CS 352 Midterm Two (3/12/09)

name___________________

1. Given the following grammar:

   S -> AB

   A -> cAa | +

   B -> Bb | b

   (1). Present the rightmost derivation for string c+abb.

   S->AB->ABb->Abb->cAabb->c+abb

   (2). What is the handle of ccc+aaabb? You can simply underline (circle) the handle in the string.

   + is the handler

   (3). Present the shift-reduce parsing of the string cc+aabb, by filling the following table.

Stack	Input	Action
$	cc+aabb$	Shift
$c	c+aabb$	Shift
$cc	+aabb$	Shift
$cc+	aabb$	reduce
$ccA	aabb$	Shift
$ccAa	abb$	reduce
$cA	abb$	Shift
$cAa	bb$	reduce
$A	bb$	Shift
$Ab	b$	reduce
$AB	b$	Shift
$ABb	$	reduce
$AB	$	reduce
$S	$	accept

2. Translate the following to MiniJava IR, you don’t need to use Cx, Nx, Ex, unEx(), unNx(), or unEx().

   (1) A[3]=3 (A is local variable and its offset on the stack frame is f)

```
MOVE (
  MEM(
     BINOP(+, BINOP(+, TEMP(FP), CONSTANT(f),
              BINOP(*, 3, w))
        ),
 CONST(3)
)
```

   (2) if (c=3) a=1 (a and c are global variables, their addresses are &a and &c, respectively).

```
CJUMP(
 ESEQ(
   MOVE (MEM(CONST(&c)), CONST(3)),
   MEM(CONST(&c))
  ),
  CONST(1),
  t,
  f);
LABEL(t);
MOVE (MEM(CONST(&a)), CONST(1));
LABEL(f);
```

   Many students lost points in not handling the assignment correctly.

3. Given the following program

```
char ss[8]= "aabbccdd";
void main () {
    int i;
    f();
    h(ss);
}
```

```
void f () {
    static int i;
    int j;
    g();
}
```

```
void g () {
    int i;
    char * A;
    A=(char*) malloc(1000);
 L: ...
}
```

```
void h (char* s) {
    int i;
    int k=0;
    char B[4];
    int j,l;
    for (i=0;i<strlen(s);i++) {
        B[-i]=s[i];
    printf("%x, %x\n", k, j);
}
```

(1). What does the stack look like when the execution is at label L in function g. Assume the compiler does not use registers for parameter passing, return values, or locals.

1. i
2. ret_addr
3. old_fp
4. j
5. ret_addr
6. old_fp
7. i
8. A (4 bytes)

1,2,3 comprise the main's AR. 4,5,6 comprise f's AR, and 7 and 8 comprise g's AR.

i in f() is a global and the content of A is on heap although the pointer variable is on the stack.

(2). What is the maximum stack memory usage (to be safe, please show the break down in details)? What is the context corresponding to the maximum usage.

9 words, 4 for main's record, 5 for h's record.

(3). Inside function h(), the index of array B is "-i", what is the output of the printf? Why? Please state your assumption about the variable layout inside the stack frame. The hexical ascii codes for a, b, c, and d are 61, 62, 63, and 64.

0, 61626263

4. AST.

(1). Briefly explain the differences between abstract grammars and concrete grammars?

Abstract grammars are programmer-friendly, usually have left-recursion and ambiguity. Concrete grammars are parser friendly.

(2). Transform the following for loop to a while loop.

```
for (i=0;i<n;i++) sum=sum+i;
```

(3). Come up with grammar rules for for loop and while loop statements and define classes for these statements. Please show field definitions, you don't need to show method definitions. You can assume the existence of base classes such as Stmt and Exp.

```
ForStmt -> for (Stmt;  Exp; Stmt) Stmt
WhileStmt -> while (Expr) Stmt
```

(4). Write a visitor pattern that translates a for statement into a while statement, meaning given the AST of a for statement, your method returns an AST of a while statement that is equivalent. You only need to write the visit function for the for statement. Make sure you explain the return value of your visit function and you handle the return values (from accept() functions) properly.

```
Stmt visit(ForStatement s) {
   CompStmt cs;
   Stmt s1= s.init.accept(this);
   if (s1==null) s1=s.init;
   Stmt s2=s.update.accept(this);
   if (s2==null) s2=s.update;
   Stmt s3=s.body.accept(this);
   if (s3==null) s3=s.body;
   cs= new CompStmt(s3, s2);
   WhileStmt ws= new WhileStmt(s.condition, cs);
   return new CompStmt(s1, ws);
}
```

The calls to accept() are needed to handle nested loops.