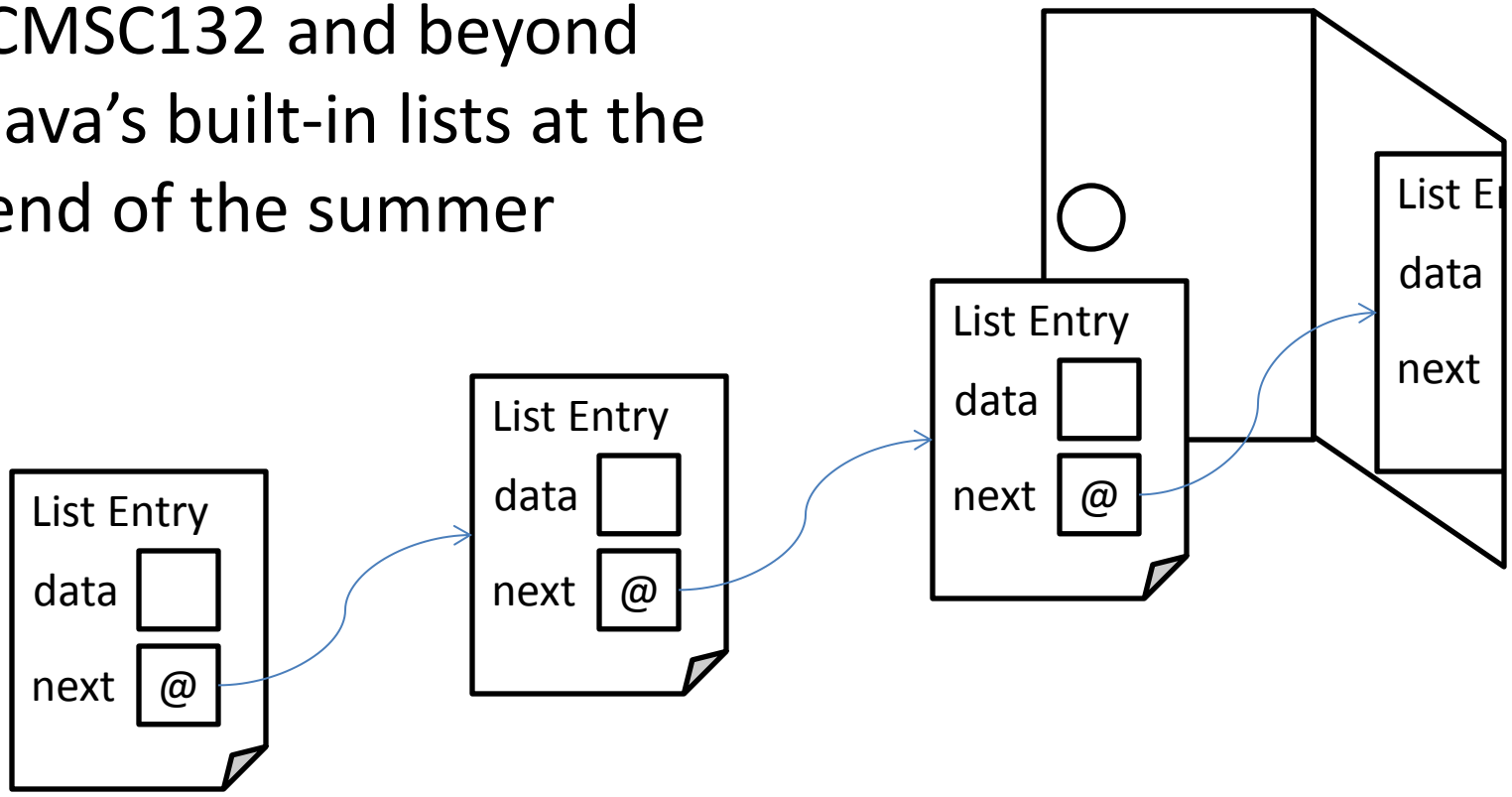
Announcements

- Lab06 tomorrow
- No discussion/quiz Thursday
- Review session Friday
- P4 due Monday

