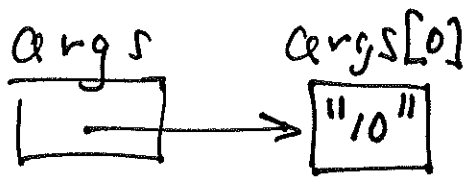


CNPS 12A 5-11-10

Re Pec 4:

% Java Prime 10

inside main(): args.length will be 1



```

    ...
    int n;
    ...
    n = Integer.parseInt(args[0]);
    ...
  
```

Suggest put this
in try block &
catch

NumberFormatException

note: to check if a is div. by b do $a \% b$ and compare to 0.

Ex. Java Prime 5

In for main()

	0	1	2	3	4
Primes	2	3	5	7	11

$isPrime(3, Primes): 2 > \sqrt{3} \therefore true$

$isPrime(4, Primes): 2 | 4 \therefore false$

$isPrime(5,) : 2 \nmid 5$

$3 > \sqrt{5} \therefore true$

$isPrime(6,) : 2 | 6 \therefore false$

$$\text{isPrime}(7,) : 2 \nmid 7$$

3

$$3 > \sqrt{7} \therefore \text{true}$$

$$\text{isPrime}(8,) : 2 \mid 8 \quad \text{false}$$

$$\text{isPrime}(9,) : 2 \nmid 9$$

$$3 \mid 9 \quad \text{false}$$

$$\text{isPrime}(10,) : 2 \mid 10$$

$$\text{isPrime}(11,) : 2 \nmid 11$$

$$3 \nmid 11$$

$$5 > \sqrt{11} \quad \text{true}$$

chap 6 : Objects & classes

[4]

A class is a new data type that bundles data values together with methods that operate on that type.

Ex.

```
class empty { }
```

what may a class contain?

- member variables (i.e. fields)
- member methods
- other classes

classes are meant to model
Some concept.

5

Ex. class Person

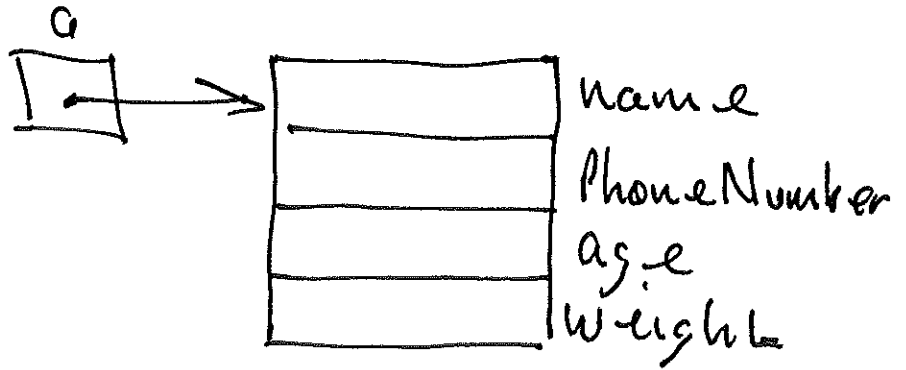
```
String name;  
String phoneNumber;  
int age;  
double weight;  
}
```

has no main(), so is not a
program. This is a new data
type.

Person a;



a = new Person();

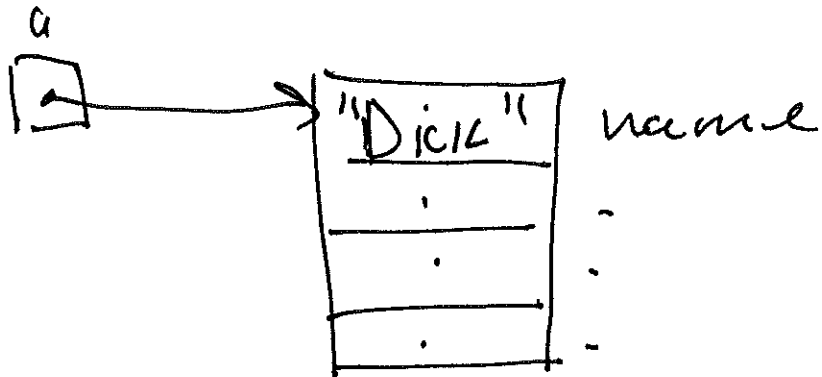


a.name = "Dick" ;

a.phoneNumber = ;

a.age = ;

a.weight = ;



note dot "." in a.age is
called member access operator.

