RECURSION.

(1) When a function calls itself until a specified condition is met.

```
f(n) {  ← a function
    f(n) ← the same function is called again.
}
```

(2) Recursion to print 1.

```
(3) → static void f() {
        S.O.P(1); (4) // Prints 1
    (5) → f();    // calling same function again.
    }

(1) → P S V M ( ) {  ← Program first comes into the main
(2) →   f();
    }
```

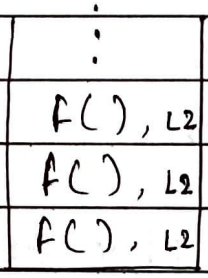
output screen

1
1
1
.
.
. . . . . .

This program that we saw previously, it will keep on going forever since there is no stop condition or base condition. (Infinite Recursion.)

The program will eventually stop in such cases only when it runs out of memory. This is called Stack overflow.

The function f() again calls itself but f() is not yet completed. So, f() will wait in stack.



Stack

The function calls are waiting in memory.

- BASE CONDITION.

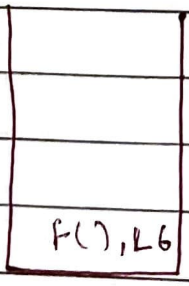
```
① cnt=1;
② f(){
③ if(cnt == 2) return;      // Base condition.
④ Print(cnt);
⑤ cnt++;
⑥ f();                      }
⑦ Main(){
⑧ f();                      ←
⑨ }
```

when return is executed, no line below it will be executed

Output

1

when a function call gets over that function is removed from the stack.



Stack.