ESV Coding Problem

## The problem

Given a fixed amount of memory (100,000 blocks of 1024 bytes) we require a system that allows clients to request and have allocated quantized blocks of storage. This is analogous to the allocation of memory from a heap.

Once a client has been assigned a block (or set of blocks) they must then be able to write and read arbitrary binary data to and from it. In addition once an area of memory has been assigned it must not be reassigned until it has been released.

The following is an illustrative diagram of the state of the storage after a series of allocations and releases:



## What we want you to code

Using either Scala (or Java), we would like you to develop an efficient system for allocating data storage to clients. This should involve the implementation of a single method, allocate, that returns a DataBlock. In Scala, the allocate method should look something like this:

def allocate(numBlocksRequired: Int): DataBlock

The DataBlock’s trait (interface) should include the following methods:

def write(data: Array[Byte]): Boolean

def read(): Array[Byte]

When implementing the allocate method we would like you to pay particular attention to the way in which free storage is tracked and reallocated after being released.

Provide any necessary unit tests using the testing framework of your choice.

## What we want to discuss with you

Although we only require you to implement the allocate method of this storage manager we would also like you to consider the following aspects of the system:

* What should happen when a client releases memory back to the storage manager?
* How can data be efficiently written and read?
* How should the fragmentation of the available storage be managed?

We look forward to discussing these issues with you at your interview.

## What we don’t expect you to do

We don’t expect you to develop a complete heap allocation system or provide any written documentation (other than comments) for your solution.