Princeton University
COS 217: Introduction to Programming Systems
The System V AMD64 Function Call Conventions

When `f()` calls `g()`:

Rule 1

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

A: In registers RDI, RSI, RDX, RCX, R8, and R9, in that order.

Rule 2

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

A: In register RAX.

Rule 3

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

A: On the stack.

Rule 4

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

A: **Callee-saved** registers (informally, the **g-saved** registers)
RBX, RBP, R12, R13, R14, R15
The callee/g **cannot** 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)
RDI, RSI, RDX, RCX, R8, R9, RAX, R10, R11
The callee/g **can** 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.