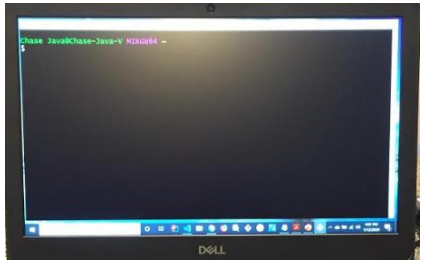
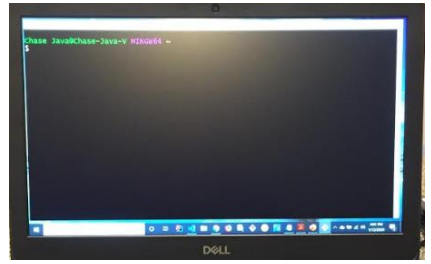


Cohort Laptop



Instructor Laptop

Bitbucket

JPMC Tech Start

Repositories

Filter by: All Public

Repository

Java java-main

Java victoria-hinkle-moore

Java victoria-hinkle-moore

fork

Cohort Repo: first-lastname

Java java-main

push origin master

pull upstream master

push origin master

- Submit your exercise work

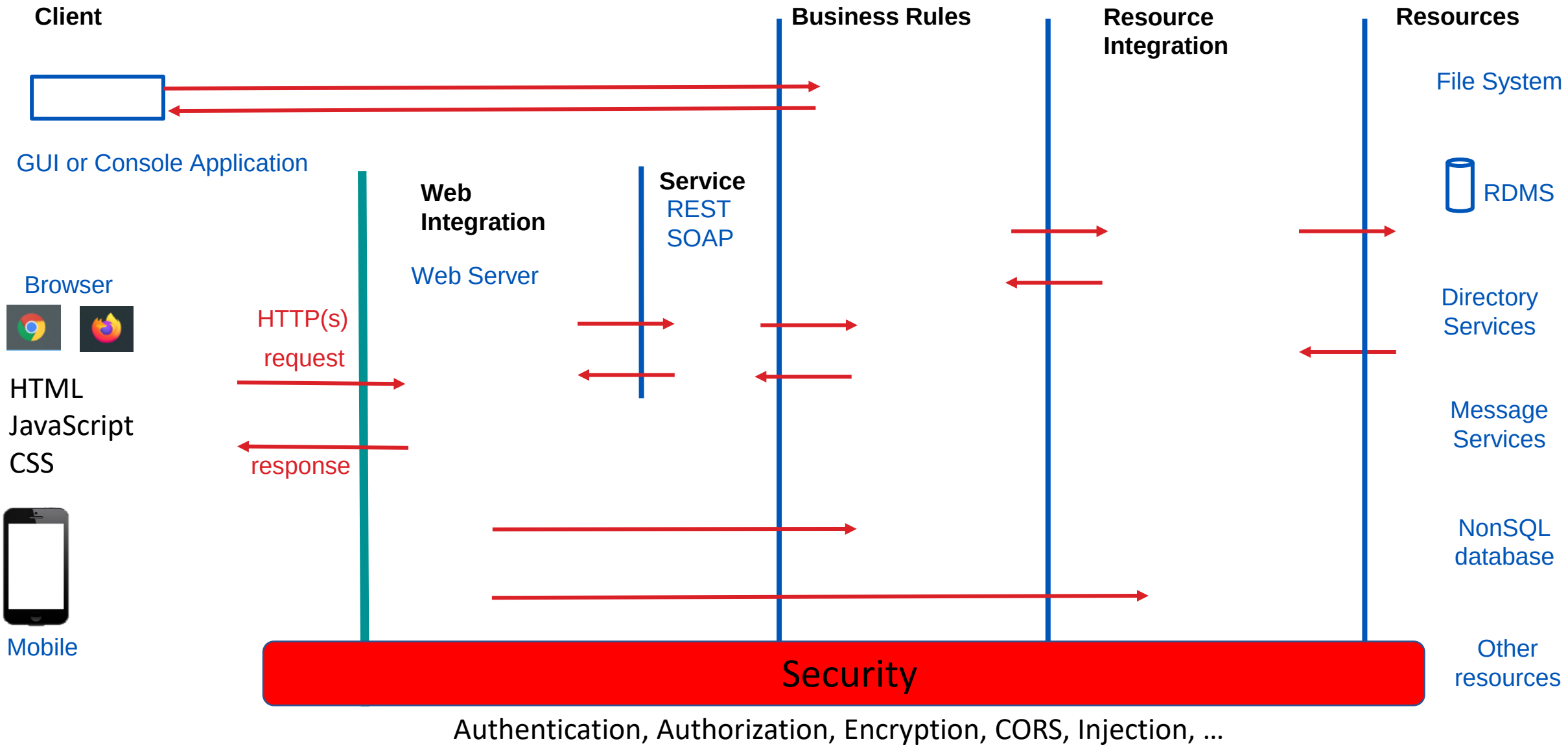
Daily Git Commands

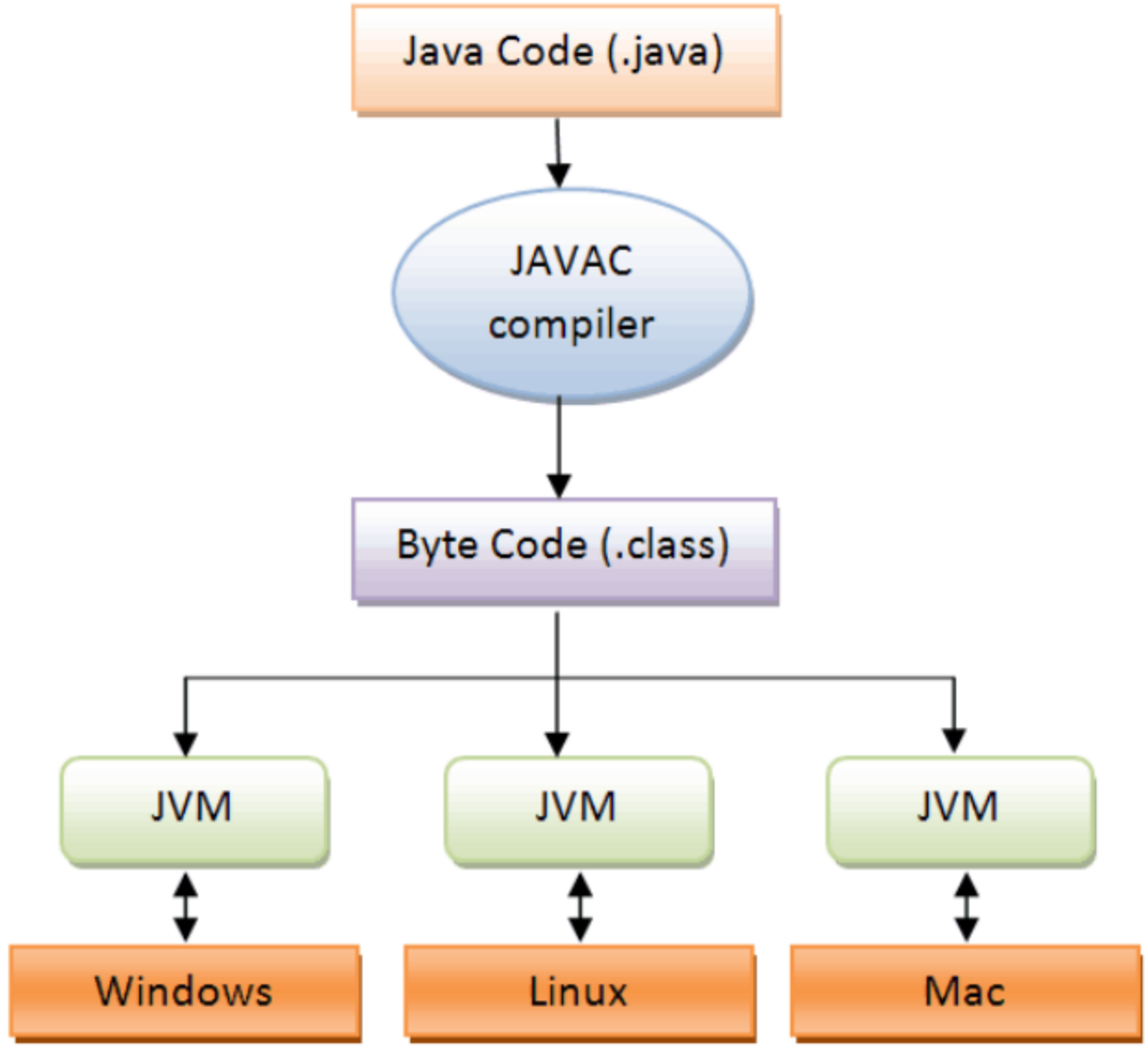
- Execute the git commands from the directory top of your repository
  - Current directory should be your-name folder inside your workspace folder

