



CSCI 1102 Computer Science 2

Meeting 7: Thursday 2/18/2021

Linked Stacks; Memory

Notes

- First quiz next Thursday 2/25
- A note on Object versus Generics

Generics vs Object

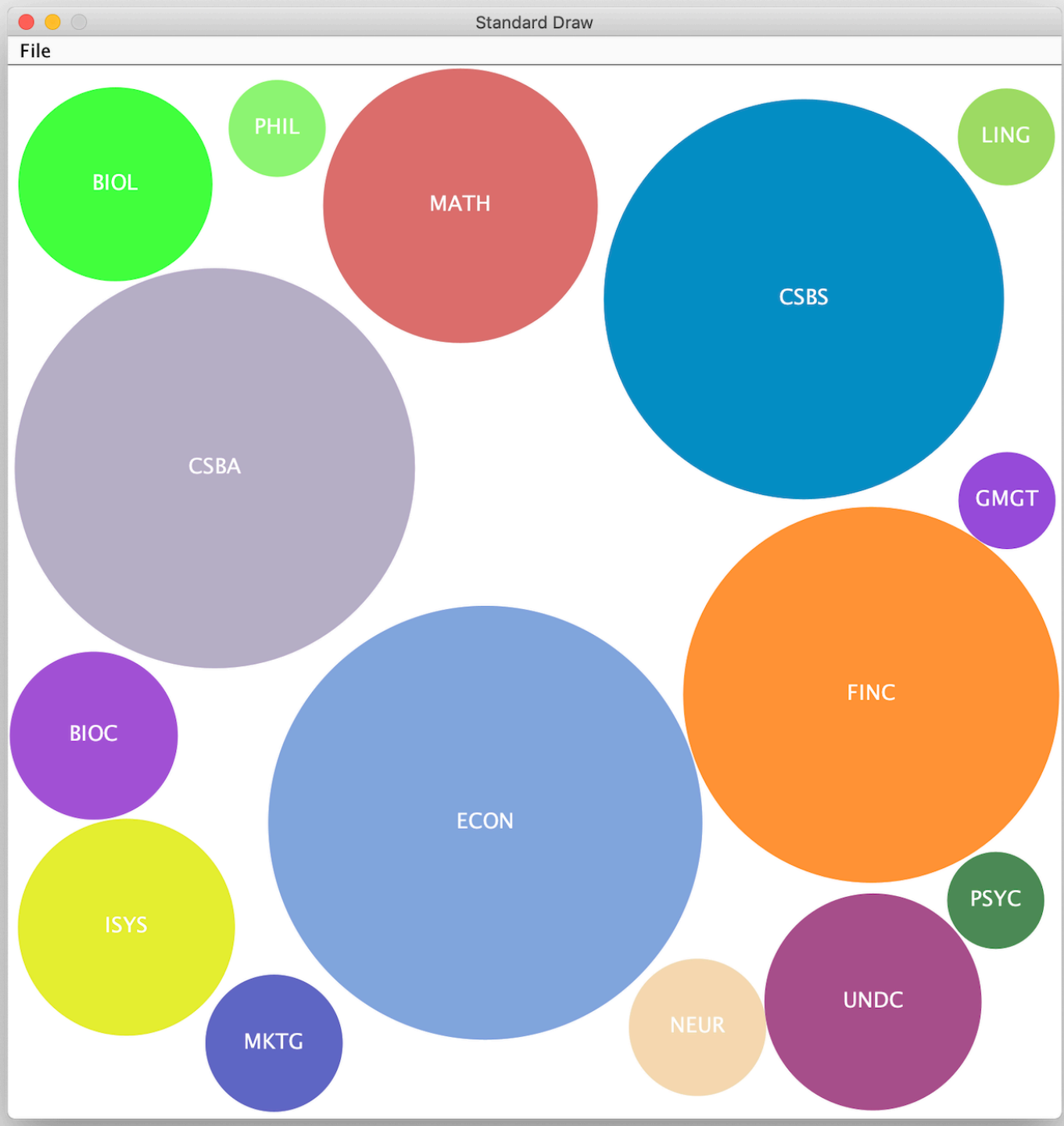
```
62 public static void main(String[] args) {
63
64     // Unit testing
65     ObjectStack myStack = new ObjectStackC();
66
67     myStack.push("Alice");
68     myStack.push("Jose");
69     myStack.push("Bob");
70     myStack.push(23);
71
72     while (!myStack.isEmpty()) {
73         String item = (String) myStack.pop();
74         System.out.format("s\n", item);
75     }
76 }
```