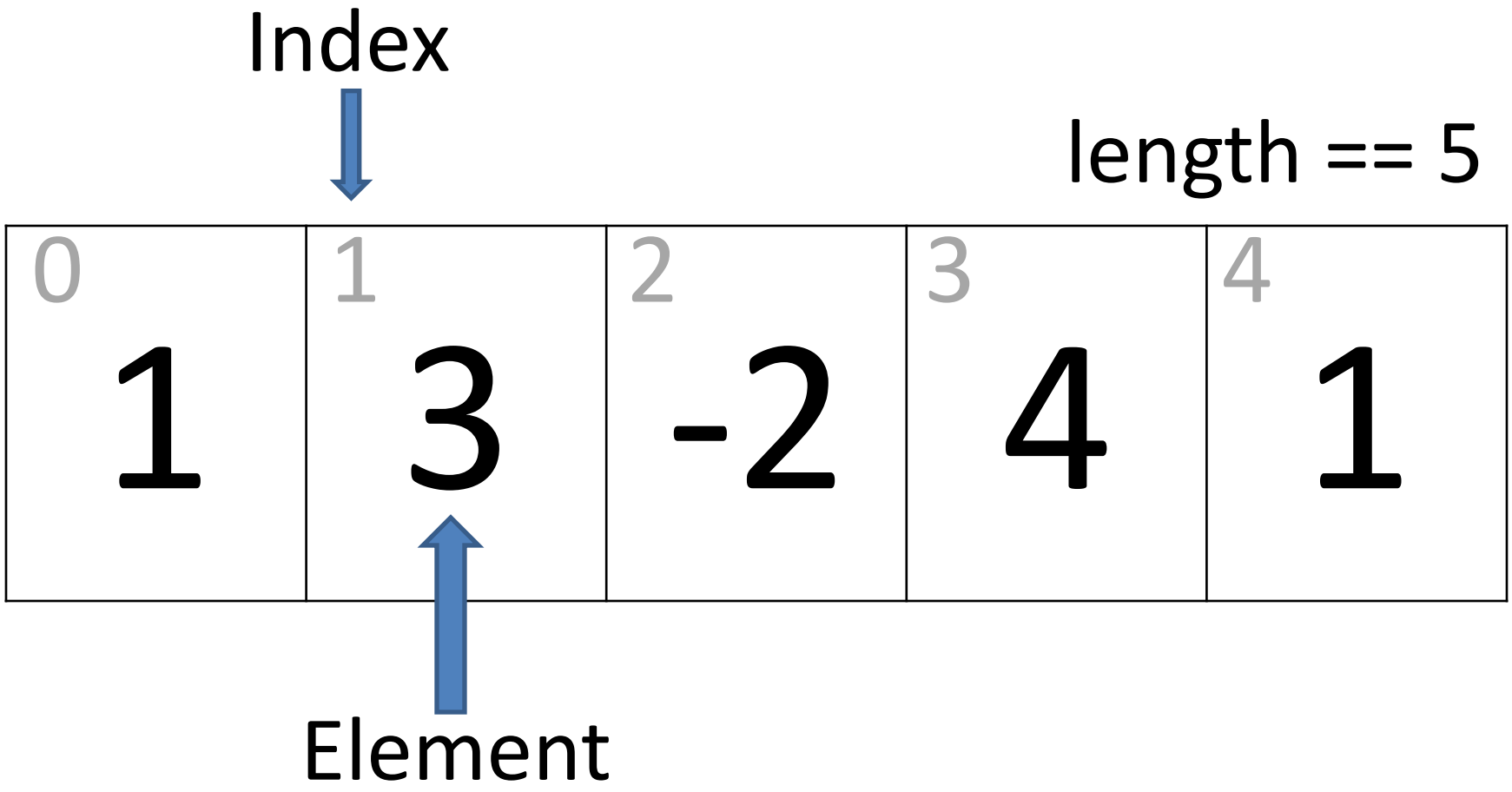
Goodbye lists

You will see lists again:

- CMSC132 and beyond
- Java's built-in lists at the end of the summer



Hello arrays

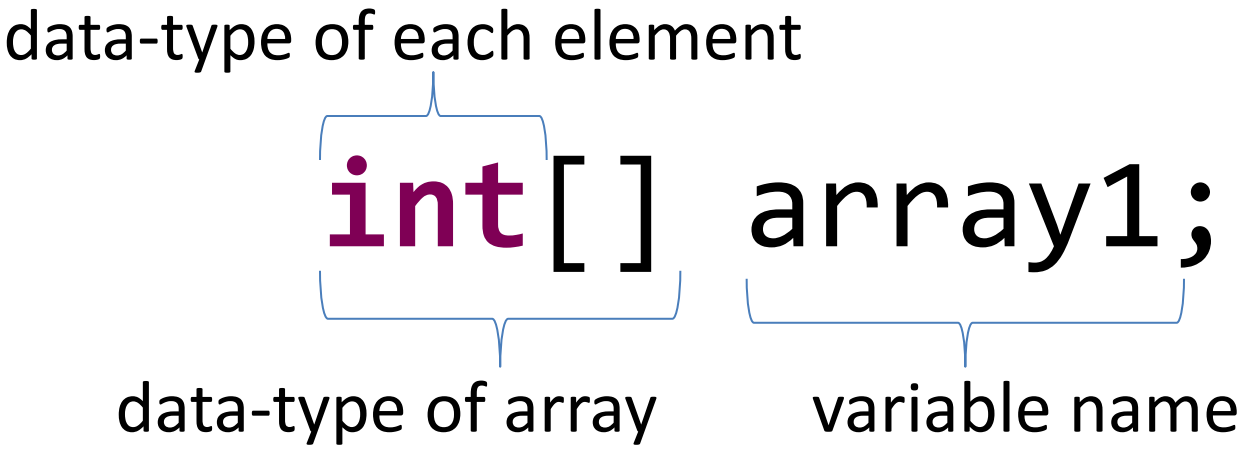


Arrays

- An array is a contiguous block of memory
  - Each element is the same data-type
  - Elements are numbered from zero
- Can be allocated at runtime
- Each element can be accessed directly
  - No list traversal

Array Syntax

- Declaring arrays:



- Alternate syntax (discouraged):

```
int array2[];
```

Array Syntax

- Initializing arrays:

```
//Allocate an array with 3 elements.
//Elements default to zero.
array1 = new int[3];
//Allocate an array with 3 elements.
//Elements are specified in {…}.
array2 = {1, 4, 3};
```

Array Syntax

- Getting individual elements:

```
int x = myArray[index];
```

- Setting individual elements:

```
myArray[index] = value;
```

- Accessing array length:

