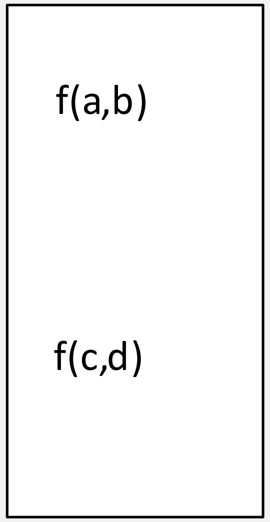
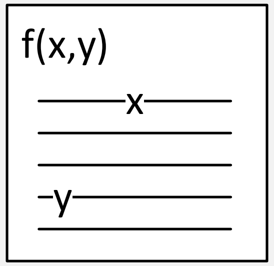
MACROS AND INLINE FUNCTIONS

Eliminating the call and return

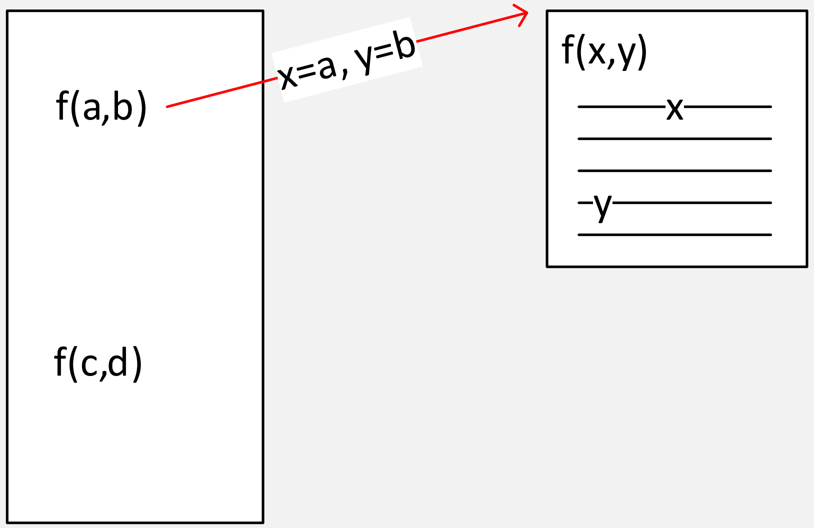
Delroy A. Brinkerhoff

“REGULAR” FUNCTIONS

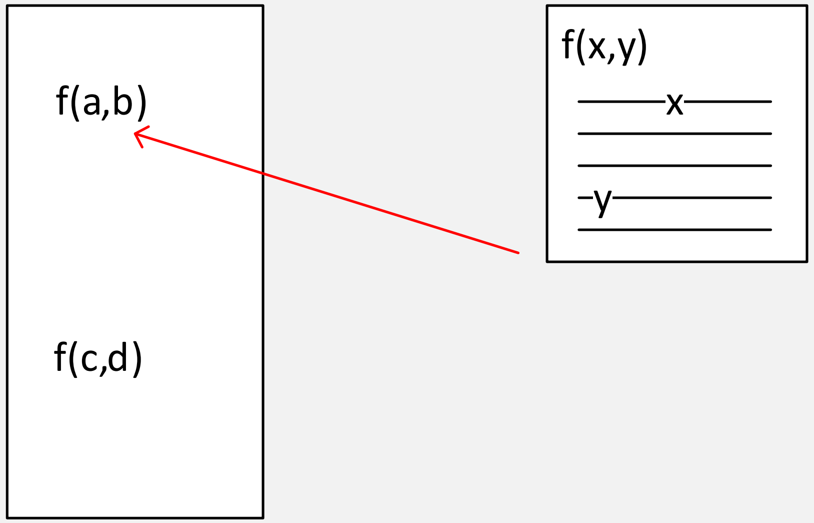




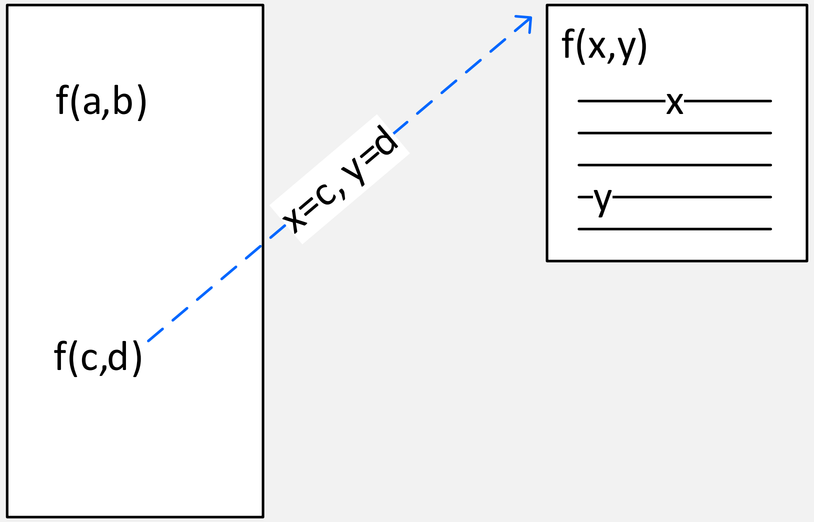
“REGULAR” FUNCTIONS



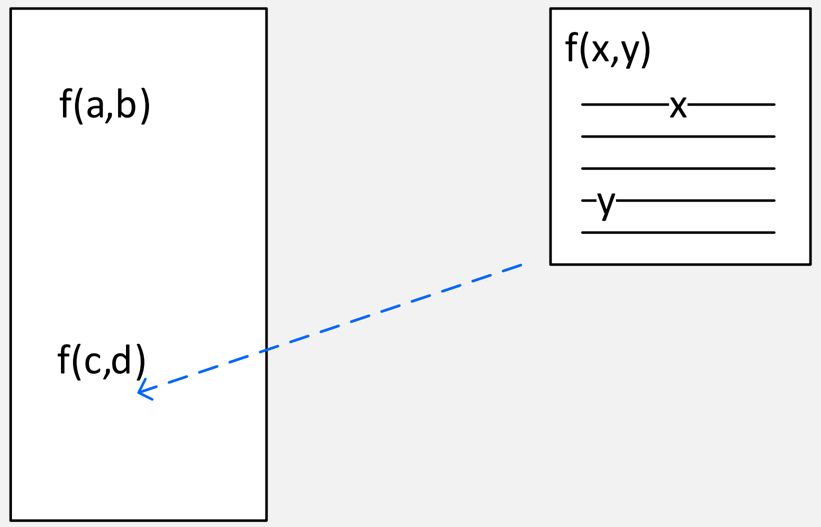
“REGULAR” FUNCTIONS



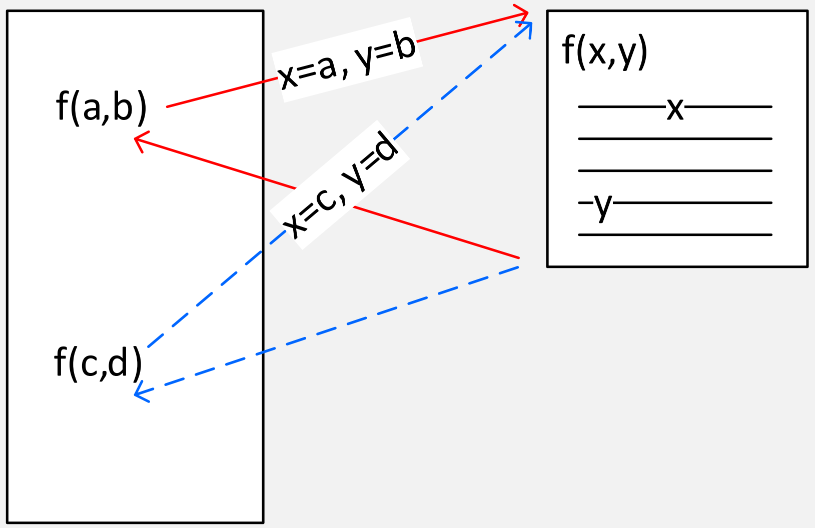