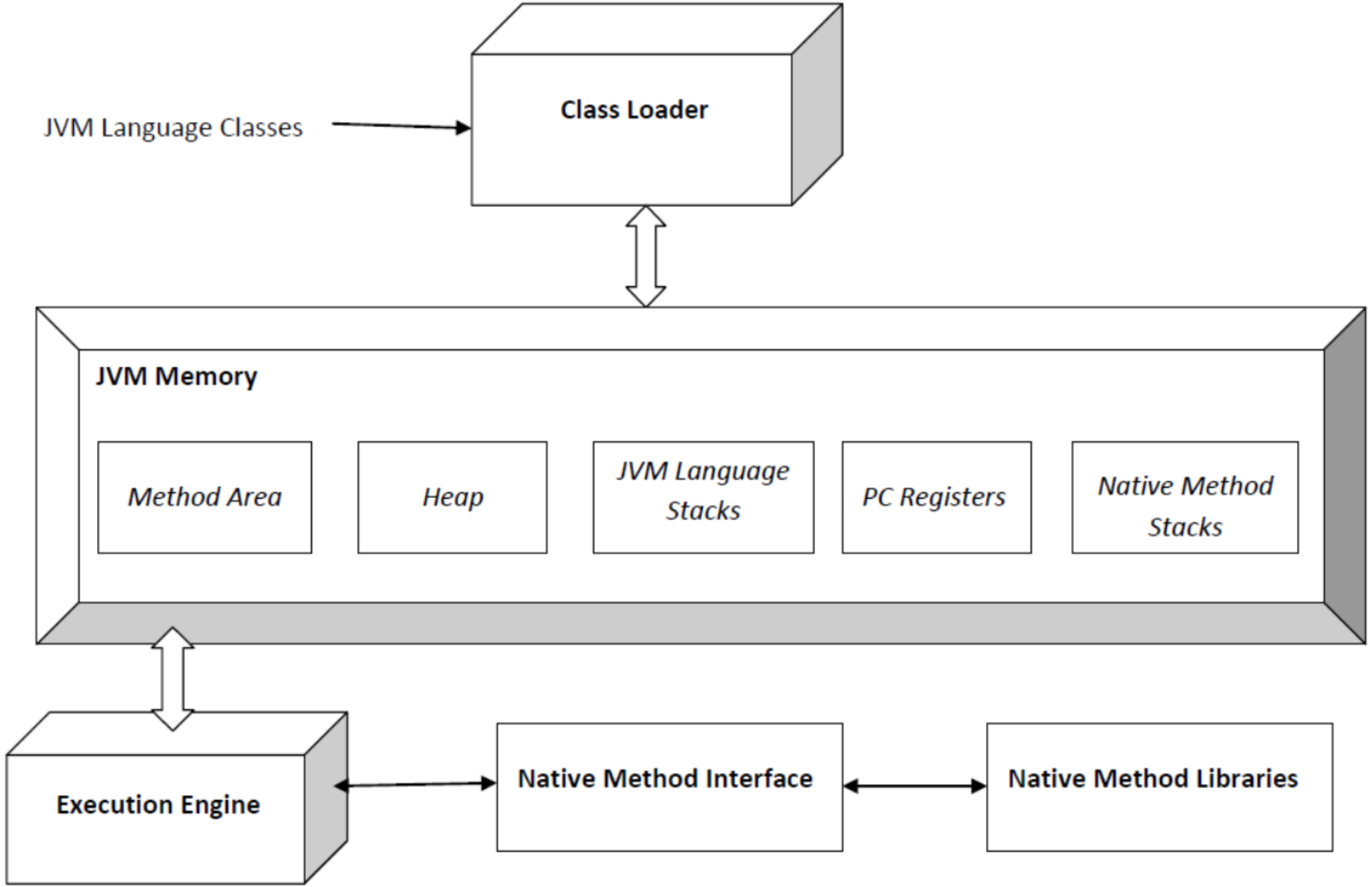
- Git Commands to **Pull** Daily Work From BitBucket
  - git pull upstream master