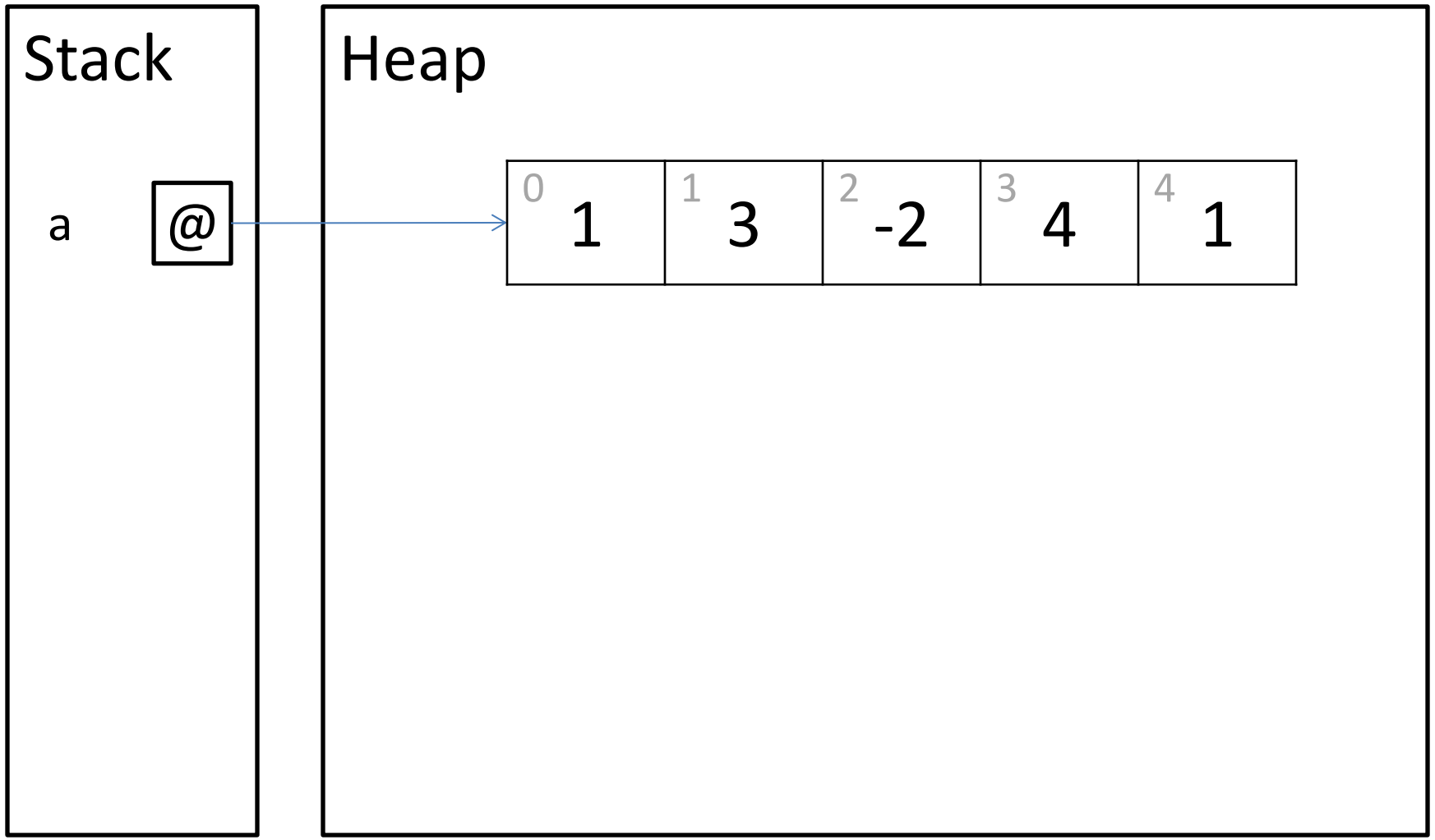
```
int len = myArray.length;
```

- Histogram example

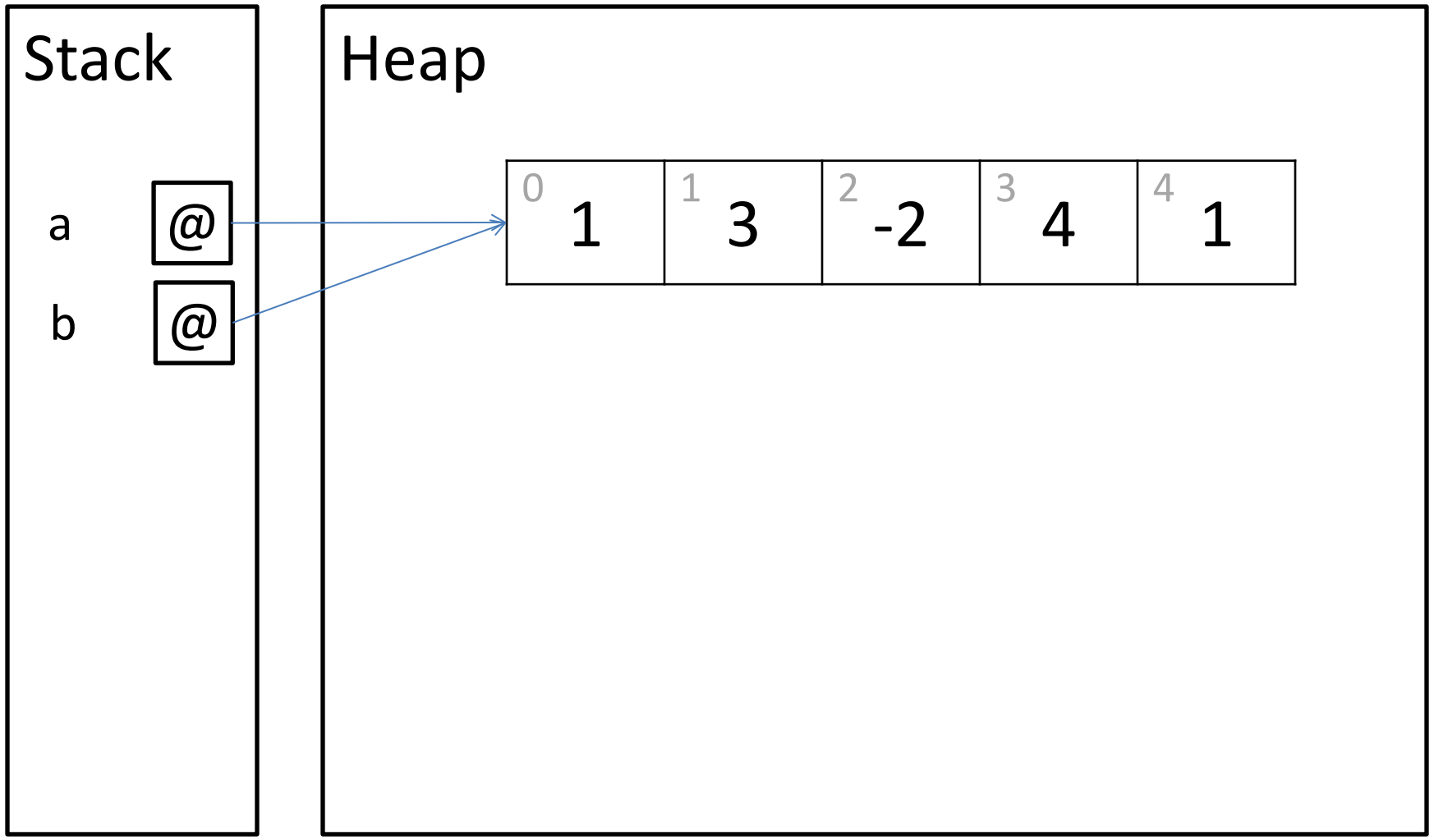
Arrays in memory

- Arrays are allocated on the heap just like objects
  - Array variables are reference variables
    - Aliasing is possible
  - Length is determined at the time of allocation and is fixed there-after
  - Contents can be accessed and changed throughout execution
- Array elements can be any data-type
  - Primitives: ints, booleans, doubles, etc.
  - Reference variables: Objects, other arrays

