolve symbolic references between objects decks |
| Option C: | Physicaly place the machine instructions and data into memory |
| Option D: | program that automates the translation |
|  |  |
| Q18. | In a resident -OS computer, which of the following systems must reside in the main memory under all situations |
| Option A: | Assembler |
| Option B: |  Linker |
| Option C: | Loader |
| Option D:  | Compiler |
|  |  |
| Q19.  | In distributed system,the capacity of the system to adapt the increase services load is called |
| Option A: | Tolerance |
| Option B: | Scalability |
| Option C: | Capability |
| Option D:  | Loading |
|  |  |
| Q20. | Relocation is used by relocation loader are |
| Option A: | Relocation loader itself |
| Option B: | Linker |
| Option C: | Assembler |
| Option D: | Macro Processor |
|  |  |
| Q21. | What is the name of the process that determining whether a string of tokens can be generated by a grammar? |
| Option A: | Analysing |
| Option B: | Recognizing |
| Option C: | Translating |
| Option D:  | Parsing |
|  |  |
| Q22.  | A \_\_\_\_\_\_\_\_\_ is a software utility that translates code written in higher language into a low level language |
| Option A: | converter |
| Option B: | compiler |
| Option C: | text editor |
| Option D:  | code optimizer |
|  |  |
| Q23. | Which of the following groups is/are token together into semantic structures? |
| Option A: | Syntax analyzer |
| Option B: | Intermediate code generation |
| Option C: | Lexical analyzer |
| Option D:  | Semantic analyzer |
|  |  |
| Q24.  | How many parts of compiler are there? |
| Option A: | 1 |
| Option B: | 2 |
| Option C: | 4 |
| Option D:  | 8 |
|  |  |
| Q25. | What is the action of parsing the source program into proper syntactic classes? |
| Option A: | Lexical analysis |
| Option B: | Syntax analysis |
| Option C: | General syntax analysis |
| Option D:  | Interpretation analysis |