- Git Commands to **Push** Your Work To BitBucket
  - git add -A
  - git commit -m "with message"
  - git push origin master

Application Architecture




Java Code (.java)
JAVAC compiler
Byte Code (.class)
JVM
JVM
JVM
Windows
Linux
Mac


JVM Language Classes
Class Loader
JVM Memory
Method Area
Heap
JVM Language Stacks
PC Registers
Native Method Stacks
Execution Engine
Native Method Interface
Native Method Libraries

TYPE	DESCRIPTION	DEFAULT	SIZE	EXAMPLE LITERALS	RANGE OF VALUES
boolean	true or false	false	1 bit	true, false	true, false
byte	twos complement integer	0	8 bits	(none)	-128 to 127
char	unicode character	\u0000	16 bits	'a', '\u0041', '\101', '\\', '\','\n',' β'	character representation of ASCII values 0 to 255
short	twos complement integer	0	16 bits	(none)	-32,768 to 32,767
int	twos complement integer	0	32 bits	-2, -1, 0, 1, 2	-2,147,483,648 to 2,147,483,647
long	twos complement integer	0	64 bits	-2L, -1L, 0L, 1L, 2L	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	IEEE 754 floating point	0.0	32 bits	1.23e100f, -1.23e-100f, .3f, 3.14F	upto 7 decimal digits
double	IEEE 754 floating point	0.0	64 bits	1.23456e300d, -1.23456e-300d, 1e1d	upto 16 decimal digits