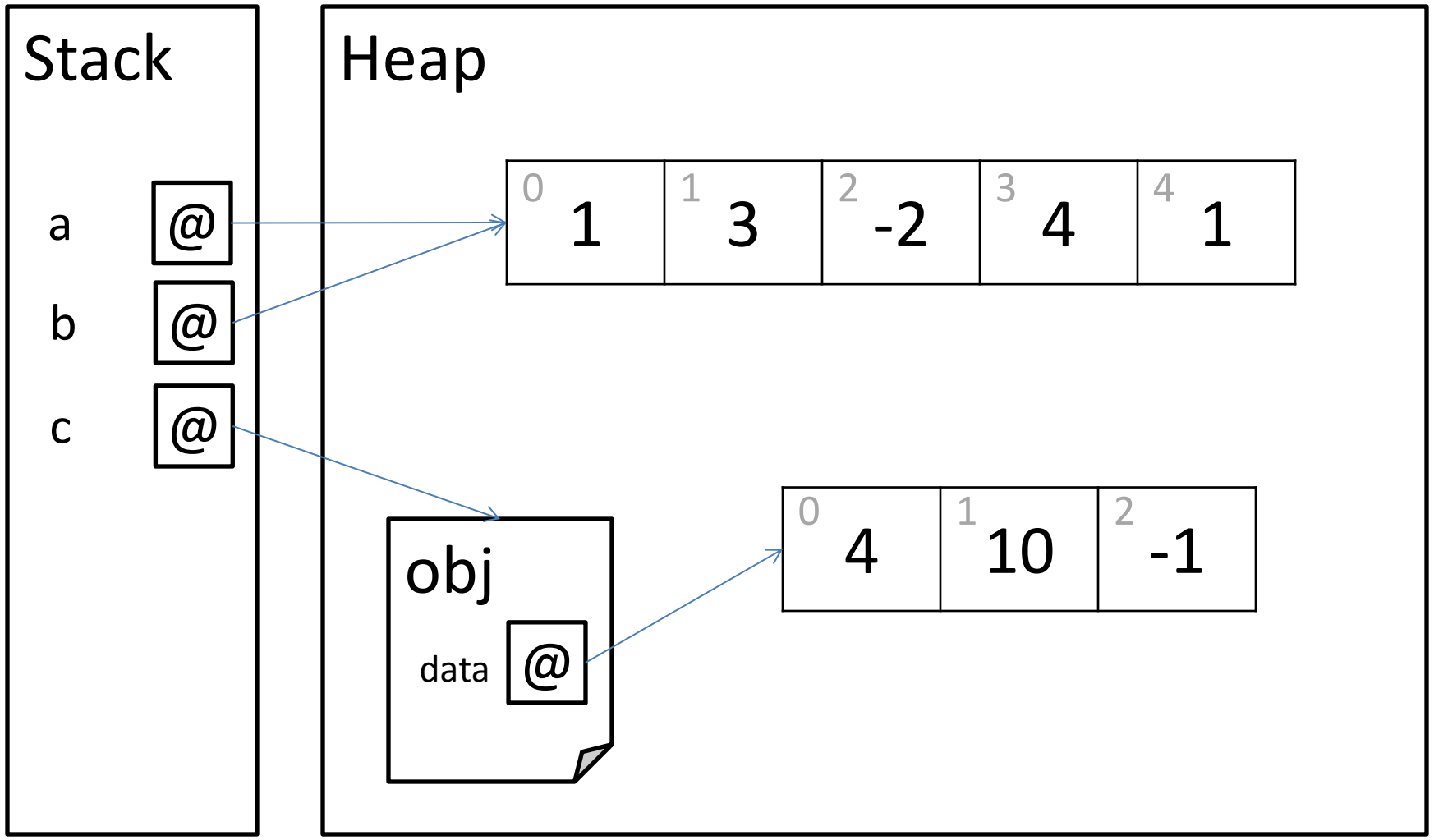
Array reference variables



Array reference variables

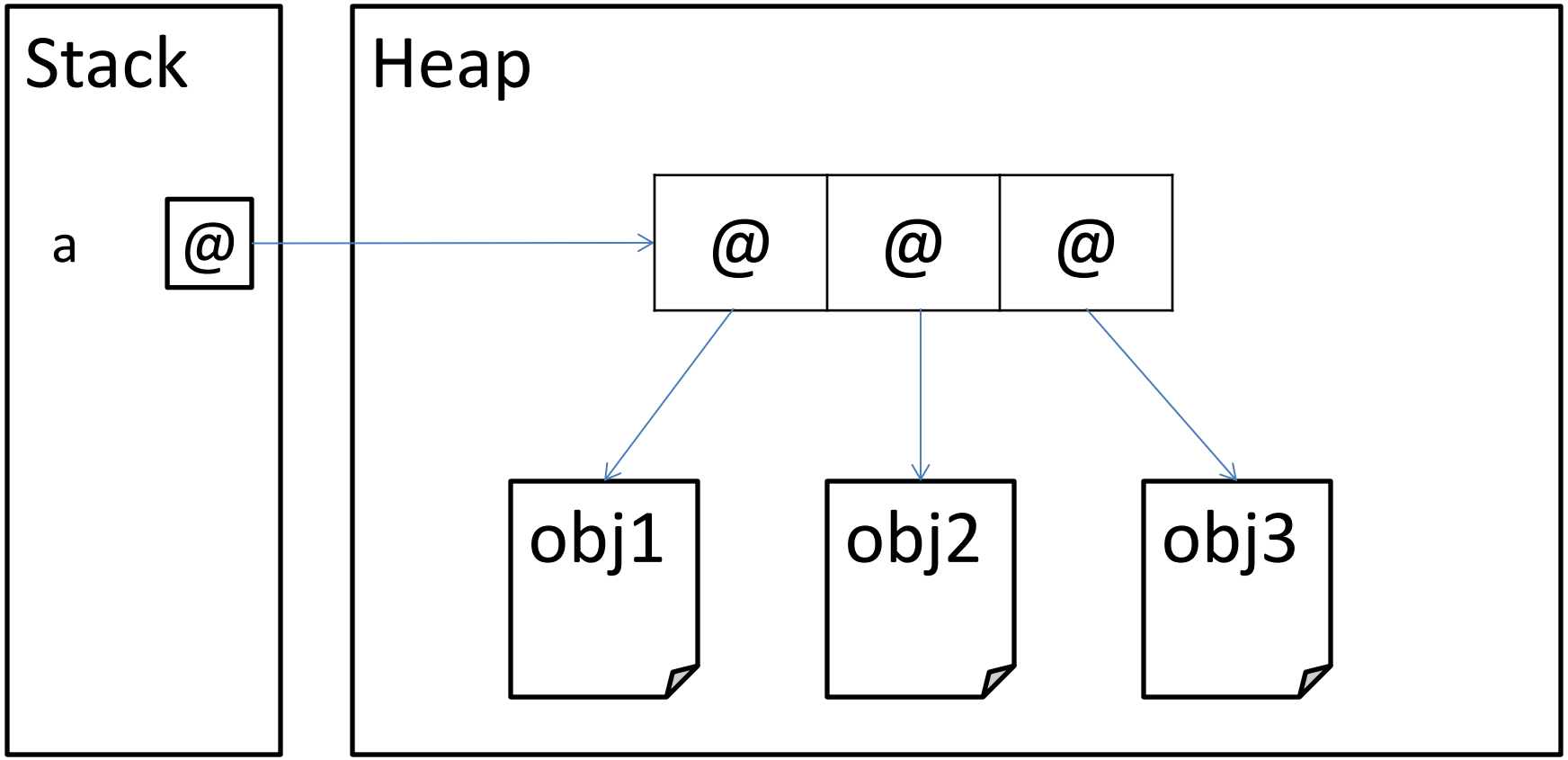


Array reference variables



Arrays of references

- Student array example



Iterating over arrays

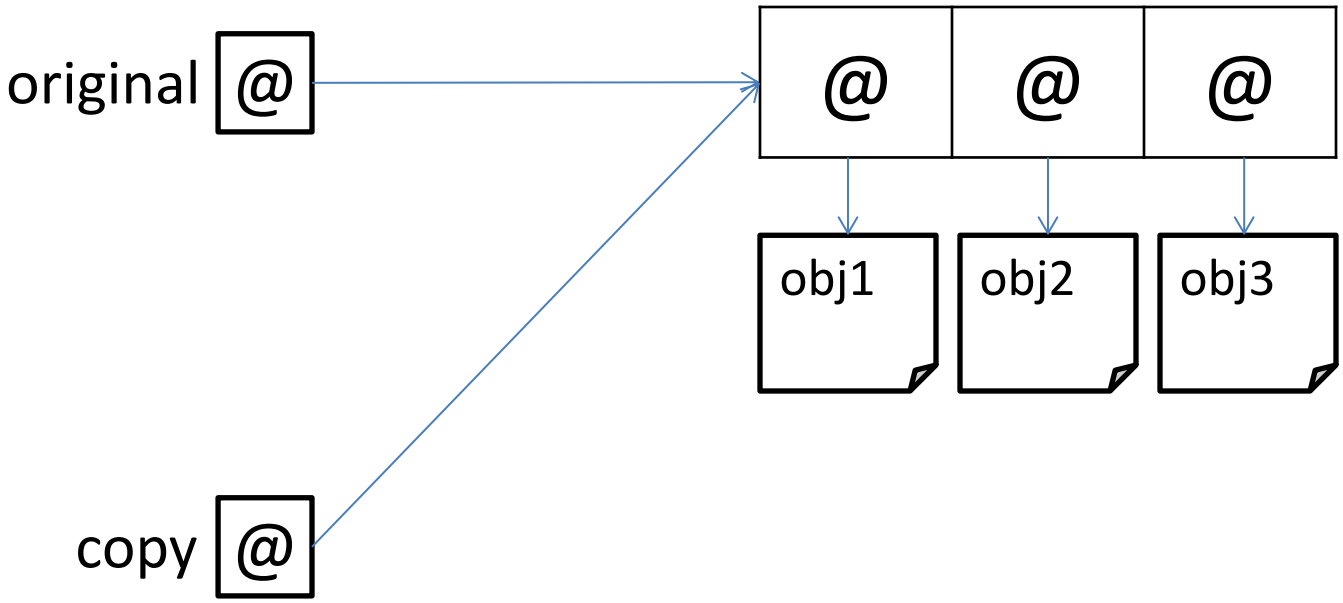
