Java-notes

* Data types:
* Here, the range for various inbuilt data type is fixed.
* Like

Int – 4 byte – 32bits

Float – 4 bytes – 32bits

* We, can add underscore between number to make human readable
* Example

Int a = 10\_00\_000;

It will convert a = 1000000

* Floating point computations follows the IEEE 754 specification. In particular, there are three special floating-point values to denote overflows and errors:
* Positive infinity -> for (dividing positive number by 0)
* Negative infinity -> for (dividing negative number by 0)
* NaN (not a number) -> for (0/0 and sqrt(-Ve) …)
* Char is of 2-bytes
* As encoding here used is Unicode instead of Ascii like c++.
* It’s recommended to not use *char* data type until we are manipulating UTF-16 code units. Why?
* UTF-16 is a variable length encoding scheme. For characters that can fit into the 16 bits space, it uses 2 bytes to represent them. For all other characters, it uses 4 bytes.
* To be fair to char, it will work fine most of the time for many applications. It isn’t broken but it has a flaw which could ‘break’ your application silently and make your users see garbled text.
* **emojis characters are supported by all popular applications these days**. And it’s represented using two characters. It may create garbled some times. Like emoji is not displaying in proper manner.
* In, short try to use String data type instead of char as much as possible In java as they are internally represented using UTF-16 encoding.
* Unlike c++, conversion between integer and Boolean is not allowed.
* We can use *var* data-type if we initializing variable at declaration time as, it will automatically select data-type according to value.(JAVA10 onwards)
* Note: Integer divides by 0 rise a exception whereas floating-point divides by 0 yield an infinite or NaN result
* Operator
* & and | are bit wise operator but, when they applied to Boolean type it will work same as && and || but difference is they will not “Short circuited”.