```
class System {
  // No fields, since System_0 had ".long 4"
  static void out(int x);
}

class Fork {
  int a;
  Spoon b;

  // This appears to be a constructor: take in two args, create an object,
  // assign the args into the newly created object, then return the object.
  //static Fork ping(int a1, int a2) {
  static Fork ping(int a1, Spoon a2) {
    Fork f;
    f = new Fork();

    f.a = a1;
    f.b = a2;

    return f;
  }

  int bat(Fork f, Fork g) {
  //int bat(int a, int b) {
    this.a = this.a * 2;
     if (this.a > 8) {
       return 4;
     }

     // TODO: access these (they need to be on the stack)
     // In other words, system.out is not the correct call. But
     // at least it is accessing the correct arguments.
     System.out(f.a);
     //System.out(f.b);
     //b.a = f.a;
     //b.b = f.b;

     //TODO left off here, stuck in the L0 loop, emailed MPJ
     //b.b = 8;
     //b.a = 7;

    return 4;
  }
}

class Knife {
  // TODO unknown return value
  // TODO unknown arg value
  int bat(int a) {
    return a;
  }
}

class Spoon extends Knife {
  static int c;

  static void begin(int a) {
    // Sets the class static:
    // movl %eax,v_Spoon_c
    c = a;
  }

  static int retr() {
```

```
    return c;
  }

  // the first arg is needed, but isn't used.
  static Spoon pong(int a) {
    c = c+1;
    return new Spoon();
  }
}

class Main {
  // No fields: ".long 4"
  static void main() {
    Fork f;
    int a;
    Spoon.begin(4*5+3);
    f = Fork.ping(3, Spoon.pong(2));
    // TODO: another Fork.ping goes here. (What args?)
    //  the second ping() should be assigned to f, not the first.

    a = 0;
    while (a < 6) {
      // TODO: reference items in this order: 2, 1, f, a
      // TODO: this subsection is incorrect.
      // system.out lines are just here to reference the correct vars.
      //System.out(f);
      f.a = f.a * f.a;

      // TODO: it seems like we need to call f.a() as a method,
      //   even though f.a is a field.

      // TODO: this section is correct, matches the bottom of the loop.
      System.out(f.a);
      a = a+1;
    }
    int b;
    System.out(Spoon.retr());

  }
}
```

```
        .file   "test.j"
        .globl  Main_main
System_0:
        .long   4
Fork_ping:
        pushl   %ebp
        movl    %esp,%ebp
        subl    $4,%esp
        pushl   $Fork_0
        call    new_object
        addl    $4,%esp
        movl    %eax,-4(%ebp)
        movl    12(%ebp),%eax
        movl    -4(%ebp),%ecx
        movl    %eax,8(%ecx)
        movl    8(%ebp),%eax
        movl    -4(%ebp),%ecx
        movl    %eax,4(%ecx)
        movl    -4(%ebp),%eax
        movl    %ebp,%esp
        popl    %ebp
        ret
Fork_bat:
        pushl   %ebp
        movl    %esp,%ebp
        movl    16(%ebp),%eax
        movl    8(%eax),%eax
        imull   $2,%eax
        movl    16(%ebp),%ecx
        movl    %eax,8(%ecx)
        movl    16(%ebp),%eax
        movl    8(%eax),%eax
        cmpl    $8,%eax
        jng     l0
        movl    $4,%eax
        movl    %ebp,%esp
        popl    %ebp
        ret
l0:
        movl    12(%ebp),%eax
        movl    8(%eax),%eax
        pushl   %eax
        call    System_out
        addl    $4,%esp
        movl    $4,%eax
        movl    %ebp,%esp
        popl    %ebp
        ret
Fork_0:
        .long   12
        .long   Fork_bat
Knife_bat:
        pushl   %ebp
        movl    %esp,%ebp
        movl    8(%ebp),%eax
        movl    %ebp,%esp
        popl    %ebp
        ret
Knife_0:
        .long   4
        .long   Knife_bat
        .comm   v_Spoon_c,4
Spoon_begin:
```

```
        pushl   %ebp
        movl    %esp,%ebp
        movl    8(%ebp),%eax
        movl    %eax,v_Spoon_c
        movl    %ebp,%esp
        popl    %ebp
        ret
Spoon_retr:
        pushl   %ebp
        movl    %esp,%ebp
        movl    v_Spoon_c,%eax
        movl    %ebp,%esp
        popl    %ebp
        ret
Spoon_pong:
        pushl   %ebp
        movl    %esp,%ebp
        movl    v_Spoon_c,%eax
        addl    $1,%eax
        movl    %eax,v_Spoon_c
        pushl   $Spoon_0
        call    new_object
        addl    $4,%esp
        movl    %ebp,%esp
        popl    %ebp
        ret
Spoon_0:
        .long   4
        .long   Knife_bat
Main_main:
        pushl   %ebp
        movl    %esp,%ebp
        subl    $12,%esp
        movl    $5,%eax
        imull   $4,%eax
        addl    $3,%eax
        pushl   %eax
        call    Spoon_begin
        addl    $4,%esp
        movl    $3,%eax
        pushl   %eax
        movl    $2,%eax
        pushl   %eax
        call    Spoon_pong
        addl    $4,%esp
        pushl   %eax
        call    Fork_ping
        addl    $8,%esp
        movl    %eax,-4(%ebp)
        movl    $0,%eax
        movl    %eax,-8(%ebp)
l1:
        movl    -8(%ebp),%eax
        cmpl    $6,%eax
        jnl     l2
        movl    -4(%ebp),%eax
        movl    8(%eax),%eax
        movl    -4(%ebp),%ecx
        movl    8(%ecx),%ecx
        imull   %ecx,%eax
        movl    -4(%ebp),%ecx
        movl    %eax,8(%ecx)
        movl    -4(%ebp),%eax
```

```
        movl    8(%eax),%eax
        pushl   %eax
        call    System_out
        addl    $4,%esp
        movl    -8(%ebp),%eax
        addl    $1,%eax
        movl    %eax,-8(%ebp)
        jmp     l1
l2:
        call    Spoon_retr
        pushl   %eax
        call    System_out
        addl    $4,%esp
        movl    %ebp,%esp
        popl    %ebp
        ret
Main_0:
        .long   4
```