**SECTION 3 – KEY POINTS**

3.14 {

statement – complete command to be executed, can include 1 or more expressions

single quotes – for ‘characters’, double quotes – for “strings”

(Error: unclosed character literal)

}

3.15 {

key word – “reserved words” that have predefined meaning in Java

In Java syntax all code is case sensitive

int = keyword

Java keywords:

<https://docs.oracle.com/javase/specs/jls/se17/html/jls-3.html#jls-3.9>

}

3.16 {

variable – way to store information in our computer (are stored under a name in RAM – random access memory)

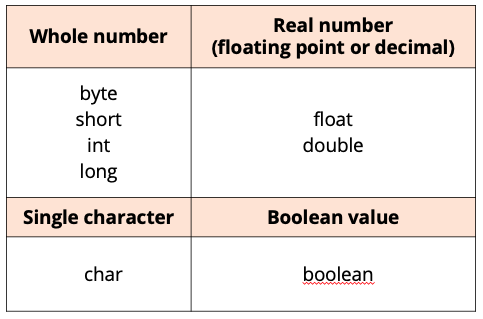
expression – the code segment that is on the right side of the equals sign

}

3.17 {

8 primitive data types in Java:

Primitive data types are simply placeholders in memory for a value

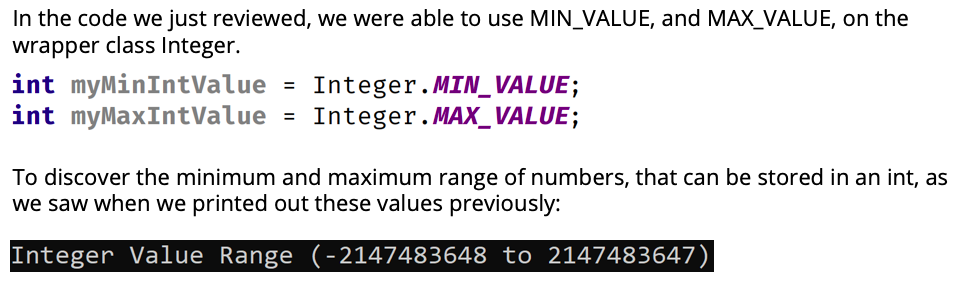


class – a building block for object – oriented programming, allows us to build custom data types

wrapper class – a special category of data type, which offers additional functionality that primitive types don’t

Java uses the concept of a wrapper class, for all of its eight primitive data types.

A wrapper class provides simple operations, as well as some basic information about the primitive data type, which cannot be stored on the primitive itself.



If you try and put a value larger than the maximum value into an int, you'll create something called an Overflow situation.

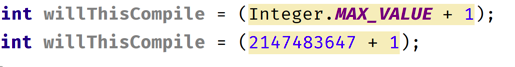
And similarly, if you try to put a value smaller than the minimum value into an int, you cause an Underflow to occur.

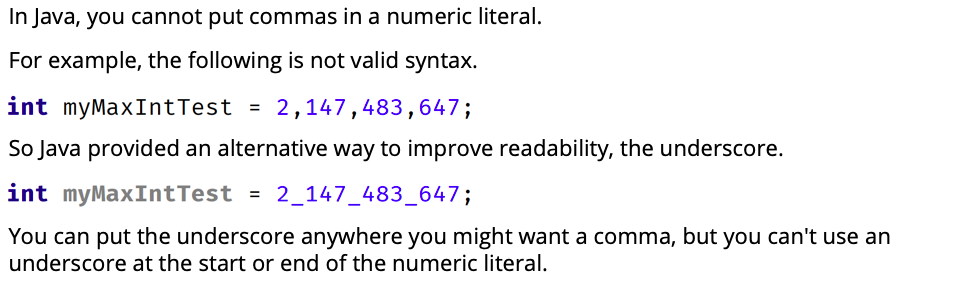
These situations are also known as integer wraparounds.

If you assign a numeric literal value to a data type that is outside of the range, the compiler DOES give you an error, for example:



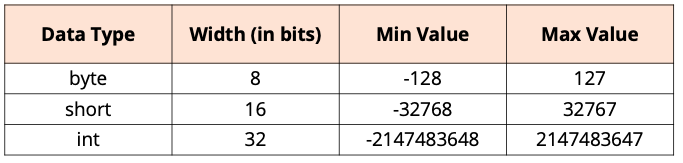
However, when we expressions the code will compile resulting in an overflow condition





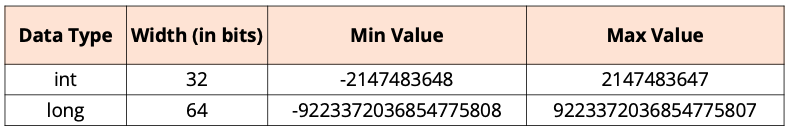
}

3.18 {



Java allows certain numeric literals to have a suffix appended to the value, to force it to be a different data type from the default type.

The long is one of these types and its suffix is an 'L'. A numeric literal that exceeds Integer.MAX\_VALUE must use the 'L' suffix.

