Cheat Sheet: Primitive Data Types

Primitive Data Types in Java: There are 8 primitive data types in total in Java. We will only use some of those within this course - those covered in the videos.

Data Type:	**Description:**	**Size:**	**Default Value**
boolean	Represent truth values, can be true (correct) or false (incorrect)	1 bit	false
byte	Integers ranging from *-128* to *127*	8 bit	0
char	Unicode characters e.g. *'a'*, *'\u0041'*, *'ü'*, or *'\n'*	16 bit	\u0000
short	Integers ranging from *-32768* to *32767*	16 bit	0
int	Integers ranging from *-2147483648* to *2147483647*	32 bit	0
long	Integers ranging from *-9223372036854775808* to *9223372036854775807*	64 bit	0
float	Floating Point Numbers ranging from *-3.4028235E38* to *3.4028235E38*	32 bit	0.0
double	Floating Point Numbers ranging from *-1.7976931348623157E308* to *1.7976931348623157E308*	64 bit	0.0

Only Attributes of a class/an object have default values, local variables don't!

Cheat Sheet: Operators

Operators in Java:

Operator:	Description:
Arithmetic Operators	
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulo, Remainder of the division
-	Negative algebraic sign
- -	Decrement
++	Increment
Comparison Operators	
==	Equal
!=	Not equal
>	Greater than
<	Smaller than
>=	Greater or equal
<=	Smaller or equal
Logical Operators	
&&	AND
||	OR
!	NOT (Negation)

Cheat Sheet: Java Keywords

Keywords in Java: About 50 reserved words, so-called keywords, exist in Java. They can not be used as identifiers (e.g. for classes, methods, attributes, ...). Here is an overview of those used in the course.

Keyword	Description
new	Instantiates new objects
return	Returns a value from a method to the caller of the method
class	Defines a class
interface	Defines an interface (Week 3)
void	Return data type of a method that does not return anything

static	Defines an attribute or method as belonging to a whole class of objects or every object instantiated from the class. The value of the static attribute is the same for all objects. Static methods do not have access to non-static elements of a class. (Week 5)
private, protected, public	Access modifiers (Week 3)
abstract	A class or method that is not completely defined yet. Inheriting non-abstract classes have to define all declared (abstract) methods. Abstract classes cannot be instantiated themselves. (Week 3)
extends	Defines an inheritance relationship between a Subclass and a Superclass. (Week 3)
implements	Defines an implementation relationship between a class and an interface. (Week 3)
this.	Access attributes or methods of an object from within its methods. Only required when there is a naming conflict between a local variable of the method and the attribute of the object. this(): acces (Week 2, 3)
this()	Call an overloaded constructor of the same class from within another constructor. (Week 2, 3)
super.	Access methods of the Superclass. Only required when there is a naming conflict within the Subclass du to overridden methods. (Week 3)
super()	Call the constructor of the Superclass. Only possible from within the constructor of a Subclass. (Week 3)
if	First branch of a conditional statement
else	Further branches of conditional statements
for	Starts a for loop.(Week 2)

continue	Resume program execution at the end of the current loop body (skip the rest of the loop body for this iteration) (Week 2)
do	Starts a do-while loop. (Week 2)
while	Starts a while loop. (Week 2)
double	Data type for floating point numbers
int	Data type for integers
boolean	Data type, which represents truth values
true	Represents a correct boolean value.
false	Represents an incorrect boolean value.
null	Represents objects which do not exist yet. (Week 3)
package	Similar to folders or directories. Define a proper namespace in Java to avoid naming collisions. (Week 3)

Cheat Sheet: Definitions of Concepts

Concept:	**Description:**
Attribute	A variable that belongs to an object and is valid throughout the object (Can be used by all methods of the object.)
Argument	A value that is passed into a method
Parameter	Parameters allow to pass arguments to methods.
Class	Represents a real world entity, is a blueprint for many objects
Constructor	A special method that instantiates objects of a class
Method	Methods define the behavior of an object. Other programming languages know similar concepts as functions, subroutines, etc.

Cheat Sheet: Control Structures

Conditions:
```
if(<<boolean expression>>){
    //statements
}else if(<<boolean expression>>){
   //statements
}else{
   //statements
}
```
Loops: While-Loop: (Week 2)
```
while(<boolean  expression>){
   //Statements
}
```
Do-While-Loop: (Week 2)
```
do {
   //statements
} while(<boolean  expressions>)
```
Counting Loop: (Week 2)
```
for(int i = 0; i < 10; i++){
   //statements
}
```
For-each Loop: (Week 5)
```
for (String x : arrayIdentifier/collectionIdentifier) {
  //statements
}
```
Read as: for each String (replace with the data type of your choice) in the given array or collection, do whatever is defined in the statements.