- Arrays are often used in conjunction with loops:

```
//Print the contents of array on a single line
for(int i = 0; i < array.length; i++) {
      System.out.print(array[i]+" ");
}
```

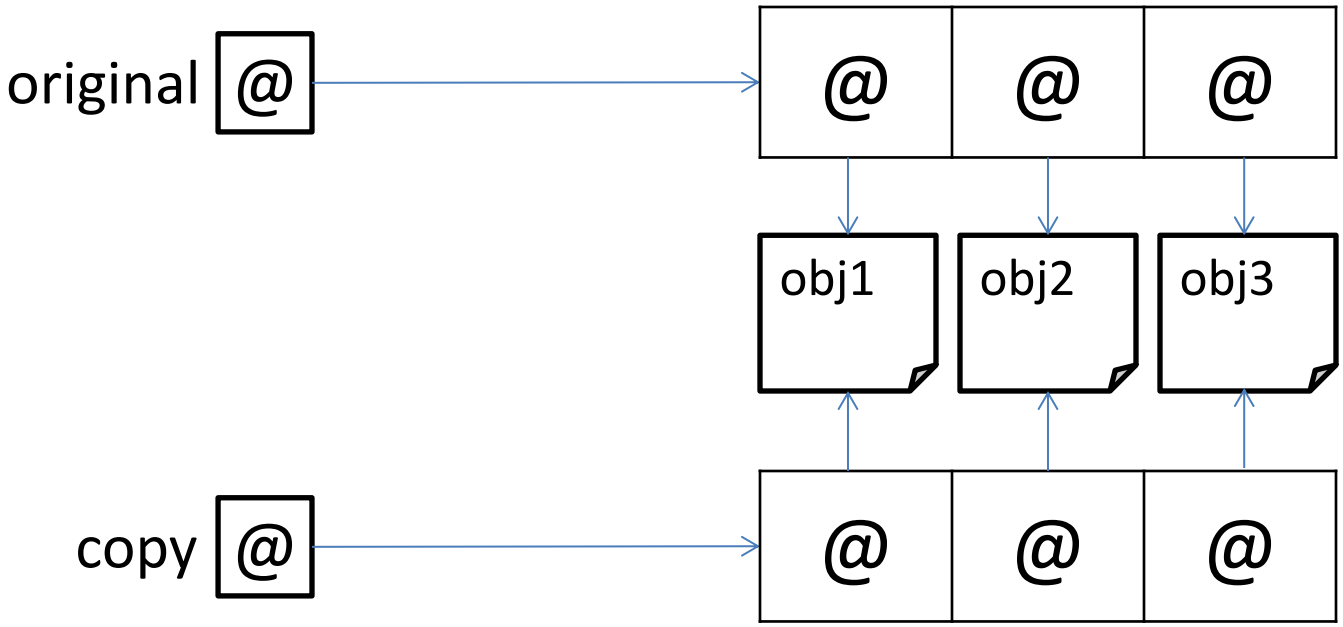
Copying arrays

- Often you may need to copy an array
  - Adding additional storage space (extend example)
  - Inserting additional elements (insert example)
- Changes to the original should not affect the copy
  - Aliasing may not be sufficient
- What if the array elements are not primitive?
  - Student array example