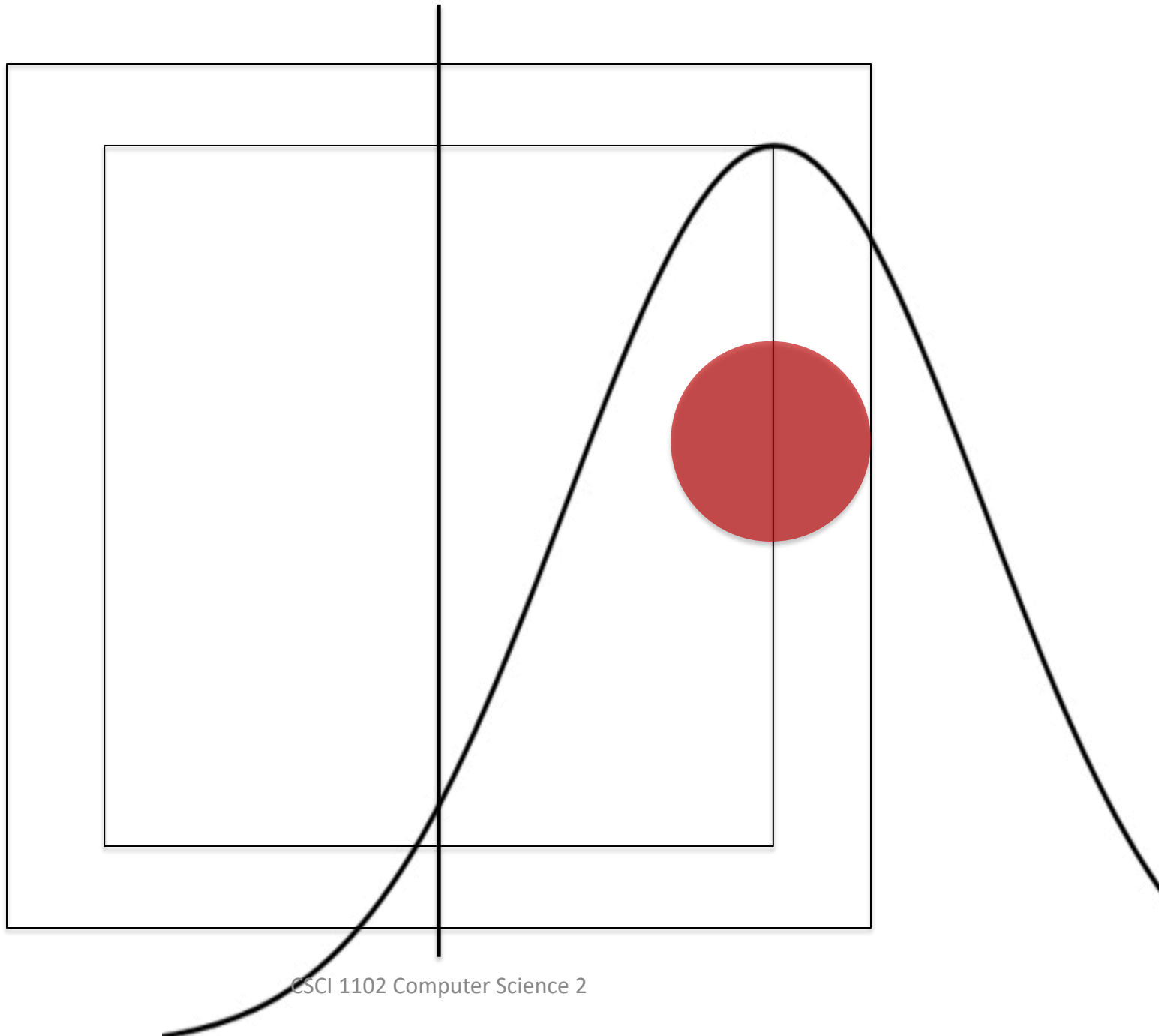
OK to mix, all of
type Object,
successful
compilation

