

Login name \_\_\_\_\_

# Quiz 1

Name \_\_\_\_\_

## CSE 131

Signature \_\_\_\_\_

Spring 2010

Student ID \_\_\_\_\_

### Compilation/Compiler Overview, Names/Scopes/Bindings

1. Give the order of the typical C compilation stages and on to actual execution as discussed in class

- |   |                            |
|---|----------------------------|
| 0 – Loader  | 6 – ccomp (C compiler)     |
| 1 – Program Execution   | 7 – ld (Linkage Editor)    |
| 2 – as (Assembler)  | 8 – Source file (prog.c)   |
| 3 – Object file (prog.o)                                      | 9 – Assembly file (prog.s) |
| 4 – prog.exe/a.out (Executable image)                         | 10 – cpp (C preprocessor)  |
| 5 – Segmentation Fault (Core Dump) / General Protection Fault |                            |

gcc \_\_\_\_\_ -> \_\_\_\_\_ -> \_\_\_\_\_ -> \_\_\_\_\_ -> \_\_\_\_\_ -> \_\_\_\_\_ -> \_\_\_\_\_ -> \_\_\_\_\_ -> \_\_\_\_\_ -> \_\_\_\_\_ -> \_\_\_\_\_

Given the following CUP grammar snippet (assuming all other Lexing and terminals are correct):

```

Stmt ::=      Expr T_SEMI
           {: System.out.println("____"); :}
        ;

Expr ::=      Expr1 AssignOp {: System.out.println("____"); :} Expr
           {: System.out.println("____"); :}
        |      Expr1 {: System.out.println("____"); :}
        ;

Expr1 ::=     T_ID {: System.out.println("____"); :}
        ;

AssignOp ::=  T_ASSIGN {: System.out.println("____"); :}
        ;

```

Fill in the blanks in the above action code to match the output below when parsing the follow statement:

a = b = c;

<u>Output</u>
A
=
2
A
=
2
A
B
3
3
1

What is the associativity of AssignOp in the above grammar?

\_\_\_\_\_

2. Give the order of the phases of compilation in a typical C compiler as discussed in class

- A – Source language file (for example, prog.c)
- C – Parser (Semantic Analysis)
- E – Target language file (for ex., prog.s)
- G – Code generation (for ex., Assembly)

- B – Intermediate Representation(s)
- D – Scanner (Lexical Analysis)
- F – Parser (Syntax Analysis)

\_\_\_\_\_ -> \_\_\_\_\_ -> \_\_\_\_\_ -> \_\_\_\_\_ -> \_\_\_\_\_ -> \_\_\_\_\_ -> \_\_\_\_\_

What is the name of the file in the compiler starterCode that defines the grammar for this quarter's language?

\_\_\_\_\_

Text \_\_\_\_\_ Data \_\_\_\_\_ BSS \_\_\_\_\_ Heap \_\_\_\_\_ Stack \_\_\_\_\_ My grade on this quiz \_\_\_\_\_

- 1) Sized dynamically at run time (size not known at compile time).
- 2) Fixed size known at compile time and thus part of the resulting executable file.

What is the name of the main data structure used in the compiler to store/retrieve information about names (variable names, function names, etc.) and other information/attributes (for example, types) associated with these names?

\_\_\_\_\_

Describe two common ways a program can contain a dangling pointer/reference error:

1)

2)

Check #1: For the T\_EQU and T\_NEQ operators, the operand types must be either BOTH \_\_\_\_\_, or BOTH equivalent to \_\_\_\_\_, and the resulting type is \_\_\_\_\_.