The Parts of a Java Program's Memory

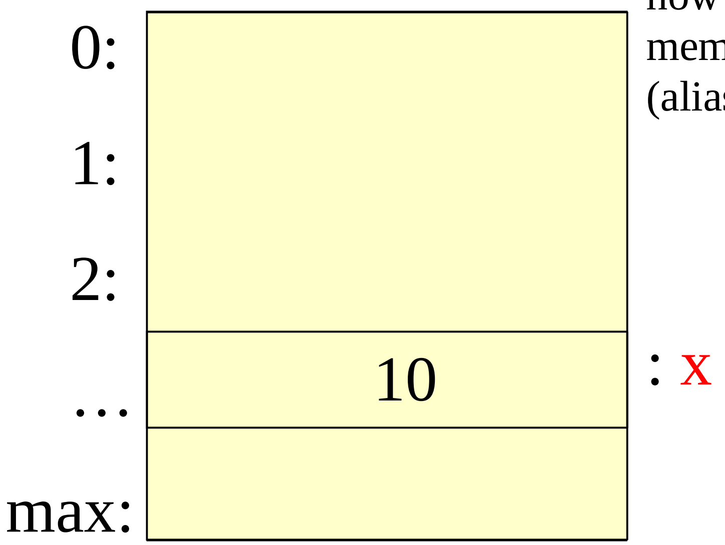
- A Running Program’s memory is a finite expanse of consecutive addressable storage locations, each is 8 bits (1byte) wide
- Address go from 0 to some max
- A program’s code and data are stored in its memory
  - when you declare a variable, a memory location is allocated for it at some address, the variable’s name is how you refer to the memory location in your program:

```
int x;      // allocate memory for an int, and associate the name “x” with it
x = 10;     // store the value 10 at the memory location associated with “x”
```

Address:

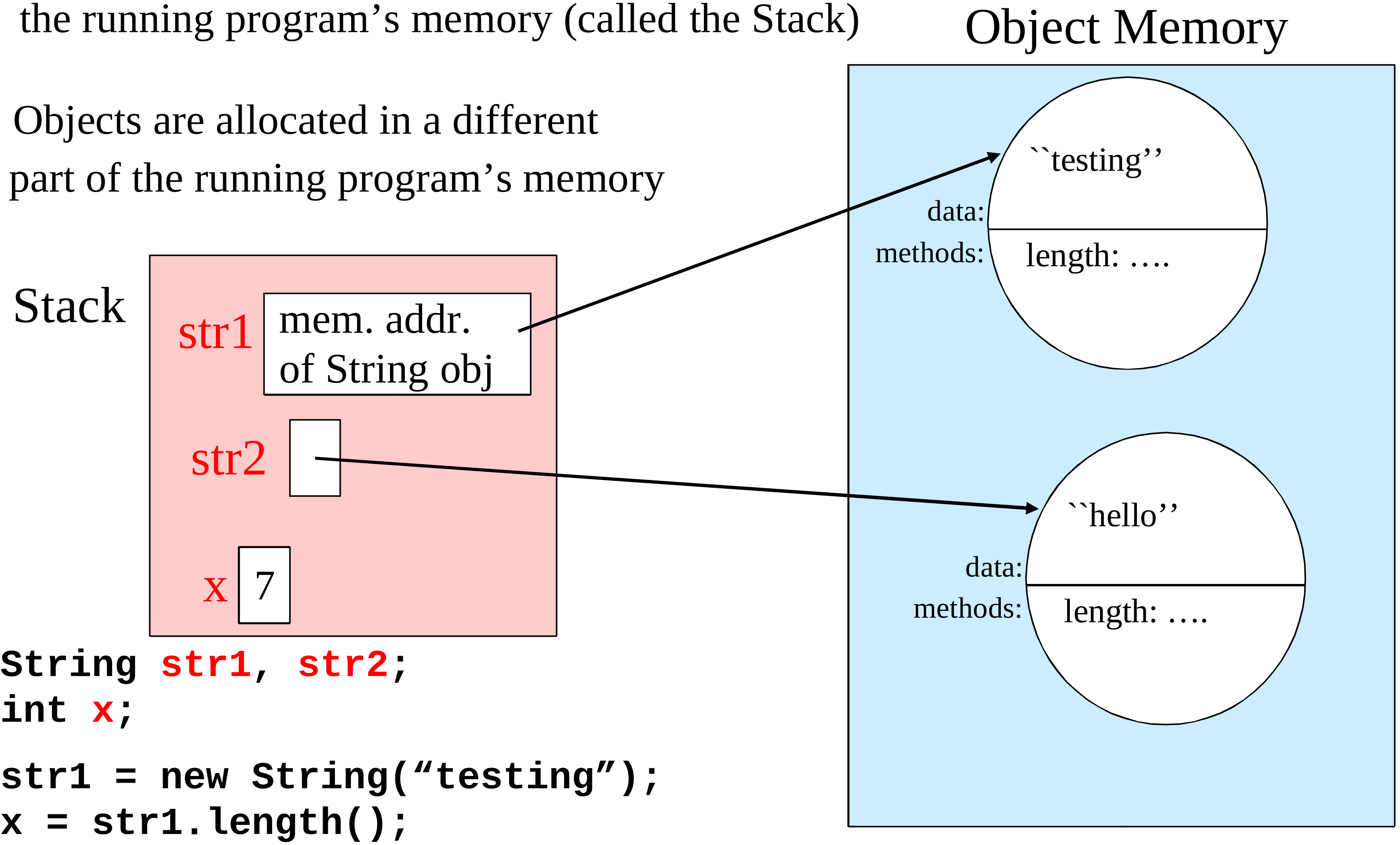
variables:

