PRESENTATION OF RUNTIME STACK

INTRODUCTION TO RUNTIME STACK IN PYTHON :-

- The runtime stack in Python is a data structure that keeps track of function calls during program during execution.
- It is responsible for storing and organizing information about the function call, including Function arguments, local variables, and the return address.
- The runtime stack works on a last-in-first-out (LIFO) basis, meaning that the most recently called function is at the top of the stack, and the first called function is at the bottom.
- When a function returns, its corresponding frame is removed from the stack, and control is passed back to the calling function runtime stack in python.

What is runtime stack in python?

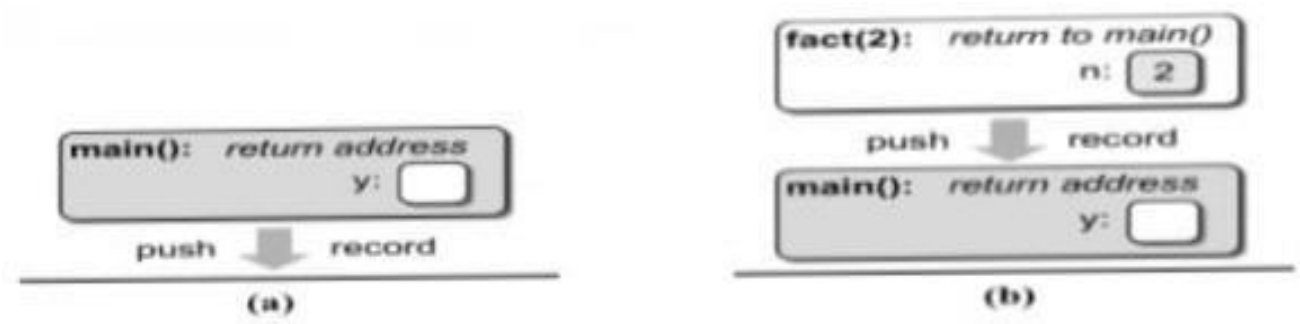
- Runtime stack is a portion of memory used by a running program.

RUNTIME STACK :-

1.The run time stack is just like the stack structure but it's hidden from the programmer and is automatically maintained.

2. Consider the execution of the following code segment, which uses the factorial function.

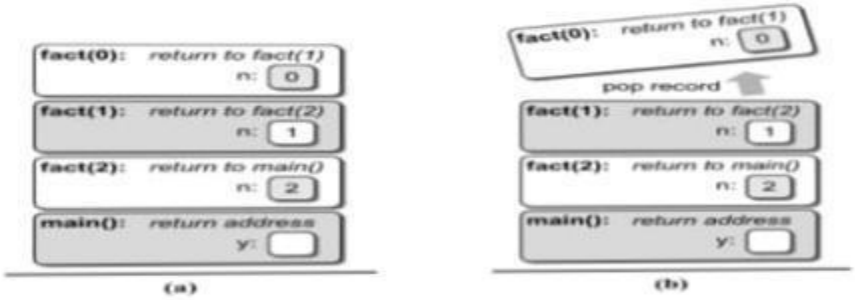
3. When the main routine is executed, the first activation record is created and pushed onto the run time stack, as illustrated in Figure (a).



4. When the factorial function is called, the second activation record is created and pushed onto the stack, as illustrated in Figure (b).

The factorial function is recursively called until the base case is reached with a value of n = 0.

- At this point, the run time stack contains four activation records, as illustrated Figure (a).
- When the base case statement is executed, the activation record for the function call fact(0) is popped from the stack, as illustrated in Figure (b), and execution returns to the function instance fact(1).
- This process continues until all of the activation records have been popped from the stack and the program terminates.



Working of runtime stack in python?

- When a function is called, a new frame is added to the runtime stack to store information about the function call, such as function arguments, local variables, and the return address.
- The frames are organized in a stack with the most recently added frame at the top of the stack. When a function returns, its frame is removed from the stack.
- The runtime stack is an essential part of the Python interpreter, allowing it to keep track of function calls and enabling efficient and bug-free Python code.

Function Calls and the Runtime Stack :-

- This frame contains information such as the function arguments, local variables, and the return address.

  When the function returns, the frame is removed from the stack.If a function calls another function, a new frame is added to the stack for the second function,and so on.