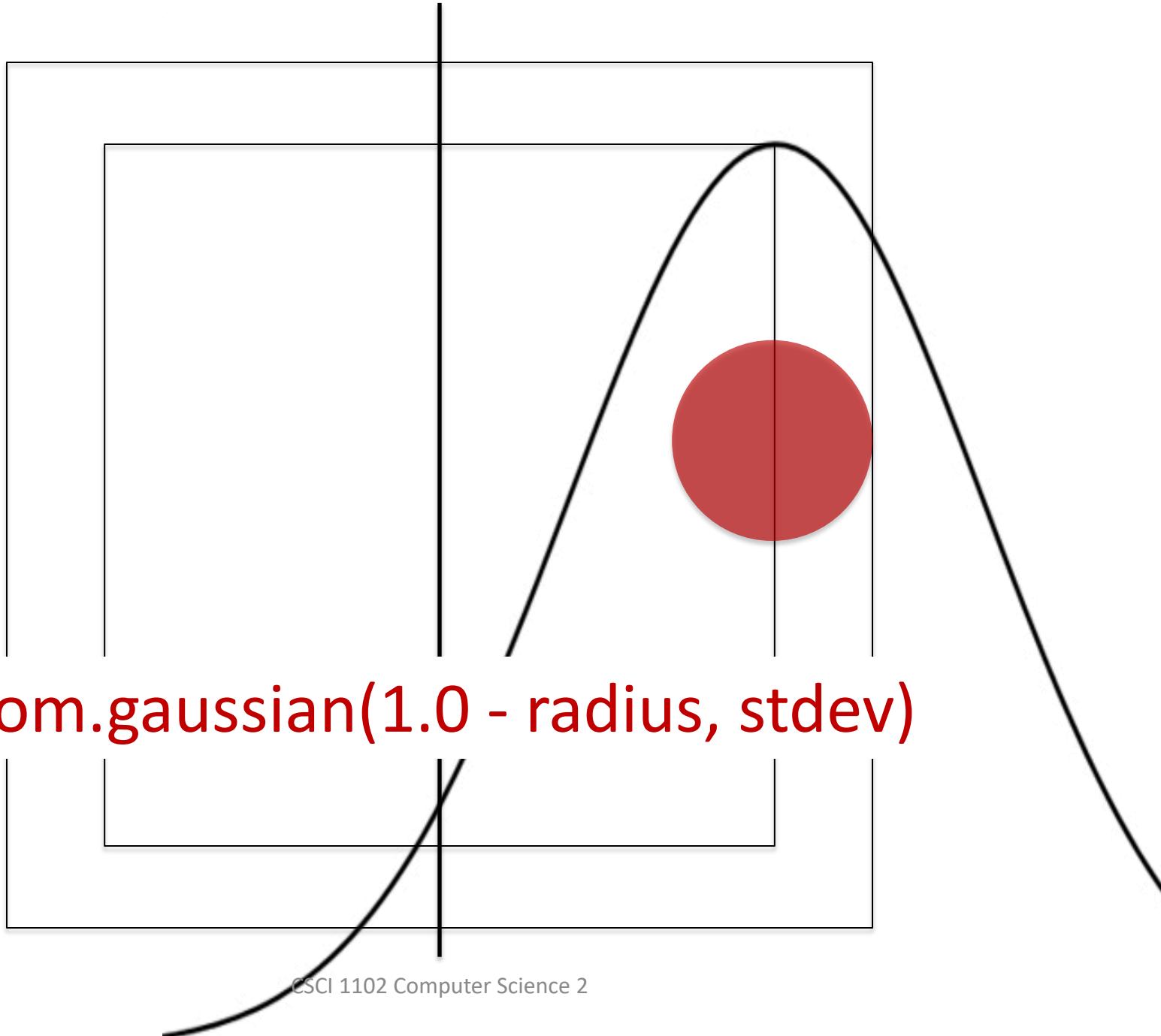
A run-time
cast error


```
157 */
158 public static void main(String[] args) {
159     Stack<String> stack = new ResizingArrayStack<String>();
160     while (!StdIn.isEmpty()) {
161         String item = StdIn.readString();
162         stack.push(item);
163     }
164     stack.push(12);
165     while (!stack.isEmpty())
166         System.out.format("%s\n", stack.pop());
167 }
168 }
169
```

