

## CS224 – Memory Management

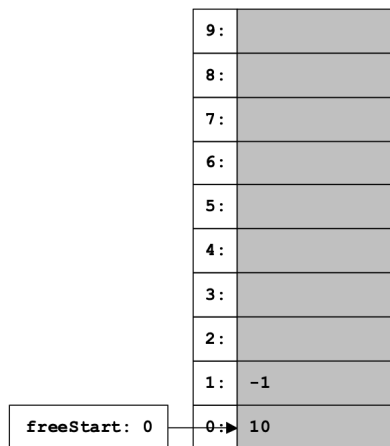
**Purpose:** You have seen that items like objects and arrays require that we allocate memory. The block of memory used for dynamic allocation is called the Heap. We will examine the management of this memory now.

**Knowledge:** This activity will help you become familiar with the following content knowledge:

- What happens to the heap during allocation and deallocation of memory.
- What mechanisms are required for memory management

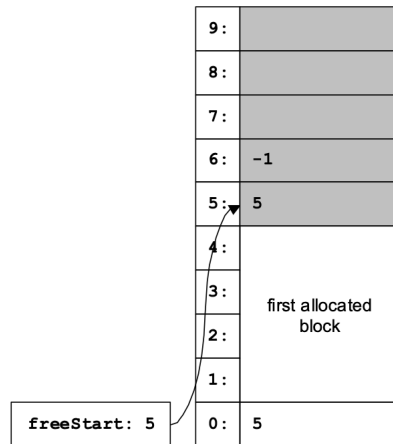
**Activity:** With your group perform the following tasks and answer the questions. You will be reporting your answers back to the class in 15 minutes.

1. We store the free memory locations in the heap as a linked list and the link to the head of the free list is held in **freeStart**. Every block, allocated or free, has its length in its first word. Free blocks have a free-list link in their second word, or -1 at the end of the free list. This illustrated in a small example with a heap with 10 words of memory.



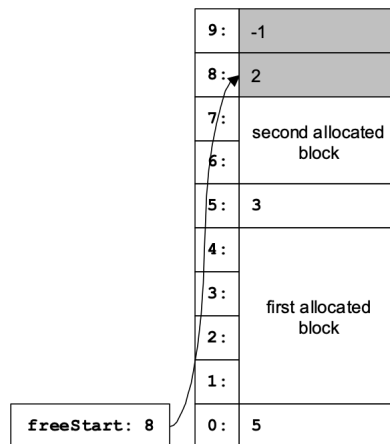
- (a) What does value 10 mean in memory address 0?
- (b) What does the value -1 mean in memory address 1?

2. Now we will allocate memory. Assume that `Memory.alloc(4)` is requesting 4 words to be allocated and will return the address of this block. So after `p1=Memory.alloc(4)` we have:

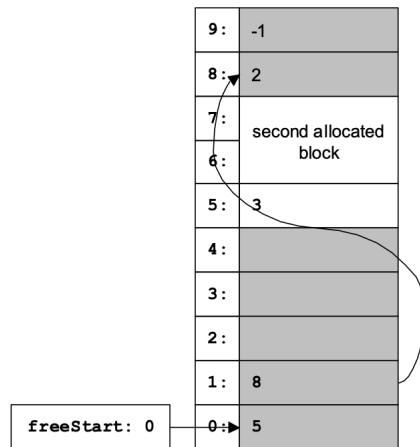


- (a) Why did this allocation use 5 memory locations rather than 4?
- (b) What does the value 5 mean at memory address 0?
- (c) Why should the address stored in `p1` be 1 rather than 0?

After a second allocation `p2=Memory.alloc(2)` the situation looks like:

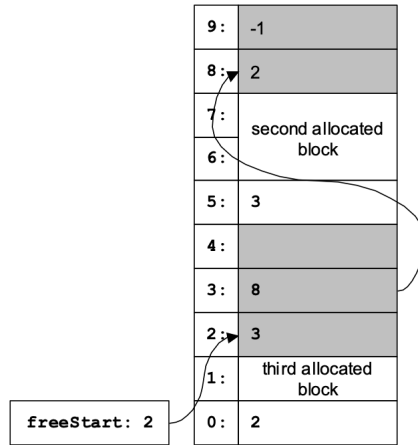


3. Next we will deallocate the object `p1` by performing `Memory.dealloc(p1)`. This memory needs to be returned to the free list so it can be used again.



- What do the 5 and 8 mean in memory addresses 0 and 1?
- What advantage do we get from maintaining the free memory in a linked list over one big block of memory?
- What disadvantage do we have from maintaining the free memory in a linked list over one big block of memory?

4. We will do another allocation: `p3=Memory.alloc(1)` :



Finally, we will do one more allocation: `p4=Memory.alloc(4)`

- What is the problem with this last allocation?
- What can be done to solve this problem?