SOLUTION NOTES

Compiler Construction 2003 Paper 5 Question 6 (AM)

[This is a question on “converting tree to stack code” and on object-oriented storage layout.]

(a) The code

```
iconst 1
iconst 2
invokestatic f
iconst 3
iconst 4
invokestatic f
invokestatic f
```

is obtained by post-order traversal of the tree for *e*.

(b)

```
f: iload 1
   istore 3
   ...
   iload 1
   iload 2
   iload 3
   imul
   iadd
   ireturn
```

The idea is that FP points to the arguments local variables while SP is the stack fringe. [picture]

(c)

```
    +--------------+
A:  | a1 | a2 | mp |
    +--------------+
      0    4    8    12
    +------------------------+
B:  | a1 | a2 | mp | b1 | b2 |
    +------------------------+
      0    4    8    12   16   20
```

This layout means that casts just work as the first 12 bytes of a B-object are just an A-object. Calls to `m()` must work via a data-member, here `mp` where objects created by `new A` have their `mp` field set to `A.m` whereas those of type `new B` have their field set to `B.m`.