CMPSC 100

Computational Expression

Data Type	Size	Min value	Max Value
byte	1 byte	-128	127
short	2 bytes	-32,768	32,767
int	4 bytes	-2,147,483,648	2,147,483,647
long	8 bytes	- a lot	+ a lot
float	4 bytes	7 decimals	7 decimals
double	8 bytes	15 decimals	15 decimals
char	2 bytes	0	65,536
boolean	(not important)	0 (true)	1 (false)

“primitive” data types

Data Type	Size	Min value	Max Value
String	Various	?	?
Scanner	Various	?	?

“reference” data type

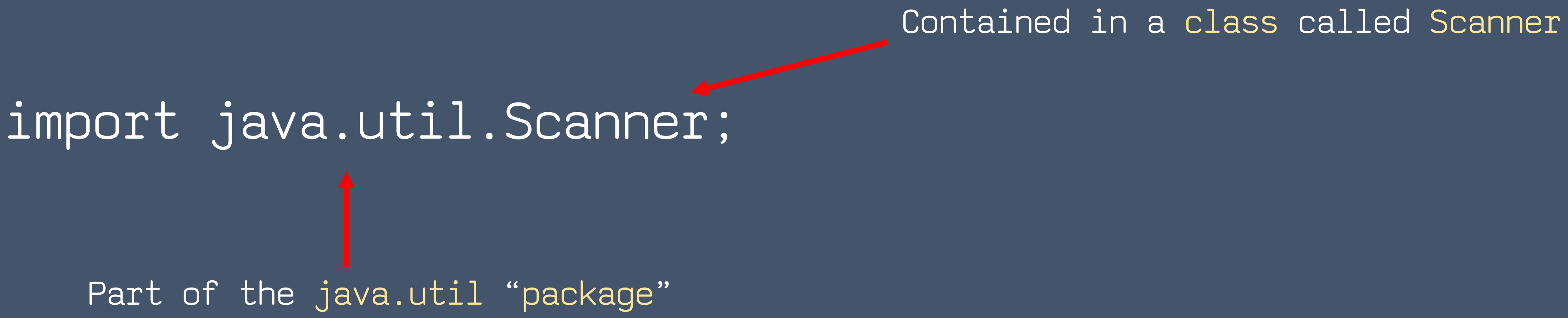
These values aren’t important anymore.

Data Type	Size	Min value	Max Value
String	Various	?	?
Scanner	Various	?	?
Random	Various	?	?

“reference” data type

Reference types: Scanner

Scanner exists outside of the Java API, so we have to import it:



Contained in a class called Scanner

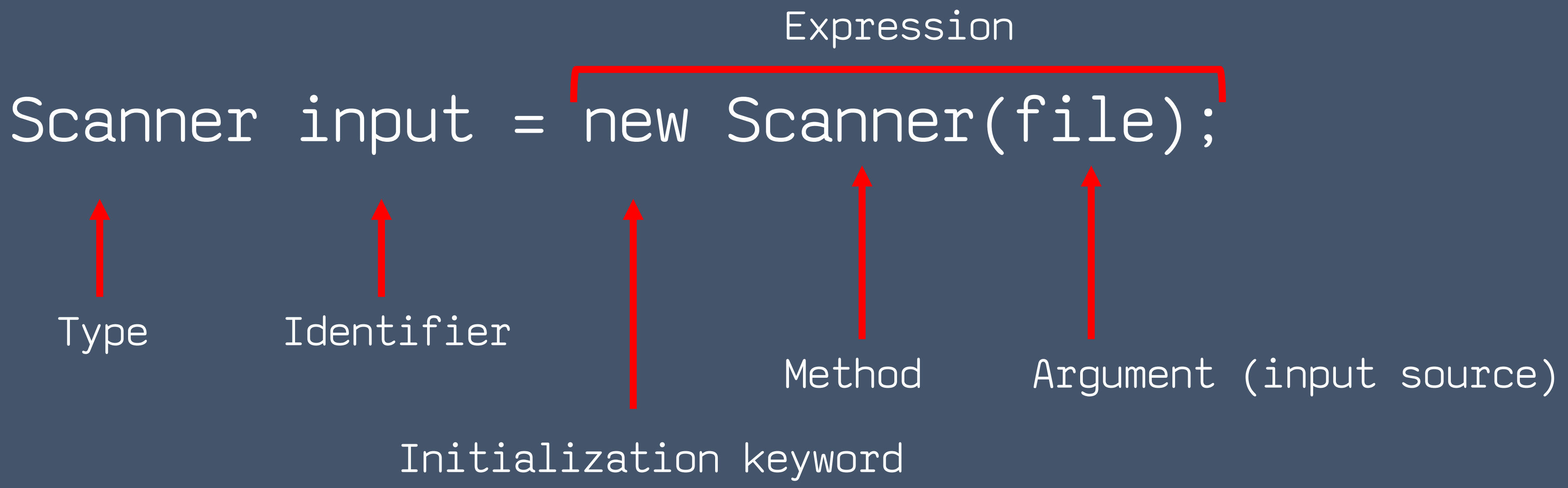
```
import java.util.Scanner;
```

Part of the java.util "package"

Reference types: Scanner

Once we’ve imported it, we can “summon” its powers when we invoke it:

```
File file = “inputs/cupcakes.nomnomnom”;
```



