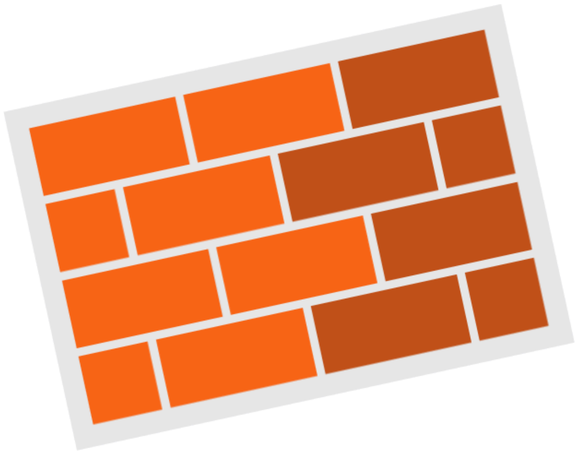
Primitive Data Types

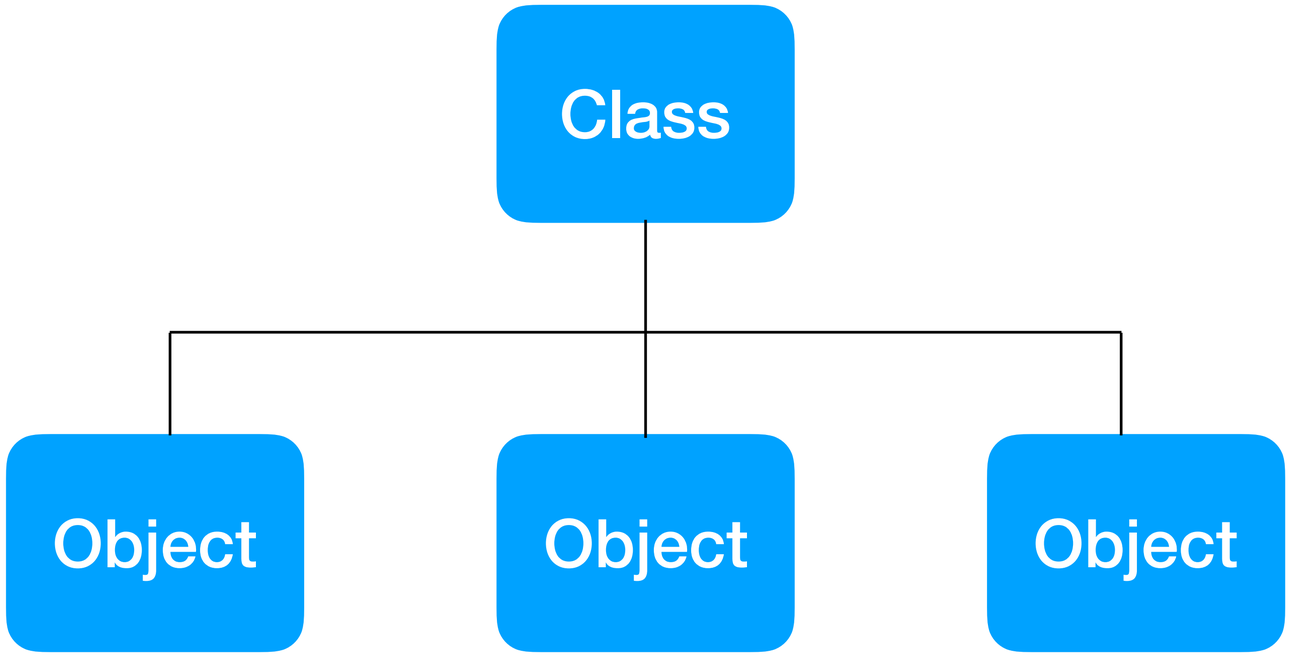
Built into the language

Foundation of all other types



Four categories of primitive types

- Integer
