NATIONAL OPEN UNIVERSITY OF NIGERIA
PLOT 91, CADASTRAL ZONE, NNAMDI AZIKIWE EXPRESSWAY, JABI-ABUJA
FACULTY OF COMPUTING
DEPARTMENT OF COMPUTER SCIENCE
2025_1 EXAMINATION

Course Code: CIT316
Course Title: Principles and Techniques of Compilers (Compiler Construction I)
Course Credit: 3 units
Time Allowed: 3 hours
Instruction: Answer **Question One (1)** and any other **Three (3)** Questions

Question 1: 25 Marks (Compulsory Question)

1 (a) Define formal grammar. **(5 marks)**
1 (b) Explain the concept of a start symbol in formal grammars. **(3 marks)**
1 (c) Describe the syntax of formal grammars, focusing on the components such as terminal and non-terminal symbols. **(7 marks)**
1 (d) Provide an example of a simple formal grammar and explain how it generates strings. **(5 marks)**
1 (e) Differentiate between context-free grammars and regular grammars. **(5 marks)**
1 (f) Discuss the relevance of formal grammars in programming languages. **(5 marks)**

Question 2: 15 Marks

2 (a) Describe any three (3) phases of a compiler. **(6 marks)**
2 (b) Briefly describe top-down parsing and its main advantage. **(3 marks)**
2 (c) What is the difference between LL and LR parsing? **(3 marks)**
2 (d) Explain the concept of lookahead in parsing and its importance. **(3 marks)**

Question 3: 15 Marks

3 (a) Define a Context-Free Grammar (CFG). **(3 marks)**
3 (b) Explain the role of the parser in a compiler. **(3 marks)**
3 (c) Discuss the concept of ambiguity in CFGs and provide an example. **(4 marks)**
3 (d) What is left recursion in CFGs, and why must it be eliminated in parsers? **(5 marks)**

Question 4: 15 Marks

4 (a) Define ambiguity in the context of CFGs. (2 marks)
4 (b) Explain why ambiguity is problematic in programming languages. (3 marks)
4 (c) How can operator precedence and associativity be used to remove ambiguity in CFGs? (6 marks)
4 (d) Discuss how parsers deal with ambiguity when it is unavoidable. (4 marks)

Question 5: 15 Marks

5 (a) Define and differentiate between compile-time errors and run-time errors in the context of compiler design. (5 marks)
5 (b) Explain the role of lexical analysis in error detection during compilation. (4 marks)
5 (c) Describe how Integrated Development Environments (IDEs) have changed the approach to handling compilation errors. (3 marks)
5 (d) Discuss the historical challenges of error recovery in compilers, especially in batch processing environments. (3 marks)

Question 6: 15 Marks

6 (a) Define a symbol table and explain its purpose in a compiler. (4 marks)
6 (b) Illustrate the difference between global and local scope with examples in terms of symbol table entries. (5 marks)
6 (c) Describe how a compiler handles the redeclaration of a variable within an inner scope, and the role of the symbol table in this process. (4 marks)
6 (d) Explain how a symbol table can be implemented using a hash table and the advantages of this approach. (2 marks)