# CS 4250 PROGRAMMING LANGUAGES FALL 2018

**KEY TO QUIZ #8 [17 points]**

1. [3 points] (a) Define an activation record:

*Answer*:

The format, or layout of the noncode part of a subprogram is called an *activation record*.

(b) Define an activation record instance:

*Answer*:

A concrete example of an activation record (a collection of data in the form of activation record) is called an *activation record instance*.

1. Define a subprogram linkage.

*Answer*:

The subprogram call and return operations are together called *subprogram linkage*.

1. [2 points] Fill in the blanks in the following statements:
2. A pointer to the base of the activation record instance of the caller is called the *dynamic link*.
3. A chain of static links that connect certain activation record instances in the stack is called a *static chain*.
4. The *static link* points to the bottom of the activation record instance of an activation of a static parent. (Sometimes it is called a static scope pointer.)
5. An integer associated with the static scope that indicates how deeply it is nested in the outermost scope is called a *static depth*.
6. [2 points] Some actions performed at the time of call to a simple subprogram and the time of return from a simple subprogram are listed in the table below. Fill in the right column of the table: put (1) if the action is performed by the caller and (2) if the action is performed by the called.

|  |  |
| --- | --- |
| Action | (1) or (2) |
| Allocate local variables | **(2)** |
| Transfer control to the called | **(1)** |
| Compute and pass the parameters | **(1)** |
| Restore the stack pointer by setting it to the value of the current EP minus one. | **(2)** |

1. [2 points]

|  |  |
| --- | --- |
| Consider the following skeletal Python program:  # global scope  …  **def** f1():  **def** f2():  **def** f3():  …  # end of f3  #end of f2  # end of f1 | Indicate static depths of:  global scope ===> **0**  -------------------------------------  f1 ===> **1**  --------------------------------------  f2 ===> **2**  ------------------------------------  f3 ===> **3** |

1. [6 points] Consider the code fragment and the sequence of function calls on the left side in the table below. The diagram for the stack contents for the point 1 is shown on the right side of the table.. However, some important details are missing on this diagram. Please, restore what is missing: **names of local variable and parameters for all functions, names of functions where control will be returned, and arrows representing the direction of dynamic links**

|  |  |
| --- | --- |
| Sequence of function calls is: |  |

1. [2 points] Explain why the return address, dynamic link, possibly, static link and parameters are placed in the bottom of activation record instance.

*Answer:*

This information is placed at the bottom of ARI because it comes from the caller (the subprogram that initiates a call to the given subprogram).