Special Characters on different keyboard layouts (e.g. Brackets)

Brackets (and other special characters) are located on different locations on different keyboard layouts. Especially on german keyboards, to find some brackets is not trivial. On others (e.g. American layout) this is a little bit easier. Here is a short non-complete overview:

Windows

Square Brackets [ ] :
german/austrian Layout (QWERTZ): press AltGr + 8 and AltGr + 9 respectively.
Curly Braces { } :
german/austrian Layout (QWERTZ) : AltGr + 7 and AltGr + 0 respectively.
US Layout/English international (QWERTY): Shift + square bracket keys.
ampersand & :
German/Austrian Layout (QWERTZ) : press Shift + 6.
US Layout/English international (QWERTY): Shift + 7.
pipe | :
German/Austrian Layout (QWERTZ): press right Alt key + < (on the left side of the

left Shift key).
US Layout/English international (QWERTY): Shift + \ (backslash).

Mac

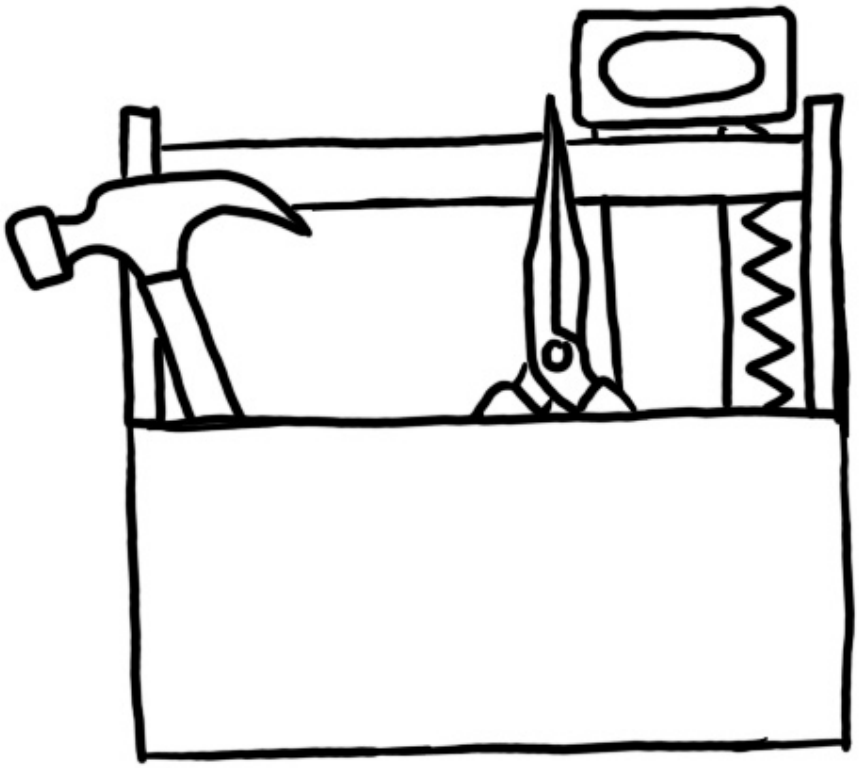
Square Brackets [ ] :
German/Austrian Layout (QWERTZ): press Alt + 5 and Alt + 6 respectively
Curly Braces { } :
German/Austrian Layout (QWERTZ): press Alt + 8 or Alt + 9 respectively.
US Layout/English international (QWERTY): Shift + square bracket keys [].
ampersand & :
German/Austrian Layout (QWERTZ): press Shift + 6.
US Layout/English international (QWERTY): press Shift + 7.
pipe | :
German/Austrian Layout (QWERTZ): press right Alt key + 7.
US Layout (QWERTY)/English international : Shift + \ (backslash).

Icons on the Slides

One icon that you will encounter quite often is the Terminal or Console. It shows results of a program that are printed on the command line (System.out.println(...))