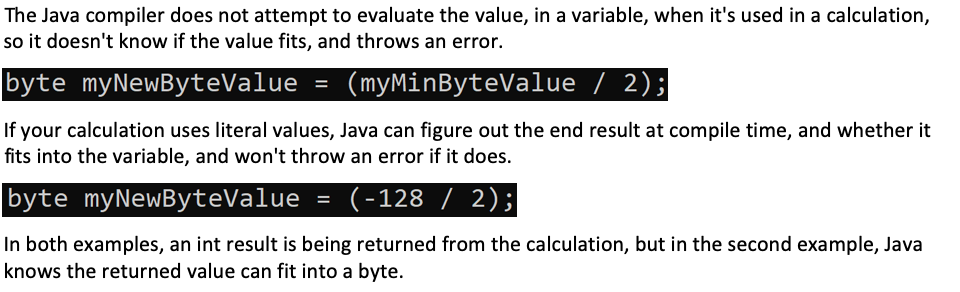


3.19 {

casting – a way to get Java to treat a variable of one type like a different type



Assigning expressions to variables with data types that don't match



this code works because we tell the compiler we know what we're doing by using this cast, and the compiler doesn't give an error:



}

3.20 {

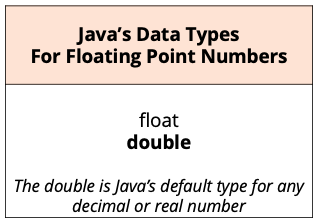
Parentheses are another way to make your code more readable.



}

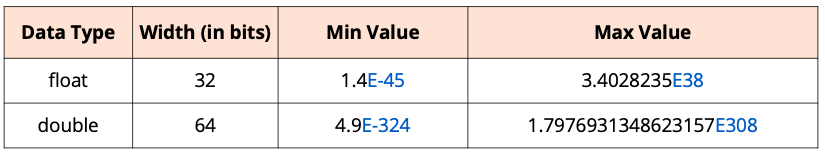
3.21 {

There are two primitive types in Java for expressing floating-point numbers, the float and the double.

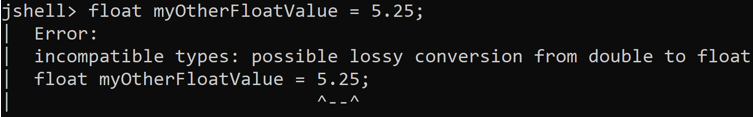


Scientific notation can be translated into more familiar terms, by replacing the 'E' in the number, with the phrase 'times 10 to the power of’.

1.4E-45 is the same as 1.4 x 10-45 and 3.4E38 is the same as 3.4 x 1038



The number 5.25 is a double, so assigning it to a float will raise an error:

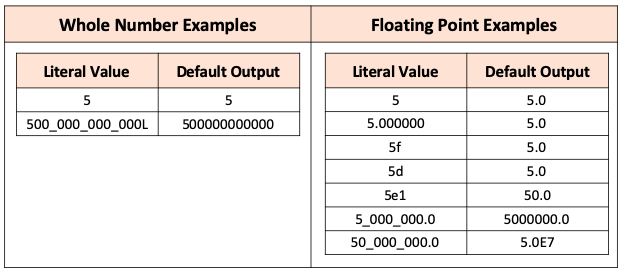


Not everyone realizes that Java's default data type for a decimal literal is a double, which is larger and more precise than a float.

Java likes to put a similar line of code in its code segments on exam questions, to what we saw earlier, omitting that 'f' suffix. Without a computer to check, this statement can look fairly innocuous.

}

3.22 {



interesting difference is that, for whole numbers, the output is never in scientific notation, but for real numbers, it could be

Why should we choose double?

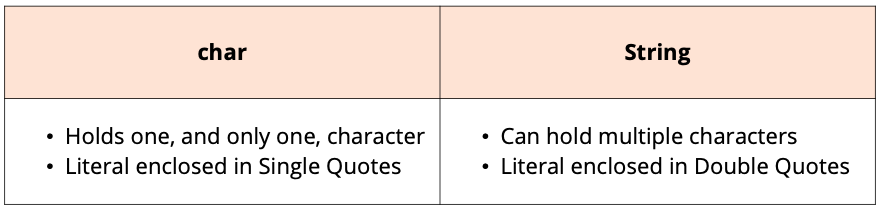
* it's more precise, and it can handle a larger range of numbers.
* faster to process on many modern computers
* the Java libraries, particularly math functions, are often written to process doubles and not floats, and to return the

result as a double

}

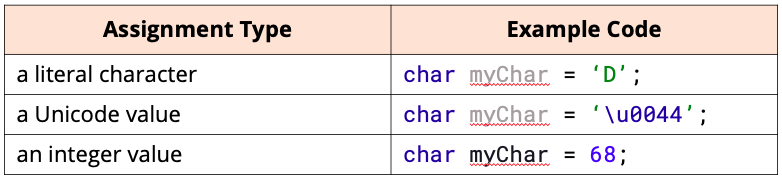
3.23 {

Differences between a char and a String:



unicode - an international encoding standard for use with different languages and scripts by which each letter, digit, or symbol is assigned a unique numeric value that applies across different platforms and programs.

Assigning values to a char variable:



A boolean value allows for two opposite choices, true or false, yes or no, one or zero.

In Java terms, we've got a boolean primitive type, and it can be set to two values only, either true or false.

Developers will often use the word, is, as a prefix for a boolean variable name.

Examples:

* isMarries
* hasChildren
* hasValidLicence
* isCustomerOverTwentyOne

}

3.24 {

A String is a class that contains a sequence of characters

String concatenation - A String + anything else, gives us a String as a result, concatenating anything after the String as text to the initial String.

Strings are Immutable! (you can't change a String after it's created)

String vs StringBuilder:

* The String class is immutable, but can be used much like a primitive data type.
* The StringBuilder class is mutable, but does not share the String's special features, such as being able to assign it a String literal or use the + operator on it.

Both are classes, but the String class is in a special category in the Java language.

}

4.32

Use camel case! (upper – project name, class name; lower – method name, variable name)

Psvm – shortcut

4.33

The public java keyword is what’s called access modifier (allows us to define which part of our code can access a particular element)

{} class body

Method – collection of statements that perform operations

() method declaration

4.34

Putting a semi colon after an if statement – the line after is not dependent on the previous line, so it will execute anyway