922014 [illegible]

85

Midterm Exam for Programming Language
by Kun-Yuan Hsieh

A. (60%) Read the following sentence. Mark O if true, X if false.

~~O~~ 1. Scope of a variable is the range which it is visible. Lifetime is the time during which it is bound to a specific memory location. They are closely related and ~~the~~ same concept.

X 2. BNF is not context-free.

O 3. Static binding means the binding occurs before run time and remains unchanged in execution

~~O~~ 4. If two variables have the same type name are compatible, we call it name type compatibility, such as C.

O 5. In the following grammar:
<exp> → <exp> + <id> | <id>, the operator + performs a left-to-right associativity.

~~O~~ 6. Compilation translates the high-level program to machine code while interpretation translates it to an intermediate code.

O 7. We call a pointer dangling if it points to a heap-dynamic variable that has been deallocated.

O 8. Operator generated in the lower-level in the parse tree has precedence over the higher-level ones.

X 9. The derivation of "A = B + C * A " bellow is a left-most derivation which is according to an unambiguous grammar.

```
<assign>
    → <id> = <exp>
    → A = <exp>
    → A = <exp> + <exp>
    → A = <id> + <exp>
    → A = B + <exp>
    → A = B + <exp> * <exp>
    → A = B + <id> * <exp>
    → A = B + C * <exp>
    → A = B + C * <id>
    → A = B + C * A
```

O 10. Bottom-up parsing is often called *shift-reduce algorithms* because it move the next token to the parser stack and replacing the RHS on the top of the stack with corresponding LHS.

[illegible]

85

Midterm Exam for Programming Language
by Kun-Yuan Hsieh

A. **(60%) Read the following sentence. Mark O if true, X if false.**

X 1. Scope of a variable is the range which it is visible. Lifetime is the time during which it is bound to a specific memory location. They are closely related and the same concept.

X 2. BNF is not context-free.

O 3. Static binding means the binding occurs before run time and remains unchanged in execution

X 4. If two variables have the same type name are compatible, we call it name type compatibility, such as C.

O 5. In the following grammar:
<exp> → <exp> + <id> | <id>, the operator + performs a left-to-right associativity.

X 6. Compilation translates the high-level program to machine code while interpretation translates it to an intermediate code.

O 7. We call a pointer dangling if it points to a heap-dynamic variable that has been deallocated.

O 8. Operator generated in the lower-level in the parse tree has precedence over the higher-level ones.

X 9. The derivation of "A = B + C * A " bellow is a left-most derivation which is according to an unambiguous grammar.

```
<assign>
    → <id> = <exp>
    → A = <exp>
    → A = <exp> + <exp>
    → A = <id> + <exp>
    → A = B + <exp>
    → A = B + <exp> * <exp>
    → A = B + <id> * <exp>
    → A = B + C * <exp>
    → A = B + C * <id>
    → A = B + C * A
```

O 10. Bottom-up parsing is often called *shift-reduce algorithms* because it move the next token to the parser stack and replacing the RHS on the top of the stack with corresponding LHS.

X 11. In the following code, variable ***i*** in test() function is a *stack-dynamic variable*, while ***i*** in main() is **NOT** because the storage of variable ***i*** in main() is bound during run time and remains unchanged in execution.

```
#include <stdio.h>
int count;
main( ) {
    int i ;
    for (i=0; i<=10; i++){
            test( ) ;
      }
}
test( ) {
    int  i ;
    static int count = 0;
    count = count + 1 ;
}
```

$7 \times 5000 + 5 + 100 + 3$

a_{000} a_{001} $a_{00\cdot}$ A[1][0] A[1][1] A[1][2]

A[2][3][5]

A[0][0] A[0][1] A[0][2]

O 12. Garbage collection invalidates all memory cells first. Second traces all the pointers to mark valid ones. After, all cells that have not been marked as valid are returned to the available space.

B. (40%) Read select following sentence and select a correct answer.

a 1. If a three-dimension array A[100][50][20] is implemented by column-major allocation. &A[0][0][0] = = 1200.& A[3][5][7]

a. 36703

b. 36507

c. 3107

d. 3108

$20 \times 500 + 7 \times 5000 + 5 \times 100 + 3$

35000 2500 35500 1200 36700

a_{211} a_{212} a_{213} a_{11} a_{12} a_{13}

$1 \times 9 + 2 \times 3 + 1$

a[3][3][2]

C 2. –a) If static scope, what's X2 after execution?

a. 5

b. 10

c. 15

d. NULL

b 2. –b) If dynamic scope, what's X2 after execution?

a. 5

b. 10

c. 15

d. NULL

```
main( ){
        int X=10;
        int X2;
        A( ){
                return X;
        }
        B( ){
                int X = 5;
                return A() + X;
        }
        X2 = B();
}
```

d 3. Which of the following characteristic **DOES NOT** affect a programming language's readability and writability?
   a. Operation overloading
   b. Orthogonality
   c. Sub program
   d. Reliability

c 4. Variables that are allocated and deallocated by some statements during run-time, they are:
   a. static variables
   b. stack-dynamic variables
   c. explicit heap-dynamic variables
   d. implicit heap-dynamic variables

d 5. Which of the following grammar ~~allows~~ top-down parsing?
   a. $A \rightarrow B\,a\,A;\ B \rightarrow bA\,b$ disallows
   b. $A \rightarrow a\,A\,B;\ B \rightarrow A\,b$
   c. $A \rightarrow B\,a\,A\,;\ B \rightarrow c$
   d. $A \rightarrow A\,B;\ B \rightarrow b$

C 6. Which of the following statements about syntax analysis is **NOT** true?
   a. Syntax analysis is broken into two distinct parts: lexical analysis and parsing
   b. Lexical analysis extracts lexemes and produce the corresponding tokens
   c. Using context-free grammar, we can perform type-checking during parsing.
   d. The hierarchical syntactic structure the grammar described is called parse tree, which is the output of a parser.

C 7. Which of the following data type is **NOT** usually being concerned as primitive?
   a. Integer
   b. Character
   c. Pointer
   d. Boolean