Similar to [] operator on
an array.

Actually class Person
does have one method:

the constructor: Person()

called by the new operator.

Constructor tells new how much
memory to allocate.

To Print out a "Person"

```
class PersonTest {
    ----- main() }
    :
    printPerson(a);
    printPerson(b);
}
```

```
static void printPerson(Person x) {
    System.out.println("name: " + x.name);
    System.out.println("Phone number: " + );
    System.out.println("Age: " + x.age);
    ----- ;
}
```

```
}
```


(9)

we could also place this (static) method in class Person.

to call a static method

ClassName.MethodName (... args ...)


member access op.

we could also implement this as an instance method.

Recall:

(10)

• class methods belong to the class itself.

use static

• instance methods belong to an individual instance of that class.

don't use static.

to call an instance method

`instance_of_class.methodname(args...)`

the general Declaration & Allocation
Statement looks like:

```
myClass x = new myClass();
```

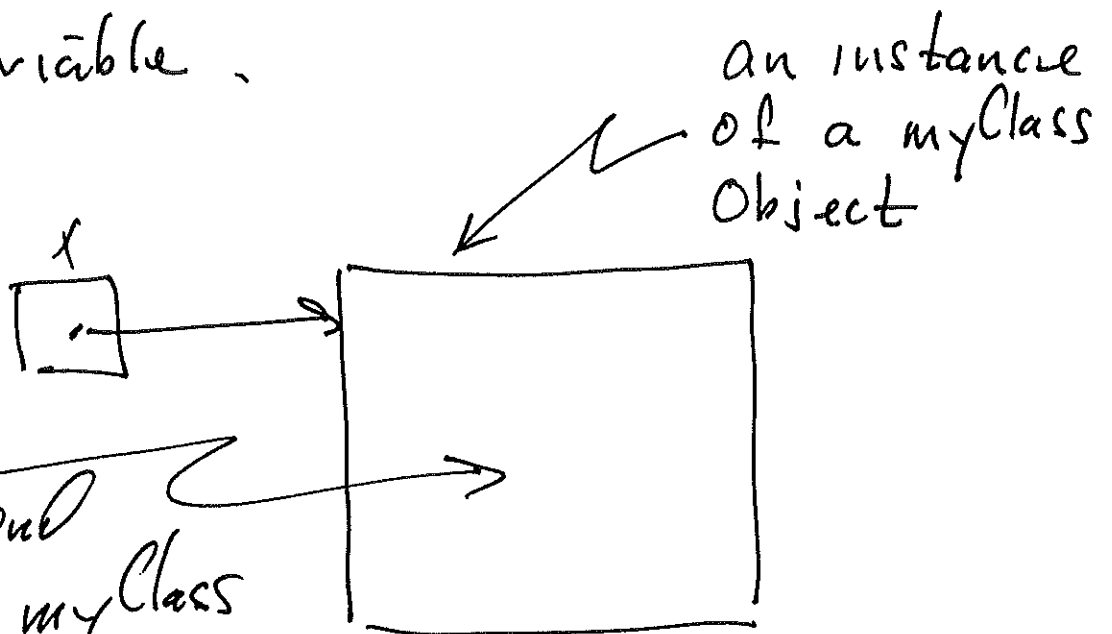
↑
name of
class

↑
variable of
type myClass
necessarily a
reference
variable.

↑
new of

↑
constructor for
the myClass
class.

Result:



Details depend
on defn of myClass

Ex. define a constructor

```
class blah {
```

```
    // fields
```

```
    int happy;
```

```
    int sad;
```

```
    // constructor
```

```
    blah (int a, int b)
```

```
        happy = a;
```

```
        sad = b;
```

```
    }
```

```
    // instance method
```

```
    int getHappy() {
```

```
        return this.happy;
```

```
    }
```

```
}
```

In some method in some class

13

```
blah x = new blah(7, -12);
```

```
System.out.println(x.getHappy());
```