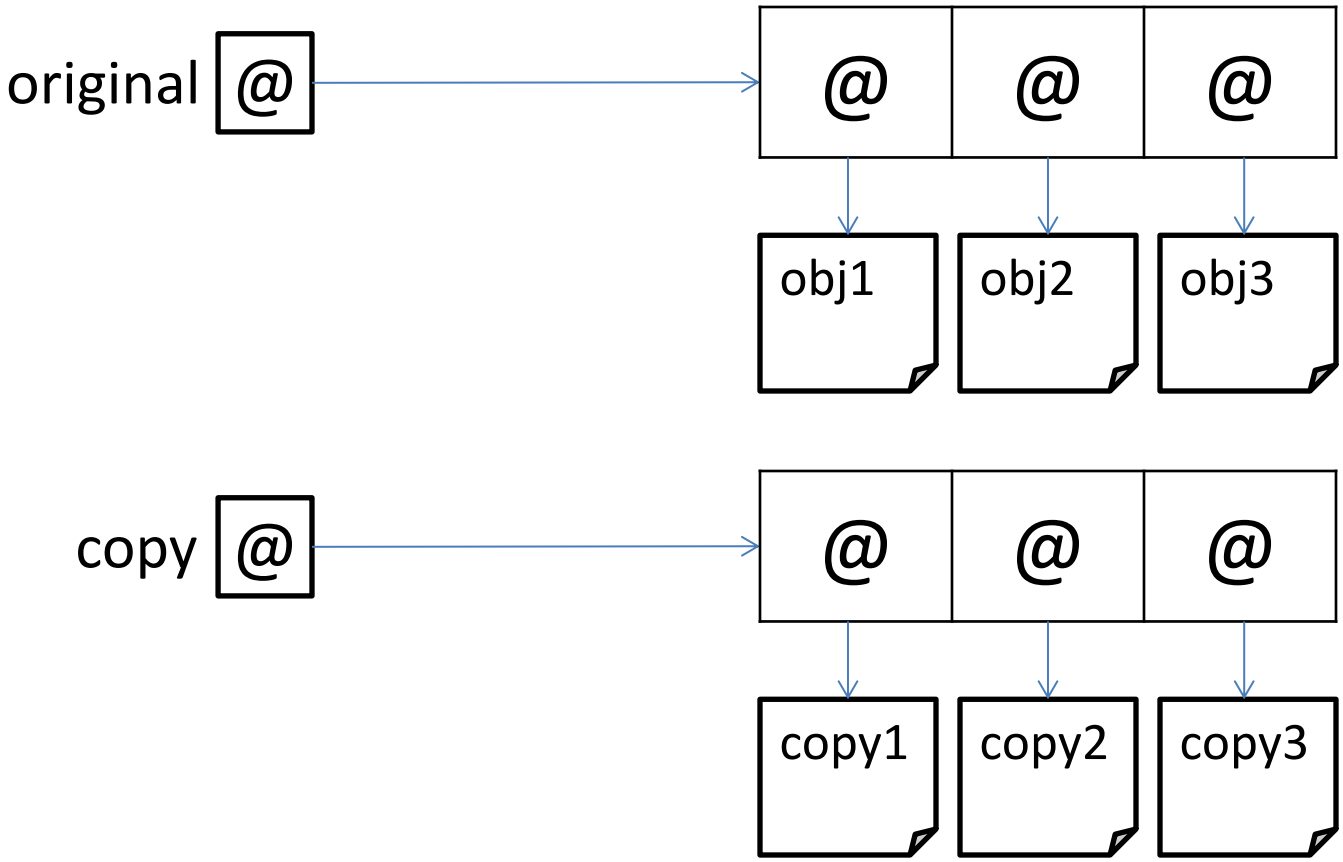
Reference copy



Shallow copy

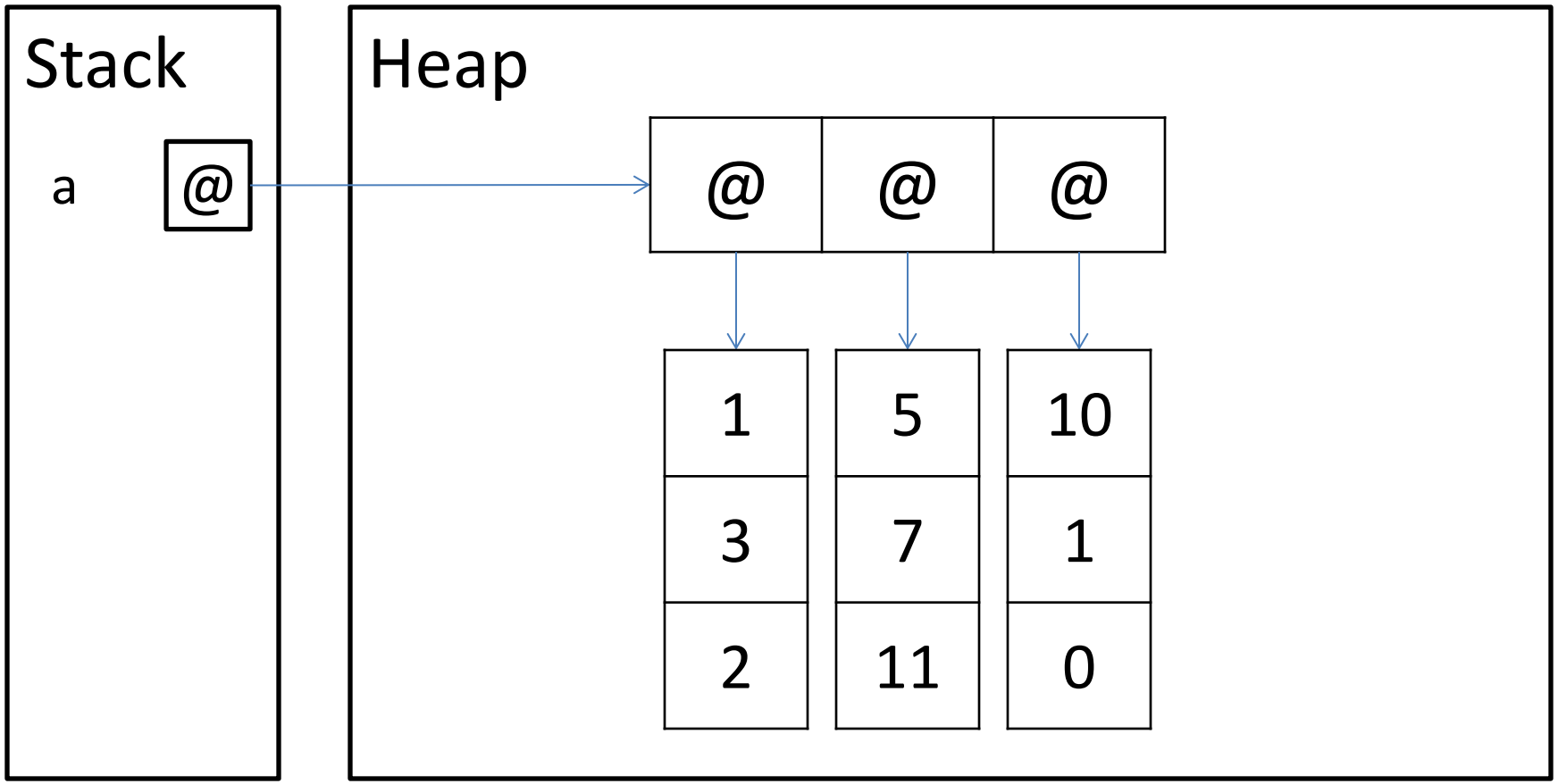


Deep copy



