

**REGULAR UNIVERSITY EXAMINATIONS**

**2018/2019 ACADEMIC YEAR**

**SCHOOL OF SCIENCE AND INFORMATION SCIENCES**

**THIRD YEAR SECOND SEMESTER EXAMINATION FOR THE DEGREE OF  
BACHELOR OF SCIENCE IN COMPUTER SCIENCE**

**COURSE CODE: COM 3204**

**COURSE TITLE: COMPILER CONSTRUCTION**

### QUESTION ONE [30 MARKS]

- a. Use examples of your own to describe the following key terms as used in compiler construction
- i. Parse
  - ii. Sentence
  - iii. Lexeme
  - iv. Syntax
- [8 Marks]
- b. For every program, the compiler uses a data structure to record 'names' and descriptions for those names. Identify this data structure and discuss its importance in the compilation processes particularly in relations to semantic analysis.
- [4 Marks]
- c. A program that performs syntax analysis is called a parser. *Discuss using an example.*
- [4 Marks]
- d. A compiler converts the high level instruction into machine language while an interpreter converts the high level instruction into an intermediate form. Qualify this statement
- [6 Marks]
- e. Discuss four major phases associated with the execution of a program on a computer system.
- [8 Marks]

### QUESTION TWO [20 MARKS]

- a. Given the following program sentence: "***The humanoid robot raised the black microchip to the massive motherboard***"
- i. Build up your own grammar from this statement and
  - ii. Construct a symbol table to represent your grammar
  - iii. Using the Bottom Up procedure, show how the parsing of the sentence would be done.
  - iv. Finally, build a parse tree from the sentence.

[12 Marks]

- b. It is usually possible to combine scanning and parsing in the program implementation, however where possible it is advisable to separate the two activities. Properly explain the reasons why this is necessary.

[8 Marks]

**QUESTION THREE [20 MARKS]**

- a) The design of a compiler involves several stages; state and discuss each stage in detail

[10 Marks]

- b) Briefly describe the significance of a Syntax Directed translation.

[4 Marks]

- c) Top-down parsers are implemented as a set of recursive functions that descent through a parse tree for a string. Explain recursive descent passing phenomena in compiler construction

[6 Marks]

**QUESTION FOUR [20 MARKS]**

a)

- i. Briefly explain how use of disambiguating rules such as precedence help in resolving ambiguity in grammar.

[4 Marks]

- ii. An expression like  $4*3 + 8$  can generate more than one parse tree. Construct any two parse trees for the expression.

[6 Marks]

- iii. Comment on the implications of this scenario during semantic analysis

[2 Marks]

- b) State and describe any four types of grammar used in the design of Compilers.

[8 Marks]

**//END**