Q1: Define the below terms with examples: *Lexemes, Token, Reserved Words, Metalanguage, Context Free Grammar, Derivation, and Production Rule*.

Q2: Explain the advantages of EBNF over BNF grammar; use examples as needed. Tale an example BNF grammar of your choice and transform the BNF grammar to the EBNF format.

Q3: Consider the below grammar and identify if the strings *babb*, *bbbabb*, *bbaaaaabc*, *aaaaaa* are derivable. Show your derivation. The grammar goes as follows:

$$< S > \longrightarrow < A > a < B > b$$

$$< A > \longrightarrow < A > b \mid b$$

$$< B > \longrightarrow b$$

Q4: Write a grammar for the language that includes all strings containing *n* copies of alphabet *a*, followed by *n+1* copies of the alphabet *b*. Assume $n > 0$. Upon completion of the grammar definition, draw the parse tree for the sentences *abb* and *aabbb* along with the derivations.

Q5: Consider the identifier definition of a hypothetical programming as follows: strings of consist of letters and numbers, and they must begin with a letter.

Q6: Explain the concepts of ambiguity of a grammar. Take an example of your choice and show that the grammar is ambiguous. Draw the parse tree as needed.

Q7: Select three flow control statements from C-programming, or from any other programming of your choice, and explain how the relevant grammars are written.