



NATIONAL SCHOOL OF BUSINESS MANAGEMENT

BSc (Hons) in Computer Science – 18.2/19.1

3rd Year 2nd Semester Examination

26 September 2021

CS305.3 – Programming Languages and Compiler Design

Instructions to Candidates

- 1) Answer All Questions
- 2) The duration of the question paper is 5 hours. Including half an hour to download the paper and upload your answers in a single file. (**Note: No email submissions are accepted under any condition.**)
- 3) Weightage of Examination: 60% out of final grade
- 4) Download the paper, provide answers to the selected questions in a word document.
- 5) Please upload the document with answers (Answer Script) to the submission link before the submission link expires
- 6) Answer script should be uploaded in PDF Format
- 7) Under any circumstances E-mail submissions would not be taken into consideration for marking. Incomplete attempt would be counted as a MISSED ATTEMPT.
- 8) The Naming convention of the answer script – Module Code_Subject name_Index No
- 9) You must adhere to the online examination guidelines when submitting the answer script to N-Learn.
- 10) Your answers will be subjected to Turnitin similarity check, hence, direct copying and pasting from internet sources, friend's answers etc. will be penalized.

Question 1 -20 Marks

1. Briefly explain the differences between the compiler and the interpreter. (3 marks)
2. What is the usage of the Symbol Table in compilation process? (3 marks)
3. Explain why the engineering effort required must be manageable in compiler code optimization. (3marks)
4. Briefly explain the tasks done by the lexical analyzer. (4 marks)
5. Why compiler designers separated lexical analysis from syntax analysis? (3 marks)
6. By using readability, writability, and cost metrices explain how we can evaluate a programming language. (4 marks)

Question 2 -20 Marks

1. Show that whether statement **begin a = b + c * d end** can be derived by using the below grammar (use BNF left most derivation). (5 marks)

<program> → begin <stmt> end
<stmt> → <var> = <expr>
<expr> → <expr> + <term> | <term>
<term> → <term> * <var> | <var>
<var> → a | b | c | d

2. Draw the parse tree for the above derivation. (4 marks)
3. Show that below grammar is ambiguous when deriving the statement, **a = b - c * d.** (8 marks)

<program> → <var> = <expr>
<expr> → <expr> - <expr> | <expr> * <expr> | <var>
<var> → a | b | c | d

4. How to resolve the ambiguity of the grammar in question 3? (3 marks)

Question 3 -20 Marks

Consider the below grammar and answer the questions accordingly. Note that non-terminal symbols are {S, A, B, C, D} and {s} is the start symbol. The terminal symbols are {a, b, g, h, r}.

S → AB
B → aAB | ε
A → DC
C → bDC | ε
D → g | hSr

- Find the FIRST set for the given grammar. (5 marks)
- Find the FOLLOW set for the given grammar. (5 marks)
- Create the LL(1) predictive parsing table. (5 marks)
- By using LL(1) parsing check that input string **gbgag** can be accepted or not. (5 marks)

Question 4 -20 Marks

Consider the below grammar and answer the questions accordingly. Note that non-terminal symbols are {E, F} and start symbol is {E}. The terminal symbols are {p, q}.

$$\begin{aligned} E &\rightarrow FF \\ F &\rightarrow pF \mid q \end{aligned}$$

- Write the augmented grammar. (2 marks)
- Find out the canonical collection of LR(0) items. (8 marks)
- Create the LR(0) parsing table. (5 marks)
- By using LR(0) parsing show that the input string **ppqq** can be accepted or not. (5 marks)

Question 5 -20 Marks

- Briefly explain how to use LEX tool to process another input after completing the processing of one input. (4 marks)
- Briefly explain the usage of semantic analyzer in compilation process. (4 marks)
- Consider below semantic rules. By using SDD (Syntax Directed Definition) show what is the data type of the expression **5+10+20**. (8 marks)

Production	Semantic Rules
$E \rightarrow E_1 + E_2$	{if((E ₁ .Type == E ₂ .Type) && E ₁ .Type = int) then E.Type = int else error}
$E \rightarrow E_1 == E_2$	{if((E ₁ .Type == E ₂ .Type) && (E ₁ .Type = int/bool)) then E.Type = bool else error}
$E \rightarrow (E_1)$	{E.Type = E ₁ .Type}
$E \rightarrow \text{num}$	{E.Type = int}
$E \rightarrow \text{True}$	{E.Type = bool}
$E \rightarrow \text{False}$	{E.Type = bool}

- Briefly explain why instruction selection is important when generating the target code? (4 marks)

..... **END OF THE PAPER**.....