S-855 Roll No. ……..

BCA–20

System Programming

Bachelor of Computer Application (BCA-11/16/17)

Sixth Semester, Examination, 2018

Time : 3 Hours **Max. Marks : 80**

Note : This paper is of **eighty (80)** marks containing **three (03)** Sections A, B, C. Attempt the questions contained in these Sections according to the detailed instructions given therein.

Section–A

(Long Answer Type Questions)

Note : Section 'A' contains four (04) long answer type questions of nineteen (19) marks each. Learners are required to answer *two* (02) questions only.

1. What are the Advanced Assembler Directives ? An assembly program contains the statement :

   X EQU Y + 25

   Indicate how the EQU statement can processed if :

   (i) Y is a back reference.

   (ii) Y is a forward reference.

2. List various phases of Language Processor. Explain any *one* phase in detail.

3. Explain types of grammars. Write a regular expression for a language containing a binary string which does not contain two consecutive 0s or two consecutive 1s anywhere.
4. Consider the statement a = b + c * 10, where a, b, c are of type float. Show the translation of the given statement by different phases of compiler to produce assembly language statement.

Section–B

(Short Answer Type Questions)

Note : Section 'B' contains eight (08) short answer type questions of eight (08) marks each. Learners are required to answer *four* (04) questions only.

1. What is peephole optimization ? Explain any *two* optimization transformation in detail.
2. What is pure and impure interpreter ? Explain design of an editor.
3. Explain Absolute Loader with example.
4. What is main task of semantic analysis phase ? Explain inherited and synthesized attributes in detail with example.
5. State different storage allocation strategies. Explain static allocation and stack allocation in detail.
6. Explain left recursion, left factoring and backtracking in top down parsing.
7. What is macro in programming language ? Write an algorithm for macro definition.
8. Explain the following :
   (i) Execution Gap
   (ii) Compiler
   (iii) Interpreter
   (iv) Language Migrator

Section–C

(Objective Type Questions)

Note : Section 'C' contains ten (10) objective type questions of one (1) mark each. All the questions of this Section are compulsory.

1. The output of lexical analyser is :
   (a) A set of regular expressions
   (b) Syntax tree
   (c) Set of tokens
   (d) Strings of character

2. Syntax directed translation scheme is desirable because :
   (a) It is based on the syntax
   (b) Its description is independent of any implementation
   (c) It is easy to modify
   (d) All of the above

3. A top down parser generates :
   (a) Rightmost derivation
   (b) Rightmost derivation in reverse
   (c) Leftmost derivation
   (d) Leftmost derivation in reverse

4. Macro-processors are ..... .
   (a) Hardware
   (b) Compiler
   (c) Registers
   (d) None of the above

5. The term environment in programming language semantics is said as :

   (a) Function that maps a name to value held there

   (b) Function that maps a name to storage location

   (c) The function that maps a storage location to the value held there

   (d) None of the above

6. The lexical analyser takes ..... as input and produces a stream of ..... as output.

   (a) Source program, tokens

   (b) Token, source program

   (c) Either (a) and (b)

   (d) None of the above

7. A grammar is meaningless :

   (a) If terminal set and non-terminal set are not disjoint

   (b) If left hand side of a production is a single terminal

   (c) If left hand side of a production has no non-terminal

   (d) All of the above

8. The optimization technique which is typically applied on loops is :

   (a) Removal of invariant computation

   (b) Peephole optimization

   (c) Constant folding

   (d) All of the above

9. Resolution is externally defined symbols is performed by :
   (a) Linker
   (b) Loader
   (c) Compiler
   (d) Assembler

10. Which of the following is used for grouping of characters into tokens ?
   (a) Parser
   (b) Code optimization
   (c) Code generator
   (d) Lexical analyser