Login name _____ **Quiz 1** **Name** _____

CSE 131B

Signature _____ **Winter 2006** **Student ID** _____

Compilation/Compiler Overview, Names/Scopes/Bindings

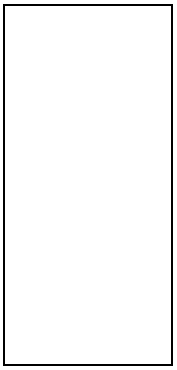
1. Briefly explain the difference between a translator and an interpreter.

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

```
Expr1 ::=           Expr3 T_ASSIGN Expr2 T_SEMI
                    {:
                        System.out.println("F");
                    :}
    ;

Expr2 ::=           Expr2 AddOp Expr3
                    {:
                        System.out.println("G");
                    :}
    |               Expr3
                    {:
                        System.out.println("H");
                    :}
    ;

Expr3 ::=           T_ID
                    {:
                        System.out.println("I");
                    :}
    ;

AddOp ::=           T_PLUS
                    {:
                        System.out.println("J");
                    :}
    ;
```

What is the output on the screen when the follow statement is given as input:

```
x := x + x;
```



(over)

3. We discussed the concepts of the lifetime of an object and the lifetime of a name-to-object binding. Match the most appropriate phrase with the following. Yes, you will need to use some letters more than one time. Some terms may have more than one appropriate phrase. To hopefully make this less complex, do not consider name overloading, using the same name in an inner scope, or recursion in answering this question.

Dangling reference	____	A) Lifetime of object and lifetime of name-object binding are the same
External static var	____ ____	B) More than one name bound to a single object
Global var	____	C) Lifetime of object lasts longer than lifetime of name-object binding
Alias	____	D) Object and name-object binding is live, but var./name not in scope
Local var	____ ____	E) Lifetime of name-object binding lasts longer than lifetime of object
Formal parameter	____ ____	
Internal static var	____ ____	
Memory leak	____	

4. What parts of the C run time environment are sized dynamically at run time (not known at compile time)?

What parts are known fixed size at the end of the compilation sequence?

5. In the CUP grammar, expressions with lower precedence are _____ in the grammar / parse.

6. The following production is in the CUP grammar:

```
Expr2 ::= Expr2 AddOp Expr3
```

along with the proper rules for Expr, Expr1, Expr3 through Expr5, Factor, etc.

If we replaced the above production for Expr2 with

```
Expr2 ::= Expr3 AddOp Expr2        /* Exchange Expr2 and Expr3 on the right hand side */
```

in essence this changes the _____ of the AddOp operators.

Draw the expression tree (just use 5 nodes) for the expression a + b + c using this new rule. This will hopefully help you answer the question.

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

A – Machine-independent code improvement (optional)	B – Code generation (for ex., assembly)
C – Source language file (for example, C)	D – Intermediate Representation
E – Machine-specific code improvement (optional)	F – Scanner (lexical analysis)
G – Parser (Semantic analysis)	H – Parser (syntax analysis)
I – Target language file (for ex., assembly)	

_____ –> _____ –> _____ –> _____ –> _____ –> _____ –> _____ –> _____ –> _____