Mixing not allowed,
a compile-time
error







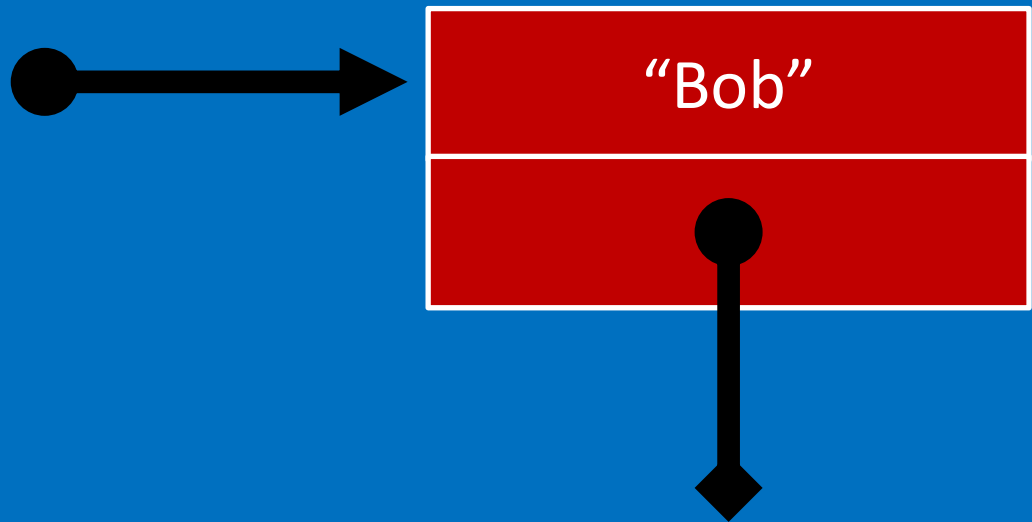
`StdRandom.gaussian(1.0 - radius, stdev)`