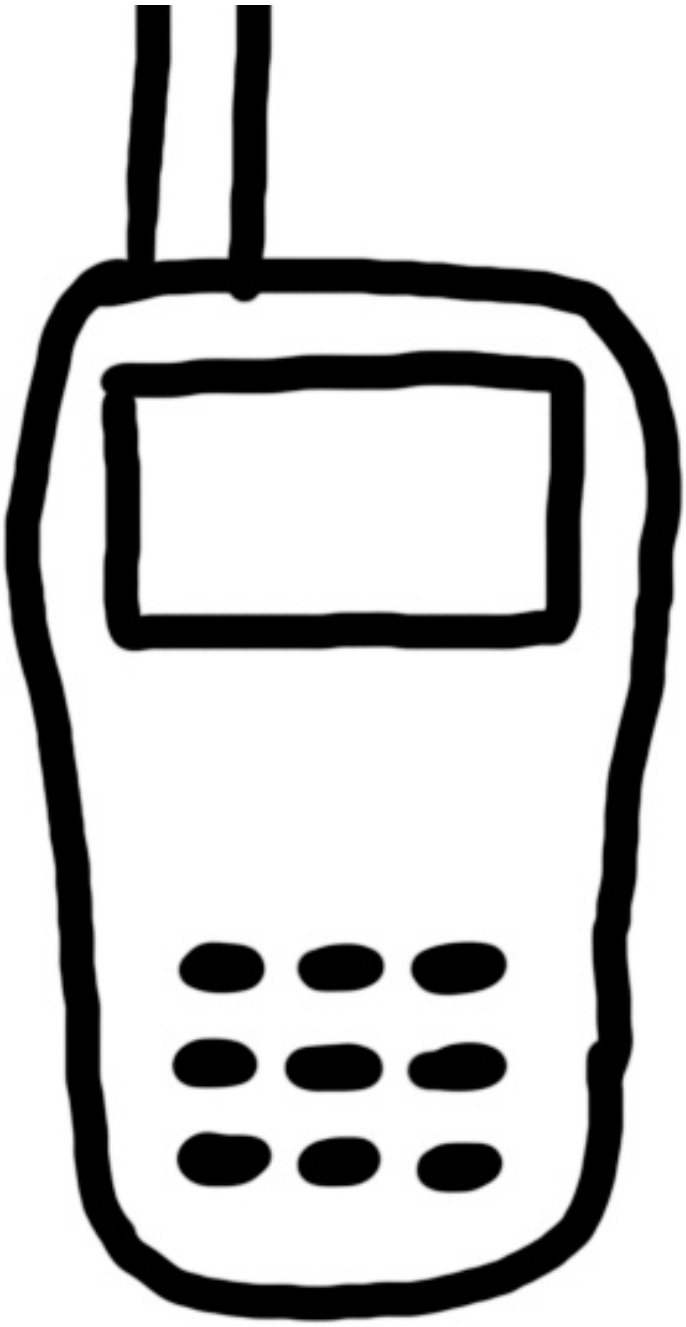
Another frequent icon is the siren. It represents situations and concepts that require special attention.

At the beginning of each new week, we summarize what we've learned so far and add the newly acquired skills to the toolbox.



The footsteps represent parts that we already had covered in previous videos. We come back to them as at this position we add some new details or need them to explain a new concept.

Whenever you see the walkie-talkie, it might be a good idea to go to the forum and discuss things.



Good to Know - Frequently Asked Questions

Which browsers are supported by our programming platform CodeOcean?

Please use a current version of Chrome (> version 60), Safari (> version 11), Firefox (> version 60), or Edge (> version 42).
We have tested all of these and made sure they work. Earlier version might work, but were not tested by us. In case you encounter other issues, please try to allow all JavaScripts that we require. NoScript or other blocking software might cause

issues. If you still struggle, please contact the helpdesk and let us know. We will then reach out to you.

I can't find the brackets on my keyboard (e.g. curly braces/square brackets/parentheses)...Help!

We have created an overview of some common keyboard layouts and where to find special characters here. If your keyboard layout is not among them: ask in the Discussion Forum or use the search engine of your choice.

I found an error in the course material.

Great, please let us know! :) Please write us via the helpdesk, or let us know in the forum.
We have created a thread explicitly for this purpose, please use it. If it is an ambiguity or you are not sure whether it is an error, we encourage you to post it in the forum: discussion fosters learning.

What do the icons (e.g. footprints, sirene) on the slides mean?

Good question! We have provided an overview of our icons and their meaning here.

How is the course structured? How is the course graded? How many points do I need to gain a Record of Achievement?

Please have a look here.

How do I format code in the forum?

Use the </> button above the text input field to properly indent and display your code snippet. It then looks like this:

```
class Plane  {

   void move() {
      fly();
   }

   void fly() {
      System.out.println("wrroooom!");
   }
}
```

First select your code snippet, then then click on the </> button. Don't be shy, try it.
You can use the same button to add smaller snippets, such as this one: `fly()` inline.
Unfortunately, the preview doesn't always show how it will look when it is submitted.
Also, you can have a look (and try it out) here.

Are Java and JavaScript the same?

No, they aren't actually. JavaScript is an interpreted language which is widely used in the web. Java and JavaScript are very different - except their name.