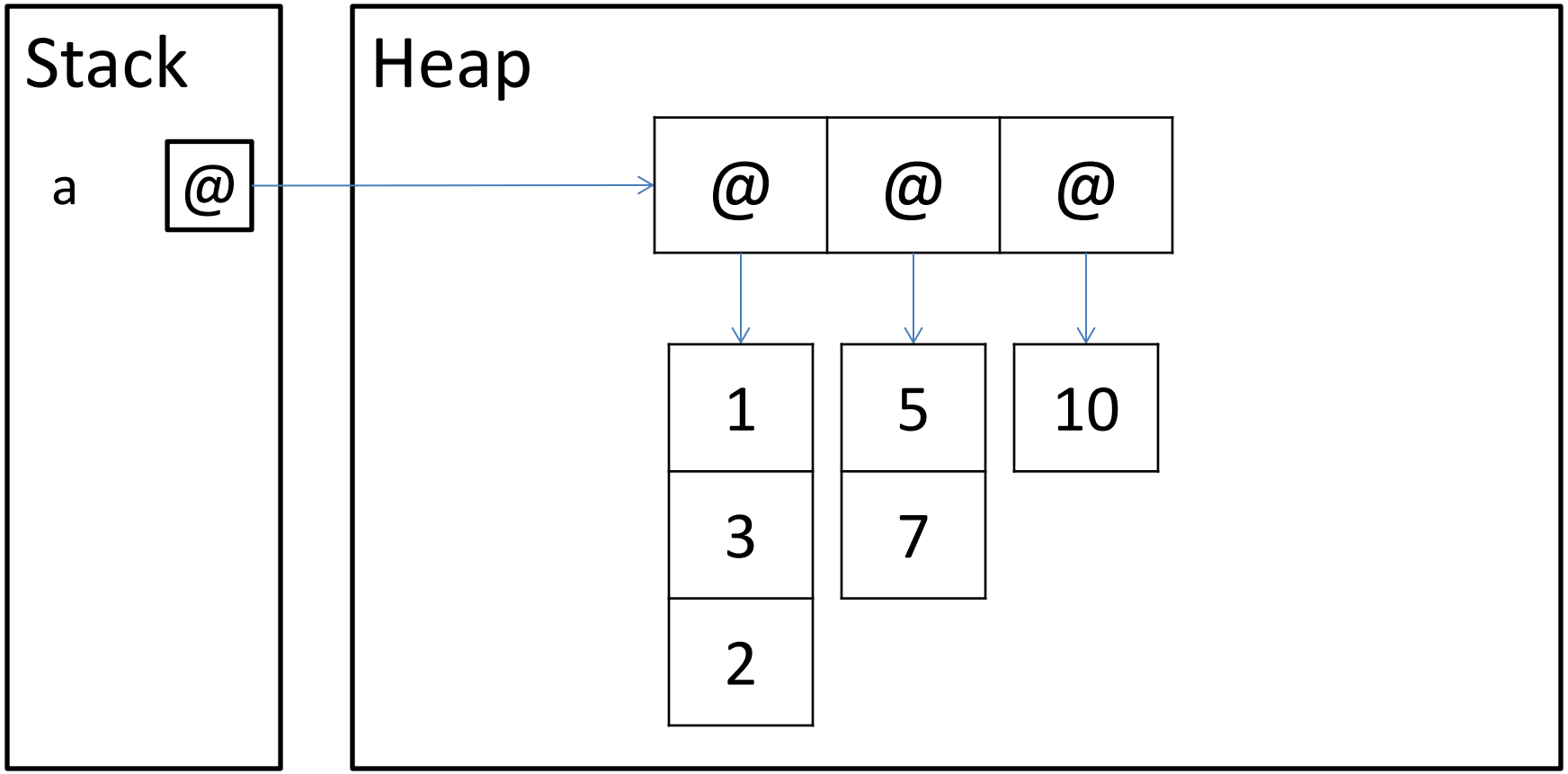
Arrays of arrays (2D)

- Matrix Example



“Ragged” arrays

- Pascal’s triangle example



3D (4D, 5D,…) arrays