Parameterless Functions

```
int i;
int k;
int y;
int f1() {
      return k*(y+3);
}
k= 10;
y = 20;
print f1();
i = f1()*2;
print i;
```

Parameterless Functions

- Add a production for function declarations
- Add a production for void function calls (a new kind of Stmt)
- Add a production for a return statement (a new kind of Stmt)
- Add a production so function calls can be part of expressions

Parameterless Function

```
int i;
int k;
int y;
int f1() {
        return k*(y+3);
}
k= 10;
y = 20;
print f1();
i = f1()*2;
print i;
```

```
f1:
        addi    $sp, -8
        sw      $ra, ($sp)
        lw      $t0, k
        lw      $t1, y
        li      $t2, 3
        add     $t3, $t1, $t2
        mul     $t1, $t0, $t3
        sw      $t1, 4($sp)
        lw      $ra, ($sp)
        addi    $sp, 8
        jr      $ra
```

Parameterless Function Call
i = f1()*2;

```
jal    f1
lw     $t0, -4($sp)
li     $t1, 2
mul    $t2, $t0, $t1
sw     $t2, i
```

ParameterLess Function

- The code on slide 1 included the statement i=f1() * 2;
- What if the statement was i=2*f1()?;

Parameterless Functions
Is the following code correct?

```
li          $t0, 2
jal         f1
lw          $t1, -4($sp)
mul         $t2, $t0, $t1
sw          $t2, i
```

Store Temp Reg Values on the Stack

```
li      $t0, 2
addi    $sp, $sp, -4
sw      $t0, ($sp)
jal     f1
lw      $t0, ($sp)
lw      $t1, -4($sp)
addi    $sp, $sp, 4
mul     $t2, $t0, $t1
sw      $t2, i
```

Parameterless Functions

- See SaveSeq and RestoreSeq in CodeGen