Princeton University
COS 217: Introduction to Programming Systems
The ARMv8 Function Call Conventions

When `f()` calls `g()`...

Rule 1

Q: Where will `f()` place the return **address** and where will `g()` find that return **address**?

A: In register X30. That is, the `bl` instruction places the address of the instruction following the `bl` instruction in register X30, and the `ret` instruction branches to the address in register X30.

Rule 2

Q: Where will `f()` place its arguments and where will `g()` find its parameters?

A: In registers R0...R7 in that order.

Rule 3

Q: Where will `g()` place its return **value** and where will `f()` find that return **value**?

A: In register R0.

Rule 4

Q: Which registers may `g()` affect?

A: **Callee-saved** registers (informally, the **g-saved** registers): **R19...R28**
The callee/g **may not** change the contents of those registers.
The callee/g must either:
Not change the contents of those registers, or
Save the contents of those registers before it changes them, and restore the contents before it returns – thus giving the caller/f the illusion that the contents of those registers were not changed.

Caller-saved registers (informally, the **f-saved** registers): **R0...R7, R9...R15**
The callee/g **may** change the contents of those registers.
If the caller/f requires that the contents of those registers be preserved across its call of the callee/g, then the caller/f must do the preserving:
The caller/f must save the contents of those registers before calling the callee/g.
The caller/f must restore the old contents of those registers after calling the callee/g.