FACULTY OF ENGINEERING, SCIENCES AND TECHNOLOGY

Department: Computer Science Program: BS

COMPILER CONSTRUCTION		
Announced Date: 08/11/25	Due Date: 29/11/25	Max Marks: 05

ASSIGNMENT # 01			
Mapped CLO	**Mapped GA**	**Mapped Learning Level**	**SDG**
CLO1	GA – 2 (Knowledge for Solving Computing Problems)	C2 (Understanding)	4 & 9

Question # 01 [01 marks]

Identify whether the following context-free grammars are ambiguous or unambiguous.

a) $S \rightarrow SS \mid S \mid \lambda$ $string = ()()()$

b) $S \rightarrow SbS \mid a$ $string = ababa$

c) $S \rightarrow aB \mid ab, A \rightarrow AB \mid a, B \rightarrow ABb \mid b$ $string = ab$

d) $S \rightarrow aSbS \mid bSaS \mid \lambda$ $string = abab$

e) $S \rightarrow aAB, A \rightarrow bBb, B \rightarrow A \mid \lambda$ $string = abbbb$

Question # 02 [01 marks]

Identify whether the following context-free grammars contain left recursion, and remove it if present.

a) $A \rightarrow ABd \mid Aa \mid a, B \rightarrow Be \mid b$

b) $E \rightarrow E + T \mid T,\ T \rightarrow T * F \mid F,\ F \rightarrow id$

c) $A \rightarrow Ba \mid Aa \mid c,\ B \rightarrow Bb \mid Ab \mid d$

d) $X \rightarrow XSb \mid Sa \mid b,\ S \rightarrow Sb \mid Xa \mid a$

e) $S \rightarrow Aa \mid b,\ A \rightarrow Ac \mid Sd \mid \lambda$

Question # 03 [01 marks]

Explain how left factoring applies to the following context free grammars.

f) $S \rightarrow iEtS \mid iEtSeS \mid a, E \rightarrow b$

g) $A \rightarrow aAB \mid aBc \mid aAc$

h) $S \rightarrow aS' \mid b, S' \rightarrow SSbS \mid SaSb \mid bb$

i) $S \rightarrow a \mid ab \mid abc \mid abcd$

j) $S \rightarrow aAd \mid aB, A \rightarrow a \mid ab, B \rightarrow ccd \mid ddc$

Question # 04 [02 marks]

Explain how a recursive descent parser would handle each nonterminal (E, T, F) of the given left-recursive grammar, identify the recursive functions for each nonterminal and their roles in parsing expressions, illustrate the conceptual parsing of the input $id + id * id$, discuss how the parser detects and handles invalid input (such as a missing parenthesis), and explain how left recursion affects parsing and how the grammar can be transformed to support recursive descent parsing.

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid id$$