how my program refers to memory locations (aliases for addresses)



Parts of a Java Program’s memory

- Variables are allocated in one part of the running program’s memory (called the Stack)
- Objects are allocated in a different part of the running program’s memory



Object Memory

Stack

str1 mem. addr. of String obj

str2

x 7

``testing’’

data:

methods:

length: ….

``hello’’

data:

methods:

length: ….

```
String str1, str2;
int x;

str1 = new String(“testing”);
x = str1.length();
str2 = new String(“hello”);
```

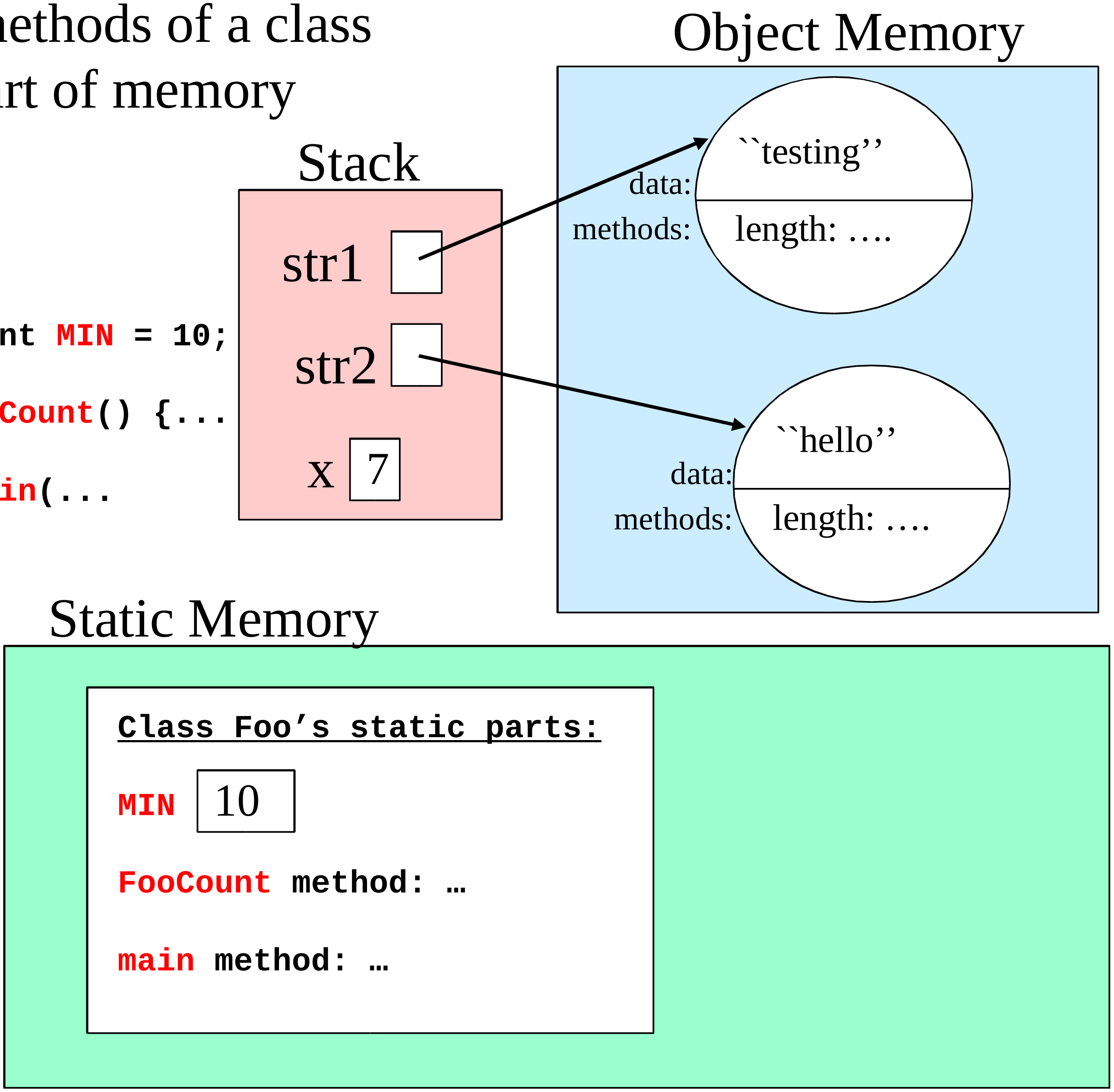
- Static data and methods of a class are in another part of memory

```
Class Foo {

public final static int MIN = 10;

public static int FooCount() {...

public static void main(...
```