Linked Stacks

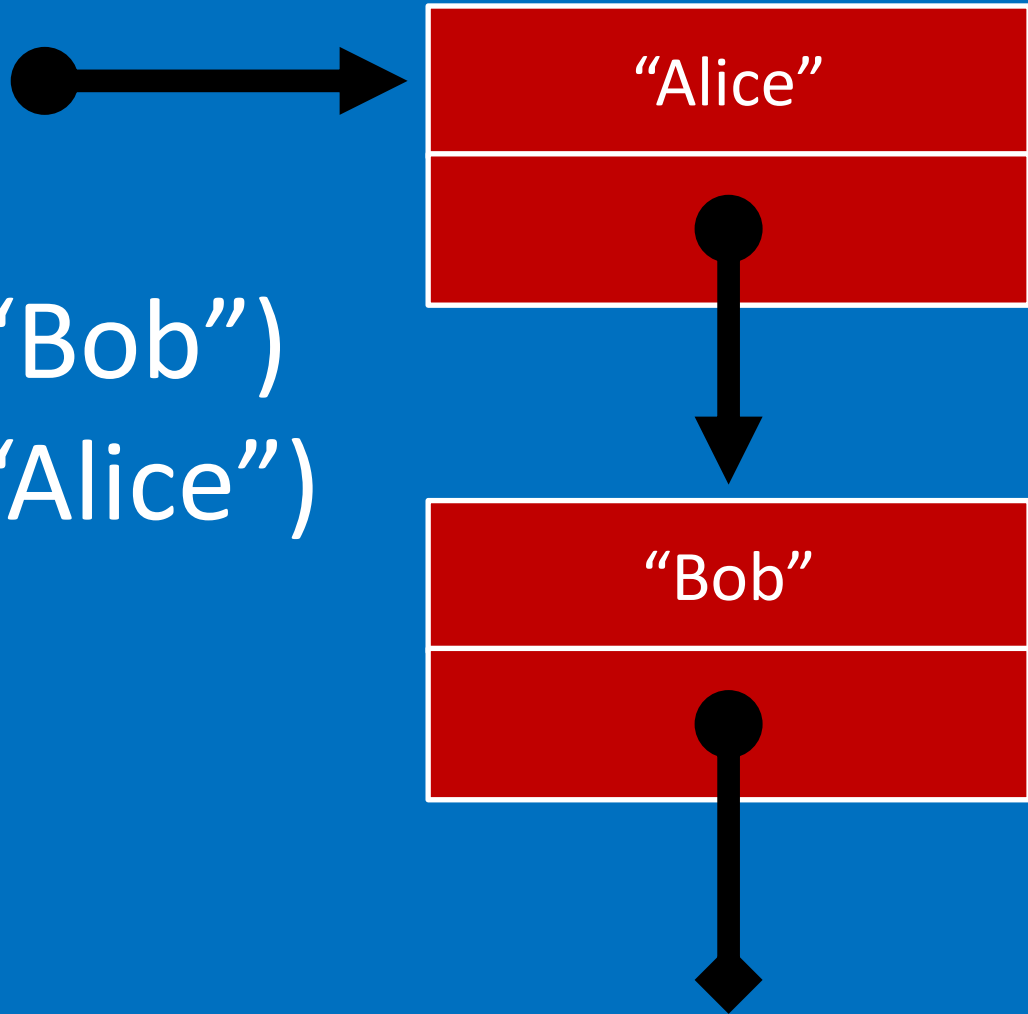
Linked Stacks

- Pro: unlimited size with no linear resizing cost;
- Con: Poor locality

`stack.push("Bob")`



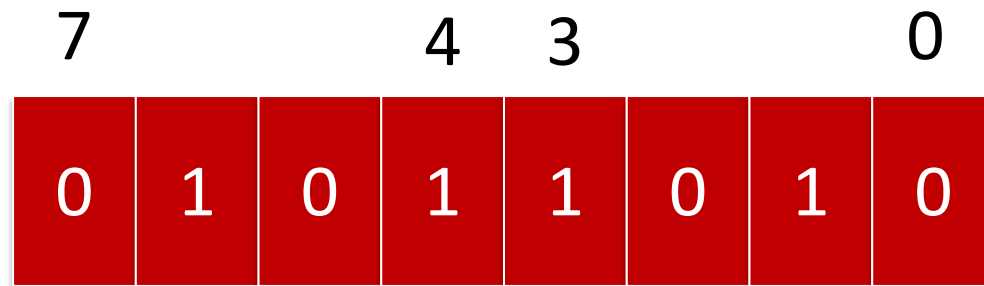
```
stack.push("Bob")  
stack.push("Alice")
```



Code

Memory

Basic unit of storage is the **byte**



There are $2^8 = 256$ 8-bit patterns

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

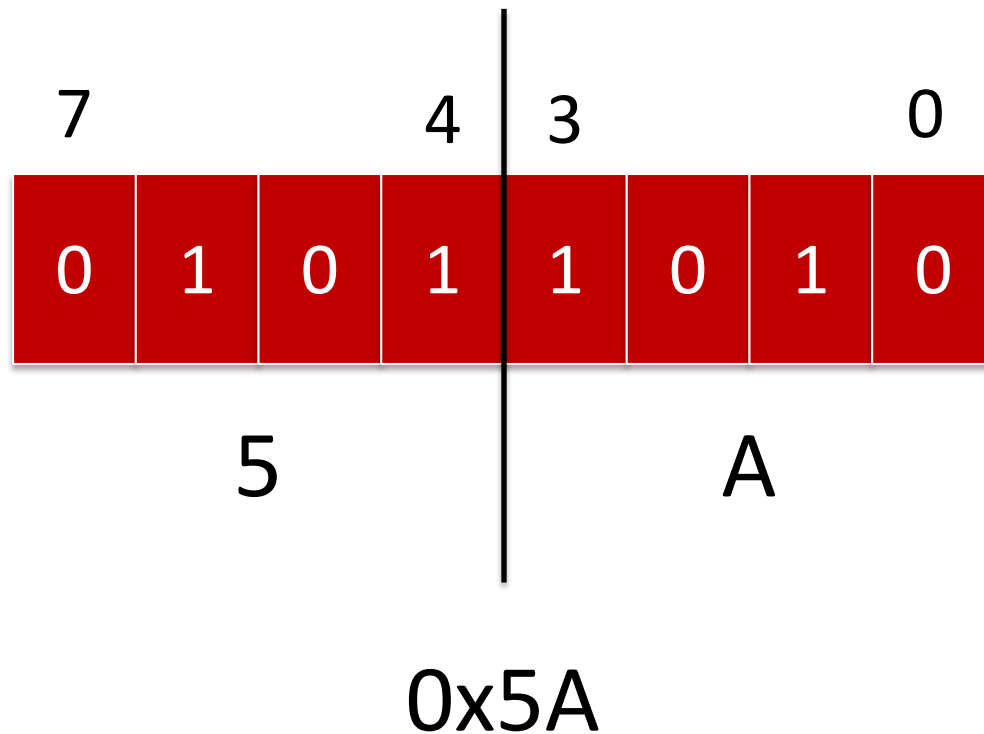
...