- Floating point
- Character
- Boolean

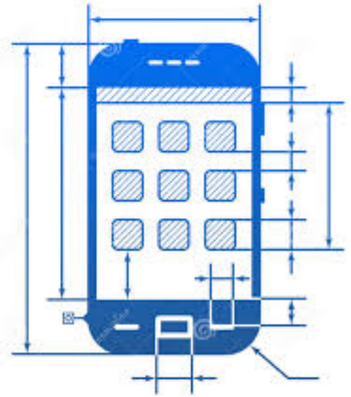


Class is a blueprint for an object

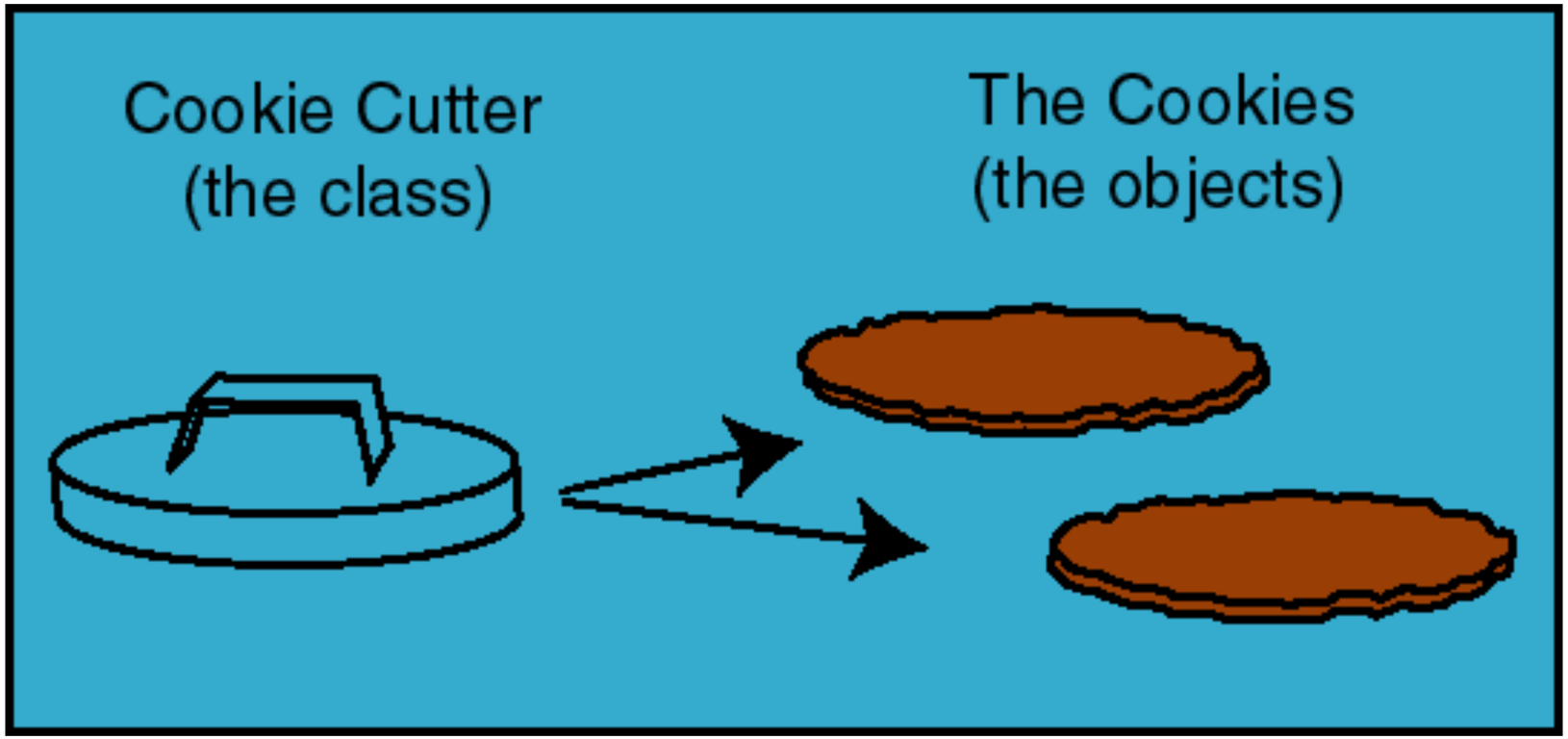
From a single class we can create multiple objects