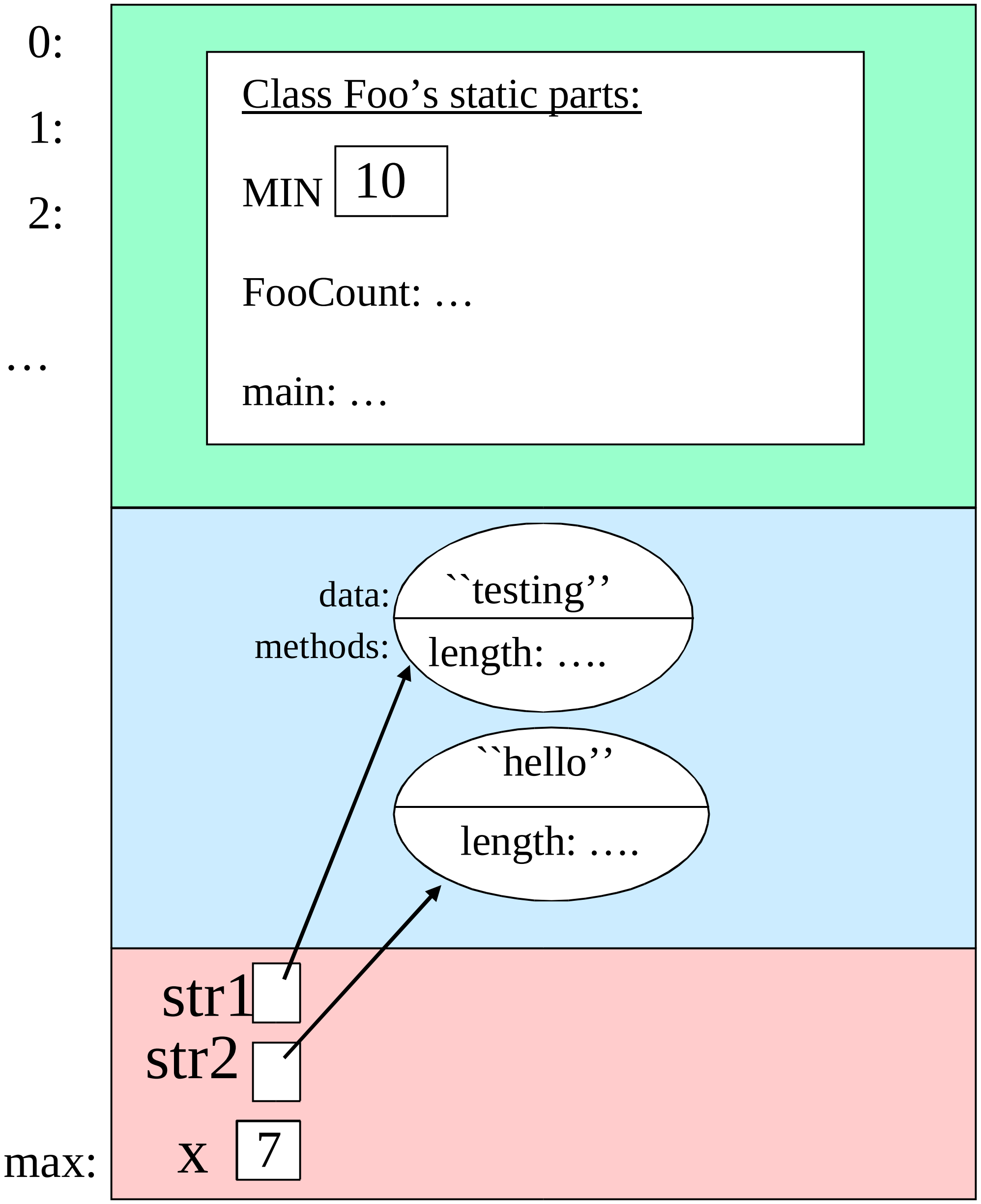
Class Foo's static parts:

MIN 10

FooCount method: …

main method: …

- A Running Program's memory may look like:



0:

1:

2:

…

Class Foo's static parts:

MIN 10

FooCount: …

main: …

data: ``testing''

methods: length: ….

``hello''

length: ….

str1

str2

x 7

max:

Static Memory: static data and methods go here

Object Memory: objects created with **new** go here

Stack: variables and parameters go here