Compiler Construction 2005 – Paper 5 Question 6 (AM)
Solution notes

This concerns section 6 and 7 of the notes.

(a) Bookwork—see the notes.

(b) The ones where `g()` is called to the left of the operator do not require `a` to be saved before the proc. call.

(i) `b = g(7) + a;`

```
        iconst 7
        invokestatic g
        iload a
        iadd
        istore b
;
        movl %eax,#7
        call g
        addl %eax,<offset-of-a>(%fp)
        movl <offset-of-b>(%fp), %eax
```

(ii) `b = a + g(7);`

```
        iload a
        iconst 7
        invokestatic g
        iadd
        istore b
;
        movl %eax,<offset-of-a>(%fp)
        pushl %eax
        movl %eax,#7
        call g
        popl %ebx
        addl %eax, %ebx
        movl <offset-of-b>(%fp), %eax
```

(iii) `b = (-g(7)) + a;`

```
        iconst 7
        invokestatic g
        iload a
        ineg
```

```
        iadd
        istore b
;
        movl %eax,#7
        call g
        neg  %eax
        addl %eax,<offset-of-a>(%fp)
        movl <offset-of-b>(%fp), %eax
```

(*iv*) `b = a - g(7);`

```
        iload a
        iconst 7
        invokestatic g
        isub
        istore b
;
        movl %eax,<offset-of-a>(%fp)
        pushl %eax
        movl %eax,#7
        call g
        popl %ebx
        subl %bax, %eax
        movl <offset-of-b>(%fp), %bax
```

Note the need to save `a` before the call when it is to the left of the operator. Note also that part (iii) gives better code than part (iv) even though it involves more operators. A smart reader might note that (particularly if s/he uses ARM code) that the neg-add sequence could just be an subtraction in part (iii). This doesn't show up so well if IA32 code is chosen:

```
        movl %eax,#7
        call g
        movl %ebx,<offset-of-a>(%fp)
        subl %bax, %eax
        movl <offset-of-b>(%fp), %ebx
```

Then the difference (between code for iii and iv which would be equivalent in C) becomes starker.