# Assignment description

**Objective** The objective of this assignment is to implement the memory management of the Jack operating system.

**Contract** Complete the implementation of *Memory.jack* and compile it to obtain *Memory.vm*

**Resources** You will need three tools: the Jack compiler, to translate your program into a set of *.vm* files, the VM emulator, to run and test your translated program, and the Jack Operating System.

For details read section 12.3.7 of the text book.

## Details

Read Chapter 12 of the text book. Make sure to understand sections 12.1.2 and 12.3.7 very well. You can start by implementing a memory allocation mechanism that follows the pseudocode in Figure 12.6a. Then you can extend your implementation and design a proper deallocation mechanism.

Your program will be evaluated using the test program (in folder *tester*) that performs memory allocations and deallocations in a loop. The program is designed to check the following:

- The memory segments corresponding to allocated variables are not corrupted
- The deallocated memory segments can be reused
- `Memory.alloc(K)` returns successfully if and only if K+1 consecutive memory words are free.
    - Example: Assume your memory has size 20 and you perform:

```
i = Memory.alloc(10);
j = Memory.alloc(5);
Memory.deAlloc(j);
k = Memory.alloc(7);
```

    - Are you sure the last allocation will succeed in your implementation?

## Hints

1. As the Jack OS uses the Memory class to instantiate objects, it will not instantiate Memory. Do not write a constructor in the Memory class; that would never be called. If you need class variables within Memory you need to use static variables.
2. To test the efficiency of your implementation simply compile *Memory.jack* and obtain *Memory.vm*. Then move *Memory.vm* into the *tester* folder. Finally, use the VM emulator to load all the *tester/*.vm* files and run the program. (Note: if you do not copy your *Memory.vm* into the *tester* folder, the program will use the *Memory.vm* implementation that comes with the Jack OS. You can use it as a reference to assess how well your program performs.)

## Submission

You should submit the Memory.jack file, which should include the complete implementation of the

functions listed within Memory.jack.

In one zip-archive, include the Memory.jack file and the declaration.

Make sure that the zip filename and the email's subject field contain EP1200-Seminar10-groupN-Firstname-Lastname, if you are in seminar group N.