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

We can abbreviate 4 bits with one hexadecimal (base 16) digit

Binary	Hex	Dec	Binary	Hex	Dec
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	A	10
0011	3	3	1011	B	11
0100	4	4	1100	C	12
0101	5	5	1101	D	13
0110	6	6	1110	E	14
0111	7	7	1111	F	15

Abbreviate an 8-bit byte with a pair of hex digits



Representing Numbers

- Non-negative integers: use binary positional numerals
- All integers: use two's complement:

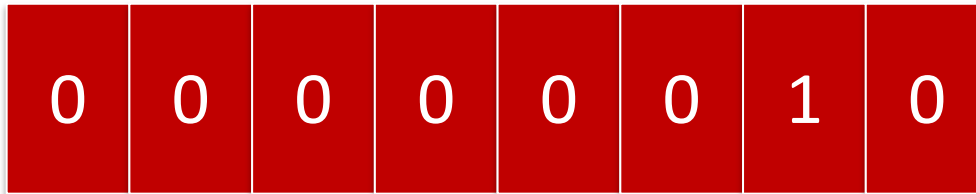


bit 1 => negative, 0 => non-negative

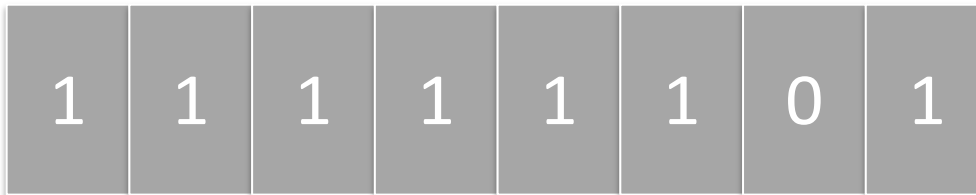
Two's Complement (8 bits)

Modus Operandi: to negate, flip all bits and add 1

0x02

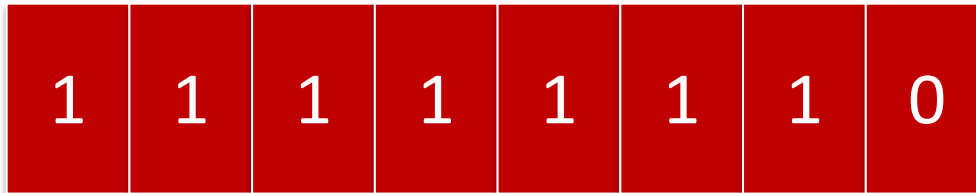


= 2



flip

0xFE



= -2

Representing Numbers

- 8 bits unsigned: $2^8 = 256$ values 0 .. 255
- 8 bits signed: $2^8 = 256$ values -128 .. 127
- Floating point: IEEE 754

$$3.14159 = 314159 \times 10^{-4}$$

Bitwise Operators

- Shifting:
 - $m \ll n$: shift m left n bits;
 - $m \gg n$: shift m right n bits, NB sign extension
- Logic: $\&$, $|$, \wedge (xor), \sim (flip)
- Seeing what is happening:

`String.format("%x", expr)`