phone blueprint

class

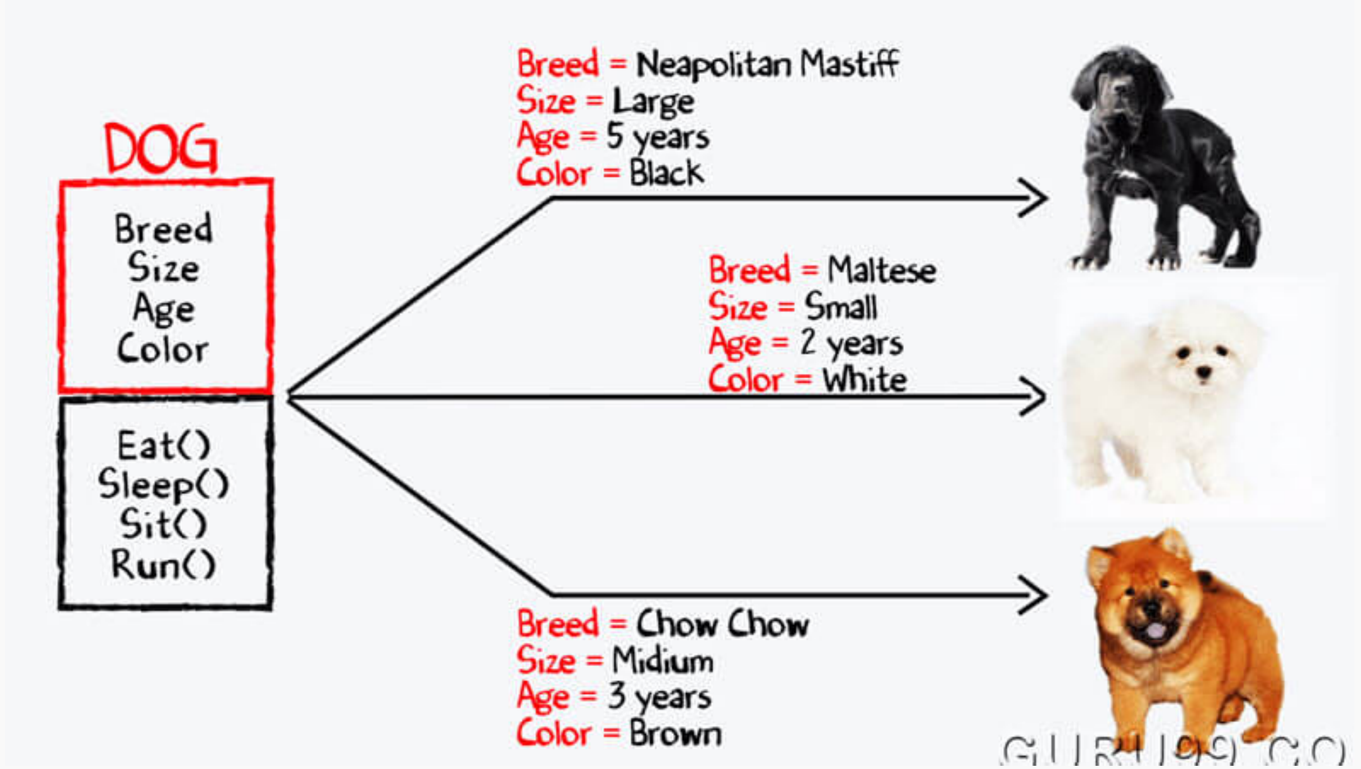


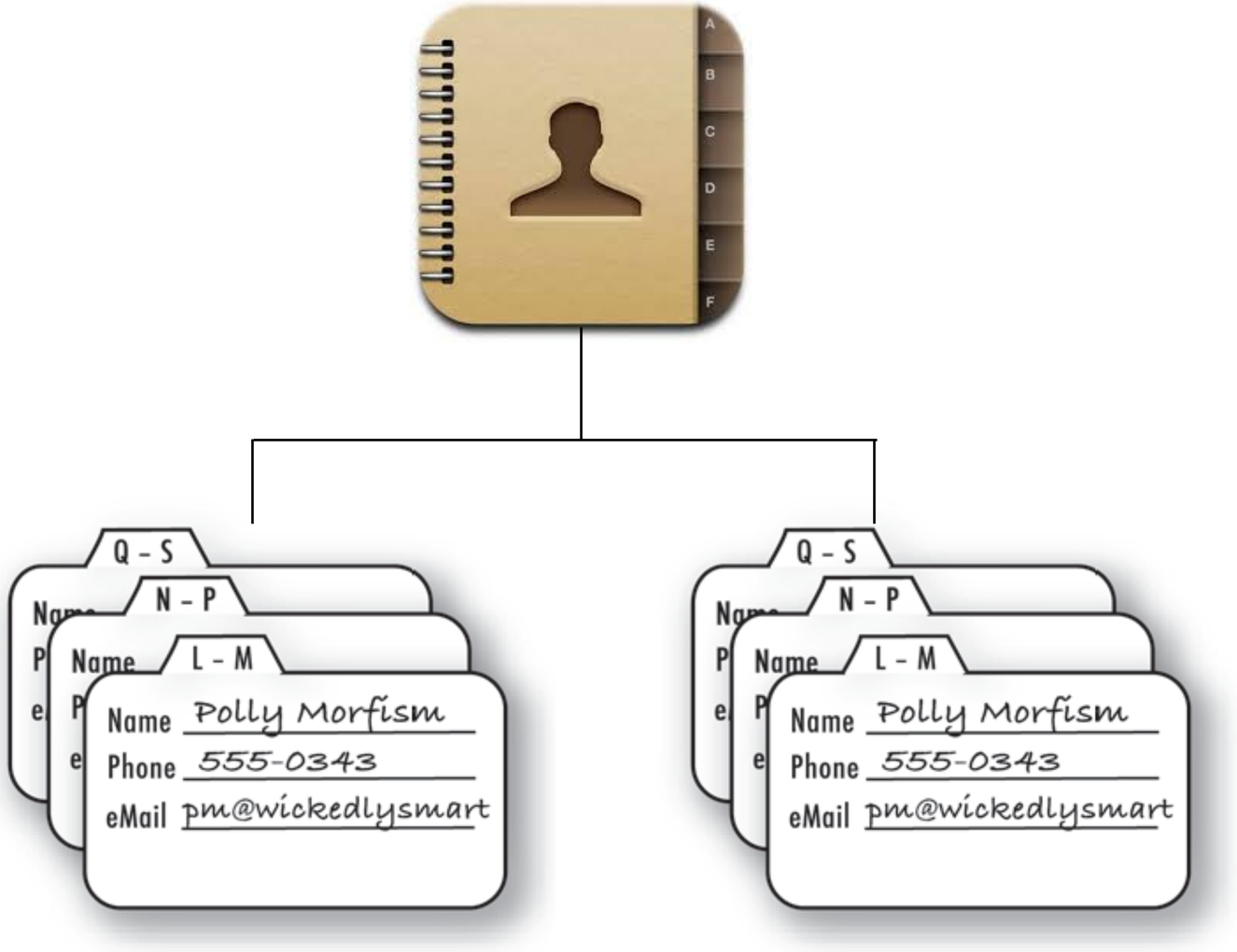
phone objects


Cookie Cutter
(the class)
The Cookies
(the objects)

Class has 2 things:

1. Data → `instance variables`
2. Behaviour → `methods`

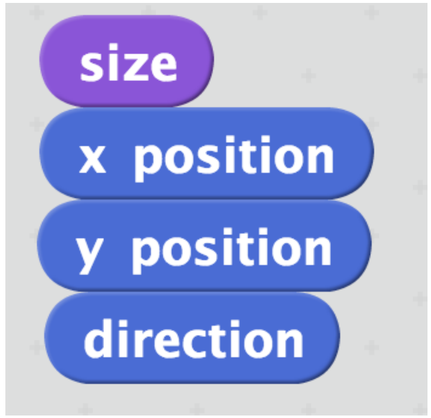



A
B
C
D
E
F
Q - S
N - P
L - M
Name
Phone
eMail
Polly Morfism
555-0343
pm@wickedlysmart

Sprite class from scratch programming

methods

variables



Student

FirstName
LastName
Email
Age
Address
Course

Apply
Code
AttendClass
MissClass
Practice
BeLazy
DoHomework

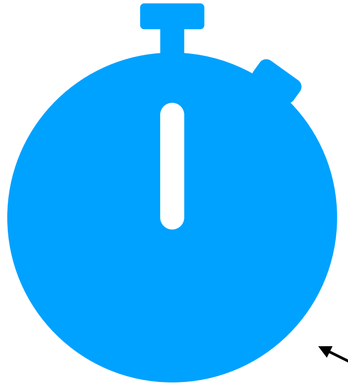
public class Student

```
String firstName;
String lastName;
String email;
String age;
String address;
int batchNum;
```

```
void apply(){}
void code(){}
void attendClass(){}
void missClass(){}
void practice(){}
void beLazy(){}
void doHomework(){}
```

CLASS

OBJECTS



WATCH:

attributes:

brand, style, color, material, price, type

Basic behavior:

Tick, show time, showDate

CYBERTEK

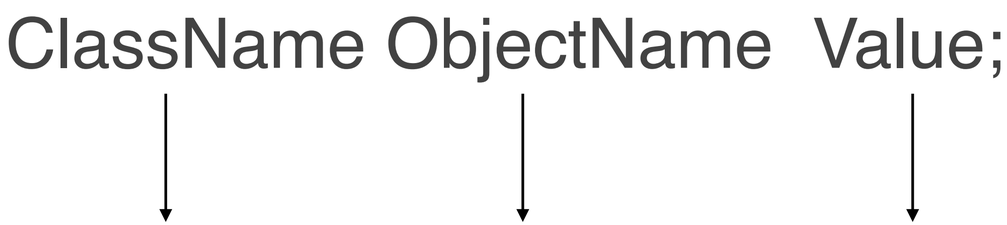
PRIMITIVE	OBJECT CLASS TYPE

Single piece of data:
 	
`byte floor=3;`
`char letter='a';`
	Can have multiple pieces of data:
	
	`Student student=new Student();`
	`student.age=33;`
	`student.course="Java";`

Cannot perform any actions on the data	Can perform actions on the data using methods: `student.submitAssignment();`

String class

How to create a String object

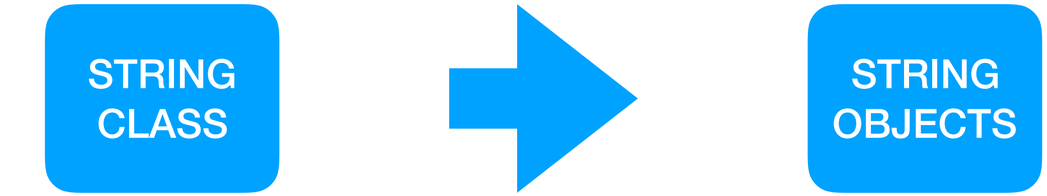



```
String fruit="apple";
```

new

Used to create an object from a class

```
String language = new String(“Java”);
String language = “Java”;
Student mark = new Student();
Account debit = new Account();
```

