Caller Example

- n = sumOf(17,42)

```
movq    $42,%rsi      # load arguments in
movq    $17,%rdi      #   either order, but use
                      #   correct registers
call    sumOf         # jump & push ret addr
movq    %rax,offset_n(%rbp)    # store result
```


37

Example Function

- Source code

```
int sumOf(int x, int y) {
  int a, int b;
  a = x;
  b = a + y;
  return b;
}
```


38

Assembly Language Version

```
# int sumOf(int x, int y) {
#   int a, int b;
sumOf:
    pushq   %rbp    # prologue
    movq    %rsp,%rbp
    subq    $16,%rsp

#   a = x;
    movq    %rdi,-8(%rbp)

#   b = a + y;
    movq    -8(%rbp),%rax
    addq    %rsi,%rax
    movq    %rax,-16(%rbp)

#   return b;
    movq    -16(%rbp),%rax
    movq    %rbp,%rsp
    popq    %rbp
    ret
# }
```


39

Stack Frame for sumOf

```
int sumOf(int x, int y) {
  int a, int b;
  a = x;
  b = a + y;
  return b;
}
```


40