Department of Higher Education
University of Computer Studies, Yangon
Third Year (B.C.Sc.)
Final Examination
Advanced Programming Techniques (CS-306)
October, 2018

Answer ALL questions. **Time Allowed:3 hours.**

1. Answer **ANY FIVE** of the followings. **(20 marks)**

(a) What is the disadvantage of having too many features in language?
(b) What is the primary use of attribute grammar?
(c) Describe three advantages of LR parsers.
(d) What is the general problem with static scoping?
(e) Define row major order and column major order.
(f) Define explicit heap-dynamic and implicit heap-dynamic variables.
(g) What is one example of a lack of orthogonality in the design of C?

2. Briefly discuss **ANY FOUR** of the following programming concepts. **(28 marks)**

(a) What are three general methods of implementing a programming language? Describe pure interpretation and Hybrid implementation systems.
(b) Distinguish between static and dynamic semantics.
(c) Describe the parsing problem for a top-down parser and a bottom-up parser.
(d) What are the advantages and disadvantages of dynamic scoping?
(e) What are the design issues for arrays?

3. (a) Using the following grammar; **(20 marks)**

<assign> → <id> = <expr>
<id> → A | B | C
<expr> → <id> + <expr>
| <id> * <expr>
| (<expr>)
| <id>

Show a **parse tree** anda **leftmost derivation** for each of the following statements:
(i) B= C* (A*C+B) (ii) A=A*(B+(C*A))

(b) Consider the traditional grammar for arithmetic expressions the follows:

1. E → E + T
2. E → T
3. T → T * F
4. T → F
5. F → (E)
6. F → id

(i) Draw the **LR parsing table**.
(ii) Show a complete parse, including the parse **stack** contents, **input string** and **action** for the string **(id + id) * id** using the grammar and parse table.

4. (a) Consider the following **EBNF** description of simple arithmetic expressions:

(20 marks)

<expr> → <term> {(+ | -) <term>}
<term> → <factor> {(* | /) <factor>}
<factor> → id | int_constant | (<exp>)

Show a **parse tree** and a **trace of the recursive descent parser** for the string **x * (y + z).**

(b) Consider the following C program:

```
void  fun (void) {
int   x, y, z ;   / * definition 1*/
      ----------
while (------) {
      int  y, z, a;   / * definition 2*/
           ----------  1 <----------
   while(--------) {
      int z, a, b;   / * definition 3*/
           ----------  2 <----------
    }
           ---------- 3 <----------
  }
           ---------- 4 <----------
}
```

For each of the **four marked points** in this function, **list each visible variable**, along with the number of the definition statement that defines it.

5. Answer the followings: **(12 marks)**

(a) Which produces faster program execution, a compiler or a pure interpreter?
(b) Convert the following **EBNF to BNF**.

A → a[b] A

(c) What is the shift action?
(d) What is a reserved word?
(e) Why C and C++ are not strongly typed?
(f) What are the two common problems with pointers?

****