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

CSE 131B

Signature _____ **Spring 2005** **Student ID _____**

Compilation/Compiler Overview, Names/Scopes/Bindings

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

A – Program Execution
B – ccomp (C compiler)
C – Source file
D – exe/a.out (executable image)
E – as (assember)
F – ld (Linkage Editor)
G – cpp (C preprocessor)
H – loader

gcc _____ –> _____ –> _____ –> _____ –> _____ –> _____ –> _____ –> _____

2. Add the following production rule from Phase 0 to the oberon.cup file using the appropriate CUP grammar syntax and add action code such that the result of this production is the value of the symbol `Designator`. Assume the symbol `NewStmt` has been defined appropriately in the Symbol List toward the top of the CUP file.

```
NewStmt -> T_NEW T_LPAREN Designator T_RPAREN
```

Which of the symbols in this grammar rule are non-terminal symbols?

Which of the symbols in this grammar rule are terminal symbols?

When this rule has been fully recognized by the parser, does this represent a shift or reduce?

(over)

3. Briefly explain the difference between syntax analysis and semantic analysis.

4. A dangling reference is a term describing a condition in which a name (typically a pointer variable) is still bound to an object (memory location) but the object is no longer alive (the memory location allocated for that object has been deallocated and is no longer associated with that object). Describe two conditions where this can occur. Be specific.

1)

2)

5. The C programming language specifies any object allocated in the Data segment must be initialized with a compile time determined value. The C++ programming language allows run time determined values (such as function call return values) to be used to initialize Data segment objects. Global and external static variables are initialized before main() is called. Internal static variables are initialized the first time encountered in the function in which they are defined. Consider the following C++ program. Part of the output is supplied. Fill in the missing values that get printed on the lines in the output box.

```
#include <iostream>
using namespace std;

int a = 0;

int foo( char *s ) { cout << "In foo() initializing " << s << endl; return ++a; }

int z = foo( "z" );

static int x = foo( "x" );

int
main()
{
  cout << "In main()" << endl;

  cout << "a = " << a << endl;

  static int y = foo( "y" );

  cout << "x = " << x << endl;
  cout << "y = " << y << endl;
  cout << "z = " << z << endl;

  return 0;
}
```

Output

```
In foo() initializing z
In foo() initializing x
In main()
a = ___
In foo() initializing y
x = ___
y = ___
z = ___
```