HW08, CS4610 Programming Languages, Fall 2024

Answer the following questions. One person on your team should turn in a single pdf file that contains all of your answers.

1. Parameter Passing.
   Examine the two functions below in a Java/C-like language. Function *f* is calling function *g*. Consider the value of *i* when print is called. But to know the value of *i* when print is called, you'd have to know the parameter-passing rules of this particular language. For example, if pass-by-value were the rule, i would be 0 when print is called. If pass-by-reference were the rule, *i* would be *2* when print is called. But with a different parameter passing rule(s), the value of *i* would not be known exactly, being either *1* or *2*. Explain the case in which the value of *i* would be either *1* or *2*, but not exactly known for sure simply by looking at the code and knowing the parameter-passing rule. (Hint: If you get stuck, work through each of the 7 parameter-passing rules that we discussed.)

```
void g(a, b) {
        a = 1;
        b = 2;
     }

void f() {
        int i = 0;
        g(i,i);
        print(i);
     }
```

2. More Parameter Passing.
   This code fragment uses arrays in Java. The first line declares and allocates an array of two integers. The next two lines initialize it.

```
int[] A = new int[2];
A[0] = 0;
A[1] = 2;
f(A[0], A[A[0]]);
```

Function *f* is defined as:

```
void f(int x, int y) {
   x = 1;
   y = 3;
}
```

For each of the following parameter-passing methods, say what the final values in the array *A* would be, after the call to *f* finishes. (Note: For some questions there may be more than one correct answer, just give one.)

A. By value.
B. By reference.
C. By value-result.
D. By macro expansion.
E. By name.