Expression

```
Scanner input = new Scanner(file);
```

Type

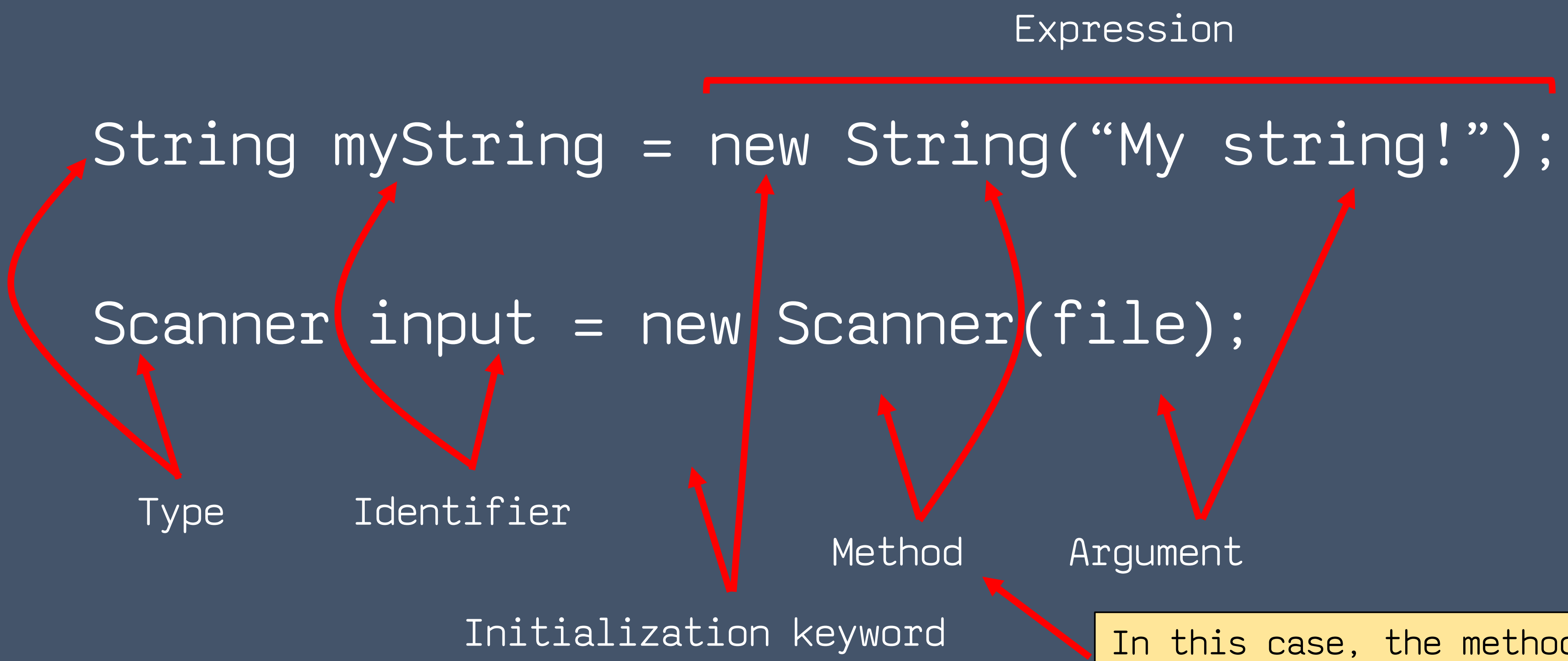
Identifier

Initialization keyword

Method

Argument (input source)

Reference types are objects



Expression

```
String myString = new String(“My string!”);

Scanner input = new Scanner(file);
```

Type

Identifier

Method

Argument

Initialization keyword

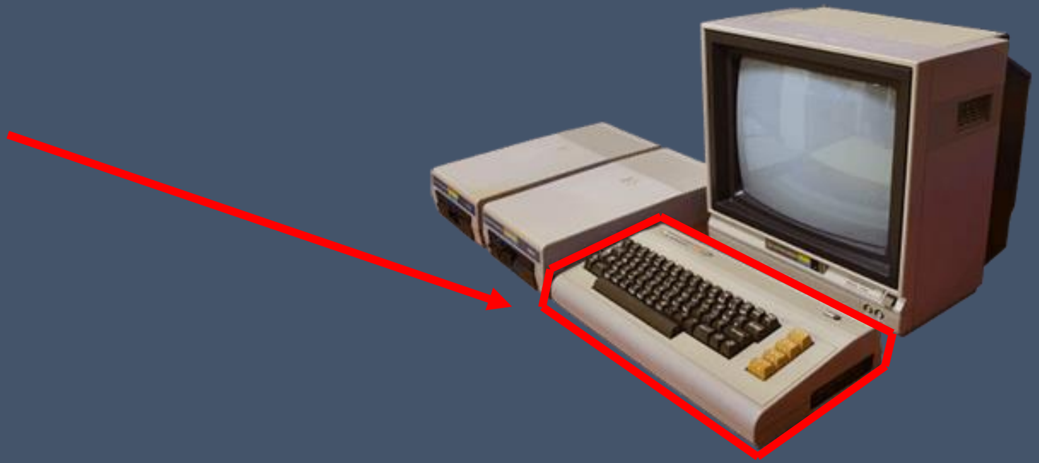
In this case, the method is called the “**constructor**” and every reference type has one - it specifies the *minimum* amount and type of data needed to create an **object**

Objects: Scanner

Creating a Scanner object requires 1 piece of data which represents an input:

- Files
- System input (STDIN)

  (System.in)
- Strings



```
int number    =  input.nextInt();               6          // 6

double number =  input.nextDouble();            6          // 6.0

String fName   =  input.next();                 G. Wiz     “G”

String fullName = input.nextLine();            G. Wiz\n    “G. Wiz”
```

cd to your activities folder

perform a git pull download master

cd to the activity-06 folder