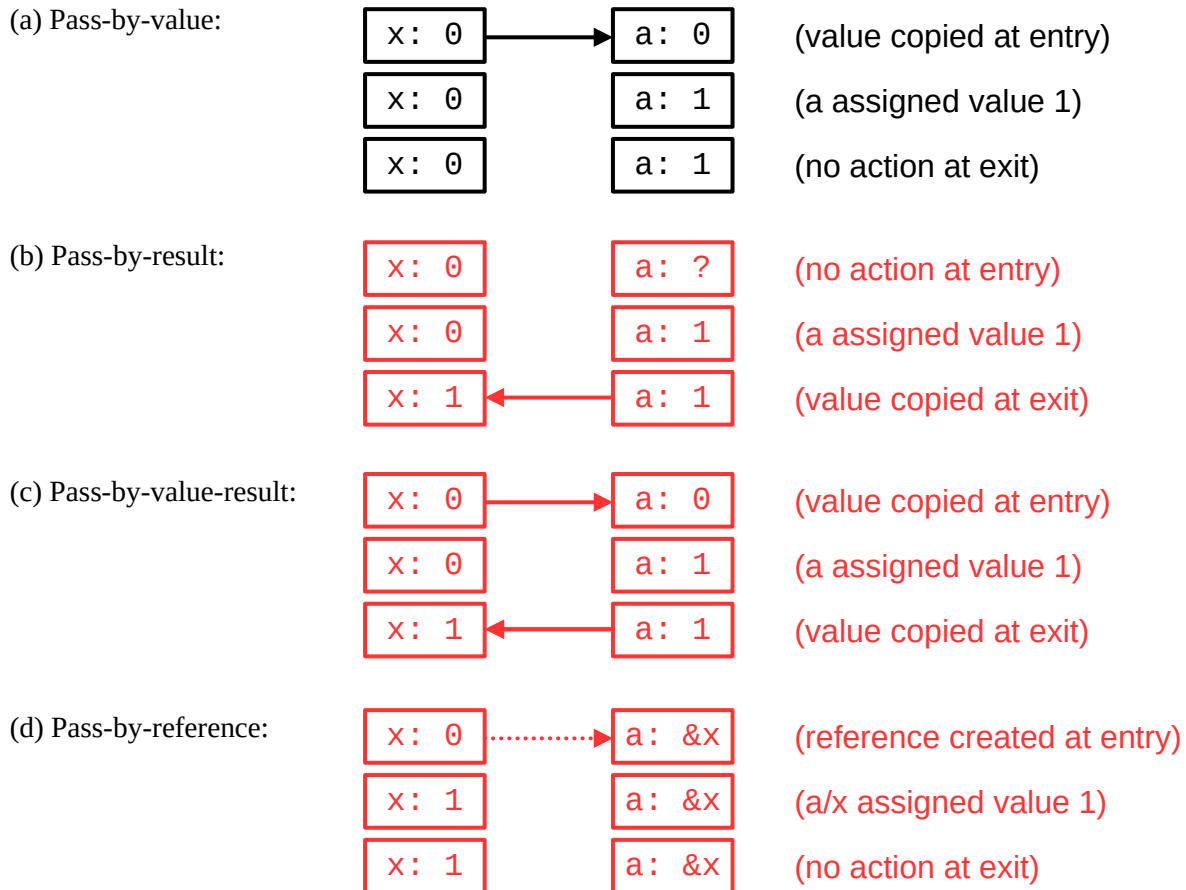


CS 430 Parameters Lab

1. Suppose a subprogram `void f(int a) { a = 1 }` is called with the statement `f(x)`, where `x` is a local variable with initial value 0. Draw pictures and labels to show how the value of `x` is passed between the subprogram and its caller using (a) pass-by-value, (b) pass-by-result, (c) pass-by-value-result, and (d) pass-by-reference. The first one has been completed for you as an example.