“REGULAR” FUNCTIONS



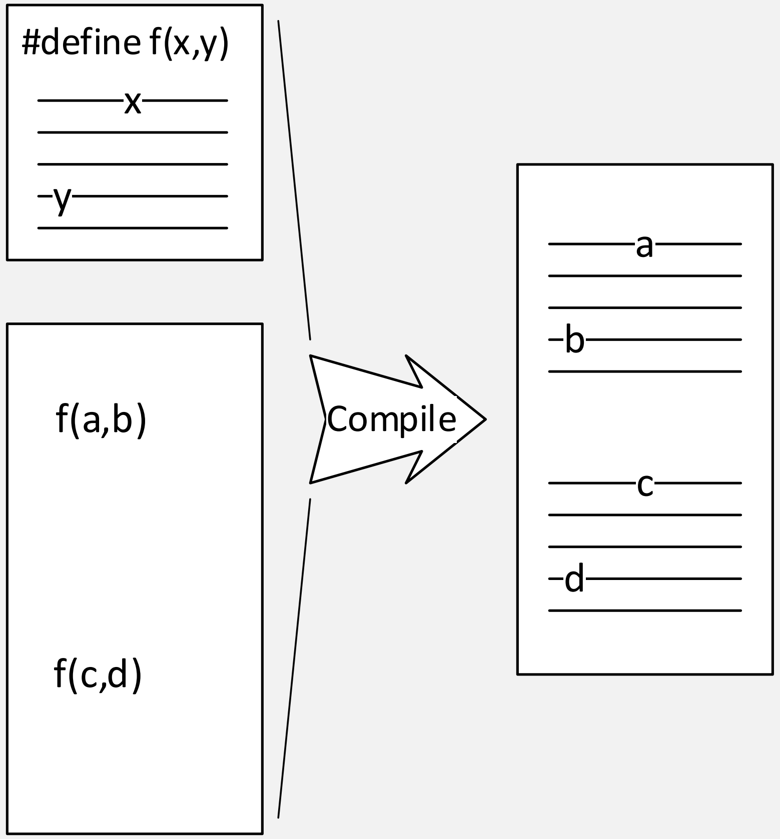
“REGULAR” FUNCTIONS



"REGULAR" FUNCTIONS



PARAMETERIZED MACROS: THE GOOD



```
#define sqr(x) x * x


sqr(5)

5 * 5


sqr(fred)

fred * fred
```

MACRO PROBLEMS:
THE BAD

```
#define sqr(x) (x * x)

sqr(2 + 3)

(2 + 3 * 2 + 3)
```

```
#define sqr(x) ((x) * (x))

sqr(2 + 3)

((2 + 3) * (2 + 3))
```

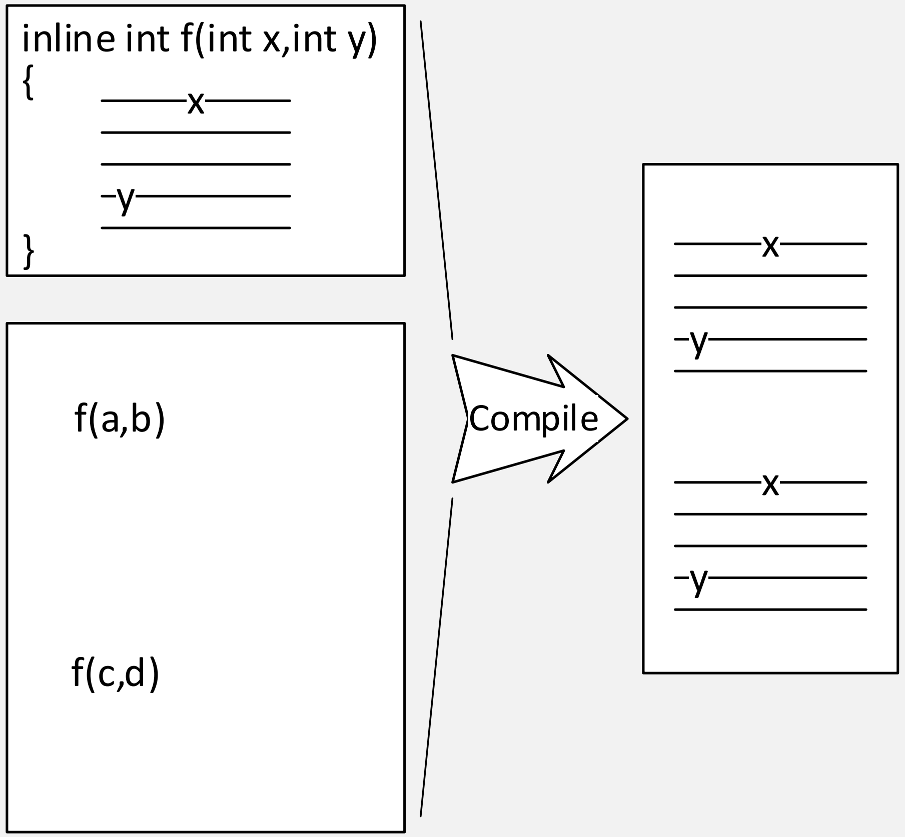
MACRO PROBLEMS: THE UGLY

```
#define min(x,y) (((x) < (y)) ? (x) : (y))
```

```
min(a++, b++)
```

```
(((a++) < (b++)) ? (a++) : (b++))
```

INLINE FUNCTIONS



```
inline int min(int x, int y)
{
        return (x < y) ? x : y;
}

min(a++, b++)
```

INLINE RULES

- “inline” is a suggestion that the compiler may choose to ignore
- Is only appropriate for small functions
- Not appropriate when the address of a function is needed.