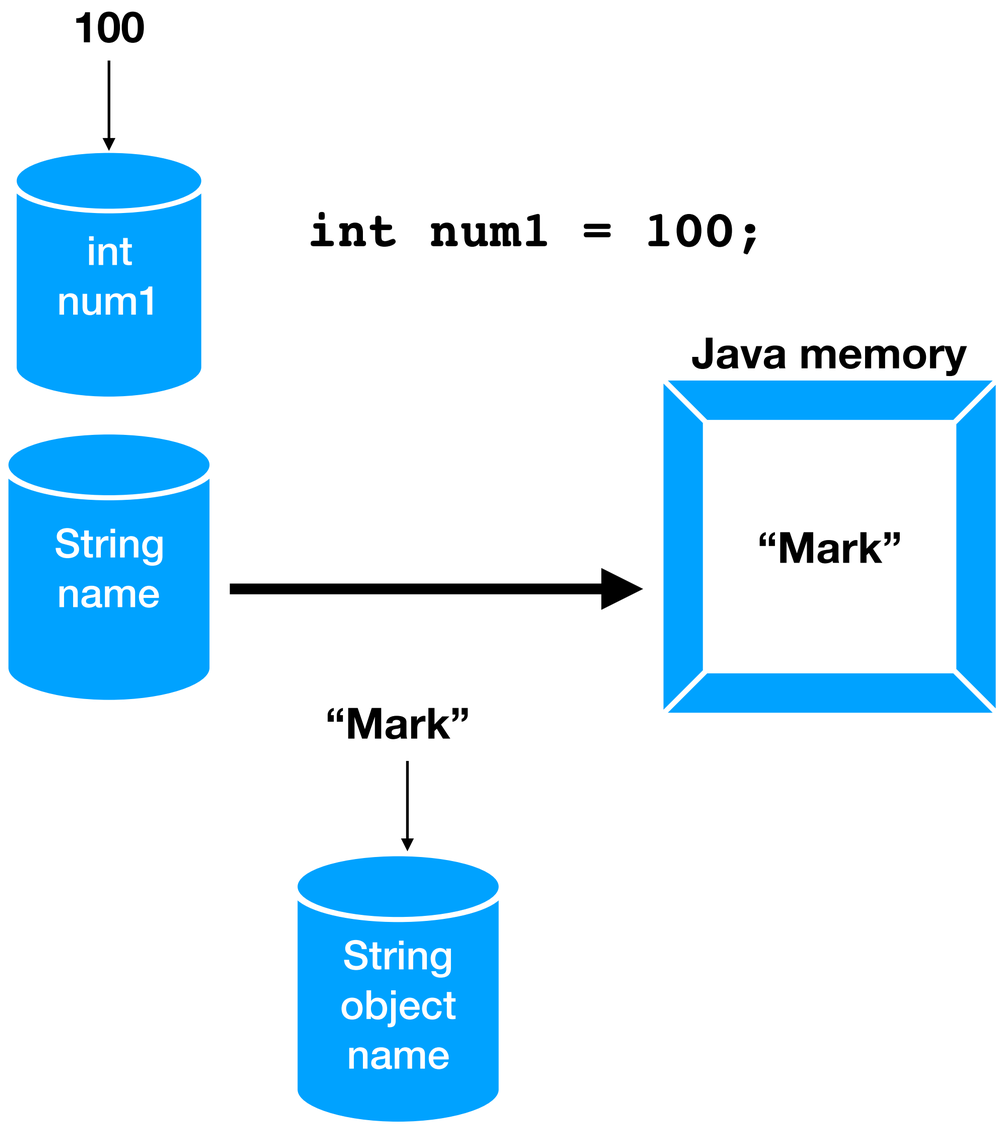


String

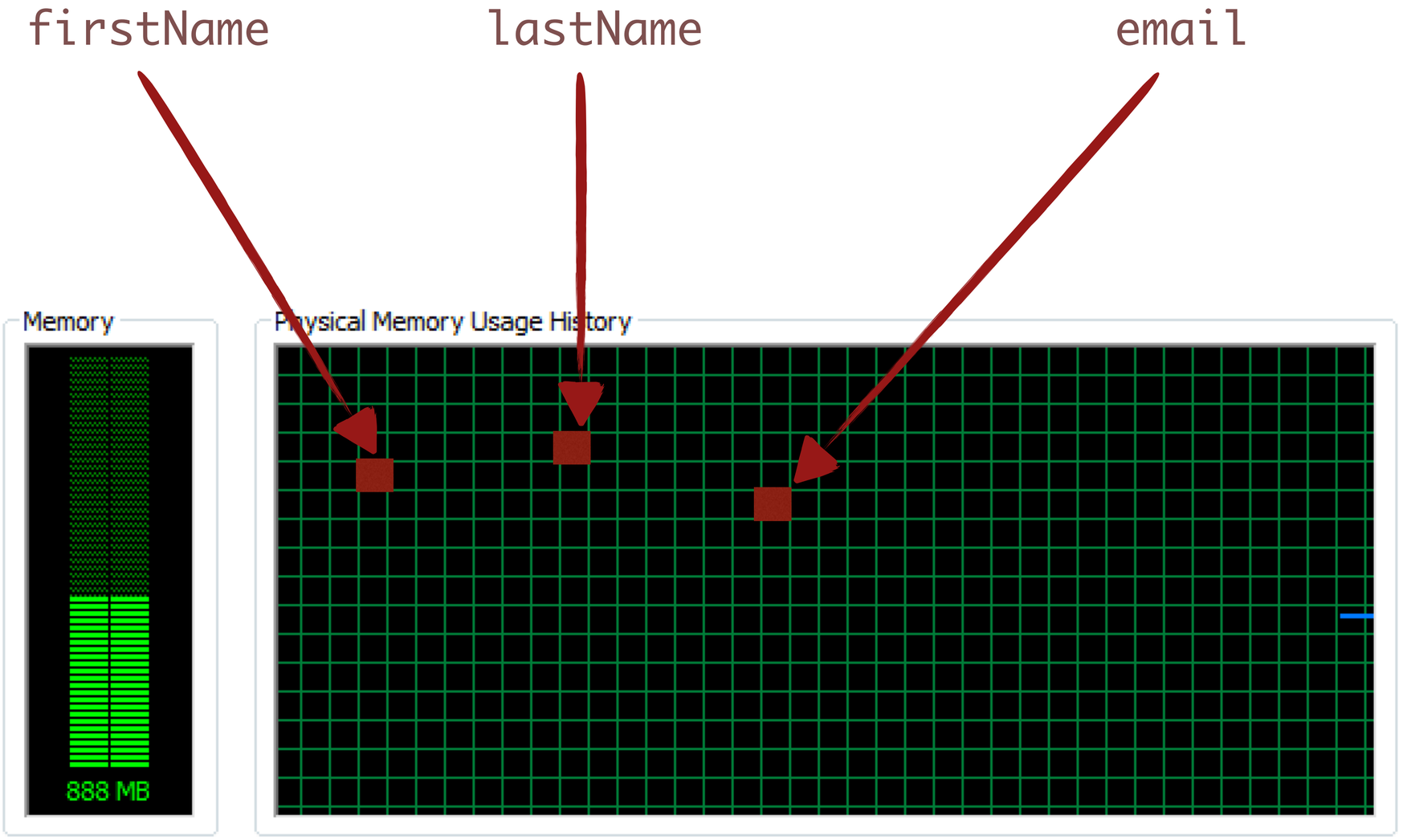
```
String firstName=new String(“Mark");

String lastName=“Smith”;
int num = 100;


String email=firstName+lastName+"@gmail.com";
```