2. Which of the five parameter passing mechanisms discussed in class implement *in* mode semantics?

pass-by-value

3. Which of the five parameter passing mechanisms discussed in class implement *out* mode semantics?

pass-by-result

4. Which of the five parameter passing mechanisms discussed in class implement *inout* mode semantics?

pass-by-value-result

pass-by-reference

pass-by-name

For #5-8, consider the following program in a C-like language with static scope. What is the output of this program under each of the following conditions?

```

program p {
  int[] m = [2, 3, 5, 7, 11, 13, 17];
  int x, y = 0, 1;

  void g(int[] a, int b, int c)
  {
    a[b] = 0;
    b += 1;
    y += b;
    a[c] = 8;
  } // end of g

  println(m[0], m[1], m[2], x, y); // Line A
  g(m, x, y);
  println(m[0], m[1], m[2], x, y); // Line B
  g(m, y, y); // different parameters!
  println(m[0], m[1], m[2], x, y); // Line C
}

```

5. All parameters are passed by value.

	m[0]	m[1]	m[2]	x	y
Line A	2	3	5	0	1
Line B	2	3	5	0	2
Line C	2	3	5	0	5

6. All parameters are passed by reference.

	m[0]	m[1]	m[2]	x	y
Line A	2	3	5	0	1
Line B	0	3	8	1	2
Line C	0	3	0	1	6

7. Integer parameters are passed by value and array parameters are passed by reference.

	m[0]	m[1]	m[2]	x	y
Line A	2	3	5	0	1
Line B	0	8	5	0	2
Line C	0	8	8	0	5

8. All parameters are passed by value-result. Results are assigned from right-to-left.

	m[0]	m[1]	m[2]	x	y
Line A	2	3	5	0	1
Line B	0	8	5	1	1
Line C	0	8	5	1	2

(OLDER-STYLE SOLUTIONS W/ INTERMEDIATE WORK)

5. All parameters are passed by value.

	<u><i>p()</i></u>	<u><i>first g()</i></u>	<u><i>second g()</i></u>
2 3 5 0 1	<i>m=[2,3,5,...]</i>	<i>a=[2 0,3 8,5,...]</i>	<i>m=[2,3,5 0 8,...]</i>
2 3 5 0 2	<i>x=0</i>	<i>b=0 1</i>	<i>b=2 3</i>
2 3 5 0 5	<i>y=1 2 5</i>	<i>c=1</i>	<i>c=2</i>

6. All parameters are passed by reference.

	<u><i>p()</i></u>	<u><i>first g()</i></u>	<u><i>second g()</i></u>
2 3 5 0 1	<i>m=[2 0,3,5 8 0,...]</i>	<i>a=&m</i>	<i>a=&m</i>
0 3 8 1 2	<i>x=0 1</i>	<i>b=&x</i>	<i>b=&y</i>
0 3 0 1 6	<i>y=1 2 3 6</i>	<i>c=&y</i>	<i>c=&y</i>

7. Integer parameters are passed by value and array parameters are passed by reference.

	<u><i>p()</i></u>	<u><i>first g()</i></u>	<u><i>second g()</i></u>
2 3 5 0 1	<i>m=[2 0,3 8,3 0 8,...]</i>	<i>a=&m</i>	<i>a=&m</i>
0 8 5 0 2	<i>x=0</i>	<i>b=0 1</i>	<i>b=2 3</i>
0 8 8 0 5	<i>y=1 2 5</i>	<i>c=1</i>	<i>c=2</i>

this is essentially the same as C semantics (for this example, at least)

8. All parameters are passed by value-result. Results are assigned from right-to-left.

	<u><i>p()</i></u>	<u><i>first g()</i></u>	<u><i>return</i></u>	<u><i>second g()</i></u>	<u><i>return</i></u>
2 3 5 0 1	<i>m=[2,3,5,...]</i>	<i>a = [2 0,3 8,5,...]</i>	<i>m=[0,8,5,...]</i>	<i>a=[0,8 0 8,5,...]</i>	<i>m=[0,8,5,...]</i>
0 8 5 1 1	<i>x=0</i>	<i>b=0 1</i>	<i>x=1(from b)</i>	<i>b=1 2</i>	<i>x=1</i>
0 8 5 1 2	<i>y=1 2</i>	<i>c=1</i>	<i>y=1(from c) 3</i>	<i>c=1</i>	<i>y=2(from b)</i>