

How to avoid memory lake

- We need to know how much data storage we are going to consume. Then we should review every function or procedure or code block, which does not overpass that storage already fixed.
- We can create dynamic memory to be used, However, a problem can arise if we do not properly pay attention at how to manage and free that memory along the execution program.
- Every time we free the memory we needed at that time, we should make sure that we do the same action for every variable.
- A memory lake could appear if we allocate dynamic memory for a variable and then, we do not free that memory before allocating dynamic memory again. It could be confusing at times. So we need to be careful about dynamic memory allocation.
- If we need examples about how to create dynamic memory and free it afterwards, you have various code C/C++ examples on the net.
- We should create a procedure or function which we could call every time we allocate dynamic memory so that we know the memory that we need to free then.
- We could create different procedures which log and print out all the dynamic memory that the program is using along its execution.
- We could create different procedures which log and print out every time we create and free dynamic memory.