VARIABLE PASSING IN MATLAB: THE PASS-BY-VALUE SCHEME

MATLAB programs communicate with their functions using a pass-by-value scheme. This means that when a function call occurs, MATLAB makes a **copy** of the actual arguments and passes them to the function arguments (also called formal arguments). This copying is very significant, because it means that even if the function modifies the input arguments, it won’t affect the original data in the caller. This feature helps to prevent unintended side effects, in which an error in the function might unintentionally modify variables in the calling program.

What happen with the variable names within a function that shares variable names with the caller? The function won’t affect the variables in the caller

1. Write the following function:

```
function out = guessresult(aa)
        % Passing-by-value scheme
        aa = aa + 1;
        out = aa;
end
```

Explain why the value of **aa** is 5 before the function **guessresult** is executed and 5 afterwards as shown in the following statements on the Command Window:

```
>> aa=5
aa =
    5

>> guessresult(aa)
ans =
    6

>> aa
aa =
   5
```

2. Add some print statements to the previous function:

```
function out = guessresult(aa)
        % Passing-by-value scheme

        fprintf(‘ Inside-the-function starting value of aa=%d \n’, aa);
```

```
            aa = aa + 1;
            out = aa;
            fprintf(‘ Inside-the-function final value of aa=%d \n’, aa);
    end
```

Explain why the value of **aa** is 5 before the function **guessresult** is executed and 5 afterwards as shown in the following statements on the Command Window:

```
>> aa=5
aa =
    5

>> guessresult(aa)
 Inside-the-function starting value of aa=5
 Inside-the-function final value of aa=6
ans =
    6

>> aa
aa =
   5
```

3. Modify the previous function to:

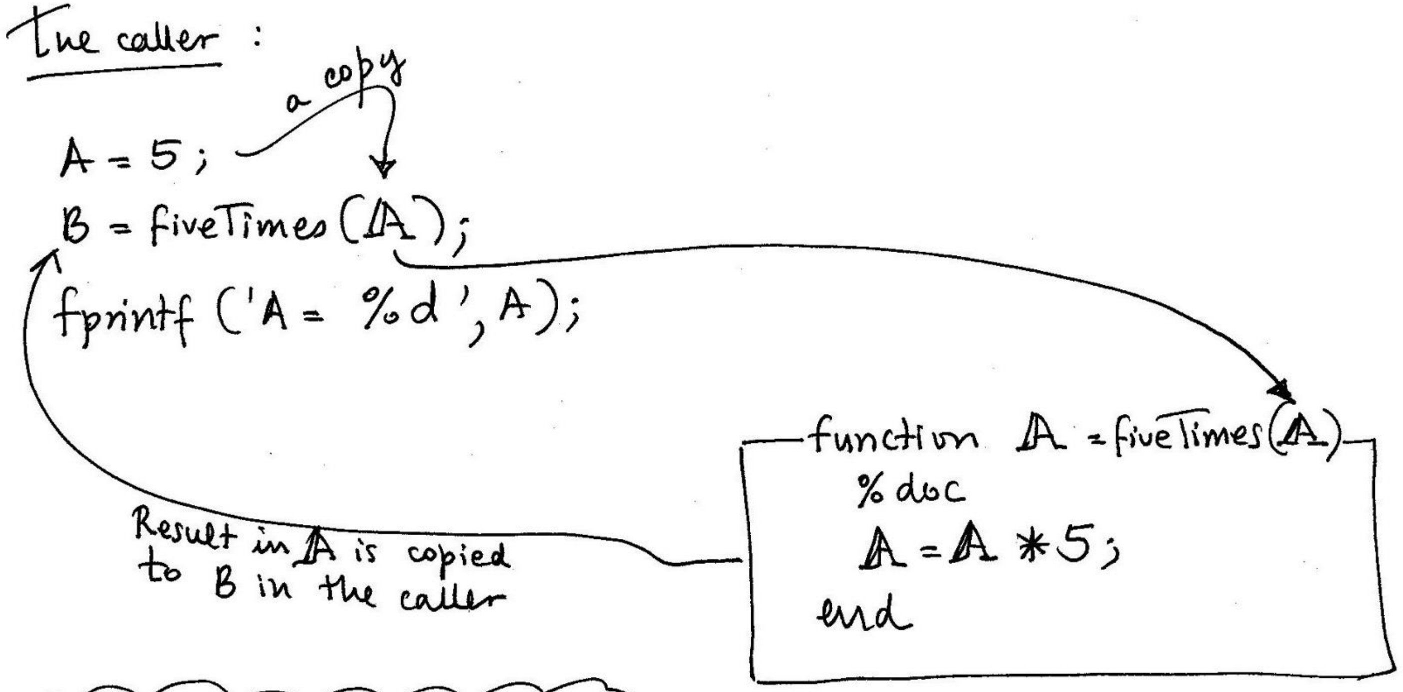
    ```
    function aa = guessresult(aa)
    % Passing-by-value scheme
    fprintf(‘ Inside-the-function starting value of aa=%f \n’, aa);
    aa = aa + 1;
    fprintf(‘ Inside-the-function final value of aa=%f \n’, aa);
    end
    ```

    And call the function as before. Explain the results
    In the function header, what does the variable **aa** at the left refers to?

4. Modify the previous function to:

    ```
    function out = guessresult(aa)
    % Passing-by-value scheme
            fprintf(‘ Inside-the-function starting value of aa=%f \n’, aa);
            out = aa + 1;
            fprintf(‘ Inside-the-function final value of aa=%f \n’, aa);
    end
    ```
    And call the function as before. Explain the results

FUNCTION PASS-BY-VALUE SCHEME



The caller :

```
A = 5;
B = fiveTimes(A);
fprintf('A = %d', A);
```

a copy

```
function A = fiveTimes(A)
  % doc
  A = A * 5;
end
```

Result in A is copied to B in the caller

A in the caller is not affected

i.e.

You will print A=5,

EXERCISES

1.-

```
% The Caller:
A=5;
B=fiveTimes(A);          % the call
fprintf(‘A=%d’, A);
```

```
function A=fiveTimes(A)
% Multiple A by 5
          A=A*5;
end
```

You have to write the function independently of the caller program

What is the output of the above program?

2.-

```
% The Caller:
A=5;
B=fiveTimes(A);          % the call
```

```
function A=fiveTimes(A)
% Multiple A by 5
          A=A*5;
          fprintf(‘A=%d’, A);
end
```

What is the output of the above program

3.-

```
% The Caller:
A=5;
A=fiveTimes(A);          % the call
fprintf(‘A=%d’, A);
```

```
function A=fiveTimes(A)
% Multiple A by 5
          A=A*5;
          fprintf(‘A=%d’, A);
end
```

What is the output of the above program?