



Semantic Analysis

Find 6 problems with this code.
These issues go beyond syntax.

```
fie(a,b,c,d) {  
    int a, b, c, d;  
    ...  
}  
  
fee() {  
    int f[3],g[0], h, i, j, k;  
    char *p;  
  
    fie(h,i,"ab",j, k);  
    k = f * i + j;  
    h = g[17];  
    printf("<%s,%s>.\n",p,q);  
    p = 10;  
}
```

What kinds of questions does the semantic analysis/code generator need to answer?

Semantic Analysis = Values/Meaning

Context-Sensitive Analysis

How can we answer these questions?

- Use formal methods
 - Context-sensitive grammars?
 - Attribute grammars?
- Use *ad-hoc* techniques
 - Symbol tables
 - *Ad-hoc* code

(attributed grammars?)

(action routines)

In parsing, formalism won; here, ad-hoc techniques dominate actual practice

SCOPING: Consider the following code segment in a language with nested function definitions in which all parameters are passed by reference.

```
• int a, b, c, d;
• function f(int a, int x)
• { int b = 0;
•   function g()
•   { int c = 3;
•     print ("in g:",a,b,c,d,x);
•     call h(a,b,c);
•     print("in g:",a,b,c,d,x);
•   }
•   /*in f*/
•   print("in f:",a,b,c,d,x);
•   call g();
•   print("in f:",a,b,c,d,x);
•   a = 6;
• }
• function h(int x, int y, int z)
• { int d = 2;
•   print("in h:",a,b,c,d,x,y,z);
•   x = 3; y= 4; z = 5;
• }
• function main()
• { a = b = c = d = 99;
•   call f(b,c); print("in main",a,b,c,d); call h(a,b,c); print("in main", a,b,c,d);
• }
```

1. Show the output of this program assuming static scoping.
2. Show the output of this program assuming dynamic scoping.