Homework 4
posted on Feb. 24, due on Mar. 8 at the beginning of the class (no late turn-ins will be accepted).
Please put a cover page over the homework. The grade will be written on the second page.

1. (30 pts) Exercise 3.11.

2. (30 pts) Given the following abstract grammar

   E-> E+E | E-E | E*E | id | num

   a. (5 pts) Design a concrete grammar for LL parsing.
   b. (10 pts) What is the AST for the input of "x-y*10"
   c. (5 pts) Write classes that represent ASTs.
   d. (10 pts) Use the visitor pattern to write an evaluator that evaluates the value of an expression. Write the classes on the paper.

3. (20 pts) Exercise 6.1 b and c.

4. (10 pts) Given the following program

```
int f (int i) {

    int j,k;
     j=i*i;
     k=i? f(i-1):0;
    return k+j;

}

void main ( ) {  f(100000); }
```

   a. Imagine a compiler that passes parameters in registers, wastes no space providing backup storage for parameters in registers, does not put local variables in registers, and in general makes stack frames as small as possible. How big should each stack frame for *f* be, in words? Please explain the reason.
   b. What is the maximum memory use of this program, with such a compiler?

5. (10 pts) Given the following program

```
void foo(char * s) {
   int i;
   char c[4];
   i=0;
   for (i=0;i<strlen(s);i++) c[i]=s[i];
   printf("i=%x\n",i);
}

int main () {
   foo ("aaaaaaaa");
}
```

Assuming the compiler places variables next to each other in their declaration order in the active record, i.e. *i* is at a higher address than *c*. What is the output of this program? Why? Note that we also assume the compiler does not deploy stack-smash protection.