100
int
num1
int num1 = 100;
Java memory
String
name
“Mark”
“Mark”
String
object
name

Each object has its own place in memory



String Methods

Advantage of String class: many built-in methods for String manipulation

```
str.length();          // get length of string
str.toLowerCase()      // convert to lower case
str.toUpperCase()      // convert to upper case
str.charAt(i)          // what is at character i?
str.contains(..)       // String contains another string?
str.startsWith(..)     // String starts with some prefix?
str.indexOf(..)        // what is the position of a character?
....many more
```

stringObject.methodName()

- concat(String str) : String - String
- contains(CharSequence s) : boolean - String
- contentEquals(CharSequence cs) : boolean - String
- contentEquals(StringBuffer sb) : boolean - String
- endsWith(String suffix) : boolean - String
- equals(Object anObject) : boolean - String
- equalsIgnoreCase(String anotherString) : boolean - String
- getBytes() : byte[] - String
- getBytes(Charset charset) : byte[] - String
- getBytes(String charsetName) : byte[] - String
- getBytes(int srcBegin, int srcEnd, byte[] dst, int dstBegin
- getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin
- getClass() : Class<?> - Object
- hashCode() : int - String
- indexOf(int ch) : int - String
- indexOf(String str) : int - String
- indexOf(int ch, int fromIndex) : int - String
- indexOf(String str, int fromIndex) : int - String
- intern() : String - String
- isEmpty() : boolean - String
- lastIndexOf(int ch) : int - String
- lastIndexOf(String str) : int - String
- lastIndexOf(int ch, int fromIndex) : int - String
- lastIndexOf(String str, int fromIndex) : int - String
- length() : int - String
- matches(String regex) : boolean - String
- notify() : void - Object
- notifyAll() : void - Object

CYBERTEK

```
String str = “hello”;

‘h’ ‘e’ ‘l’ ‘l’ ‘o’
 0   1   2   3   4
```

Internally it is a combination of chars.

```
String city = new String(“Baku”);

‘B’ + ‘a’ + ‘k’ + ‘u’
 0     1     2     3
```

```
stringObject.methodName();
```

primitive/object

```
primitive/object = stringObject.methodName();
```

```
String country =“Belize";
int size =country.length();
```

length();

"Belize".length();

6

toUpperCase()

```
String country = new String();
     country = "Belize";
     country.toUpperCase();
```

```
BELIZE
```

toLowerCase()

```
String country = "Belize";
     country.toLowerCase();
```

```
belize
```

startsWith(String s)

```
String country =  new String("Belize");
     country.startsWith("B");
```

---

true

endsWith(String s)

```
String country =  new String("Belize");
     country.endsWith("e");
```

```
true
```

contains(String s)

```
String country ="Belize is a country";
     country.contains("is");
```

---

true

Size/Length

```
"Java"
 1234
```

Index

"Java"
0123

charAt(int index)

```
String country ="Belize";
     country.charAt(0);
```

B

indexOf(String str)

```
String country ="Belize";
     country.indexOf("i");
```

3

substring(int beginning)

```
String country ="Belize";
     country.substring(2);
```

```
lize
```

substring(int beginning, int ending)

```
String country ="Belize";
     country.substring(2, 5);
```

```
liz
```

trim()

```
String country =“  Belize  ";
     country.trim();
```

“Belize”

trim()

```
String country =“  Belize  ";
     country.trim();
```

“Belize”