Operator Precedence

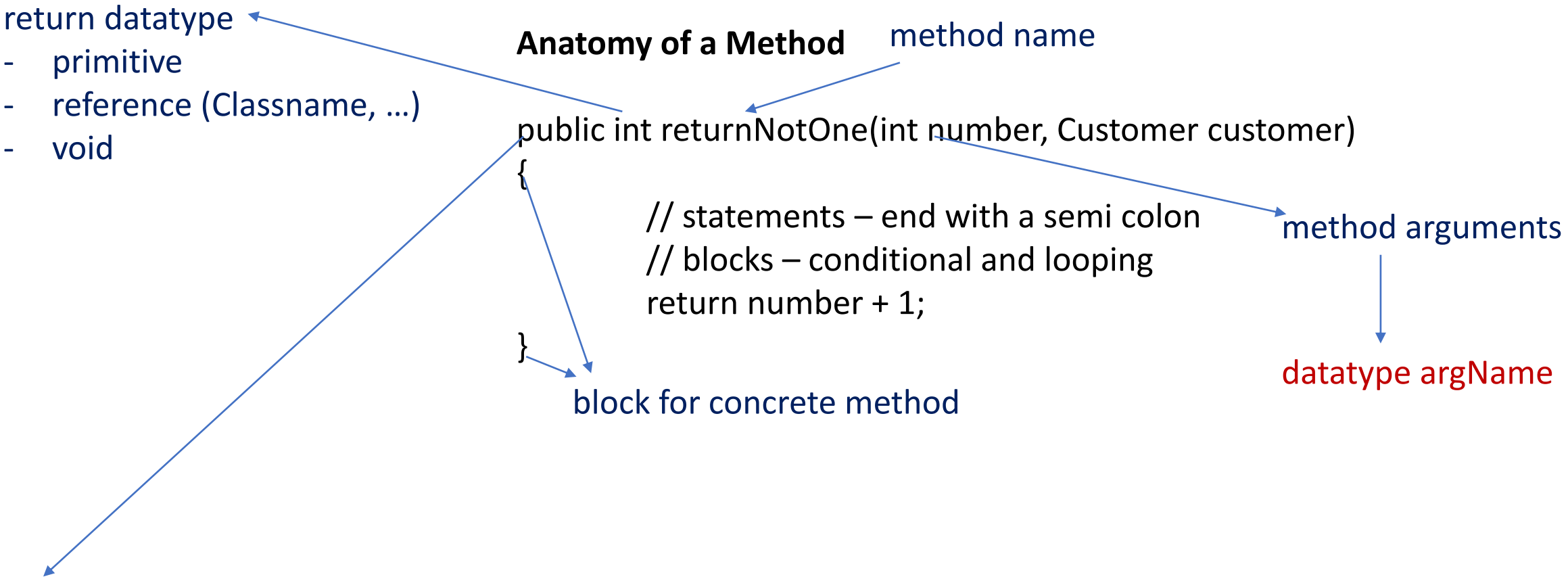
Operators	Precedence
postfix	*expr*++ *expr*--
unary	++*expr* --*expr* +*expr* -*expr* ~ !
multiplicative	* / %
additive	+ -
shift	<< >> >>>
relational	< > <= >= instanceof
equality	== !=
bitwise AND	&
bitwise exclusive OR	^
bitwise inclusive OR	|
logical AND	&&
logical OR	||
ternary	? :
assignment	= += -= *= /= %= &= ^= |= <<= >>= >>>=

Precedence	Operator	Type	Associativity
15	() [] .	Parentheses Array subscript Member selection	Left to Right
14	++ --	Unary post-increment Unary post-decrement	Right to left
13	++ -- + - ! ~ (*type*)	Unary pre-increment Unary pre-decrement Unary plus Unary minus Unary logical negation Unary bitwise complement Unary type cast	Right to left
12	* / %	Multiplication Division Modulus	Left to right
11	+ -	Addition Subtraction	Left to right
10	<< >> >>>	Bitwise left shift Bitwise right shift with sign extension Bitwise right shift with zero extension	Left to right
9	< <= > >= instanceof	Relational less than Relational less than or equal Relational greater than Relational greater than or equal Type comparison (objects only)	Left to right
8	== !=	Relational is equal to Relational is not equal to	Left to right
7	&	Bitwise AND	Left to right
6	^	Bitwise exclusive OR	Left to right
5	|	Bitwise inclusive OR	Left to right
4	&&	Logical AND	Left to right
3	||	Logical OR	Left to right
2	? :	Ternary conditional	Right to left
1	= += -= *= /= %=	Assignment Addition assignment Subtraction assignment Multiplication assignment Division assignment Modulus assignment	Right to left

Logical operator table

The following table sums up the different logical operators:

`A`	`B`	`!A`	`A && B`	`A || B`	`A ^ B`
TRUE	**TRUE**	FALSE	TRUE	TRUE	FALSE
TRUE	**FALSE**	FALSE	FALSE	TRUE	TRUE
FALSE	**TRUE**	TRUE	FALSE	TRUE	TRUE
FALSE	**FALSE**	TRUE	FALSE	FALSE	FALSE



scope

- private – class itself
- default (no modifier) – plus other classes in same package
- protected – plus subclasses in another package
- public – plus all other classes

Stack

Heap

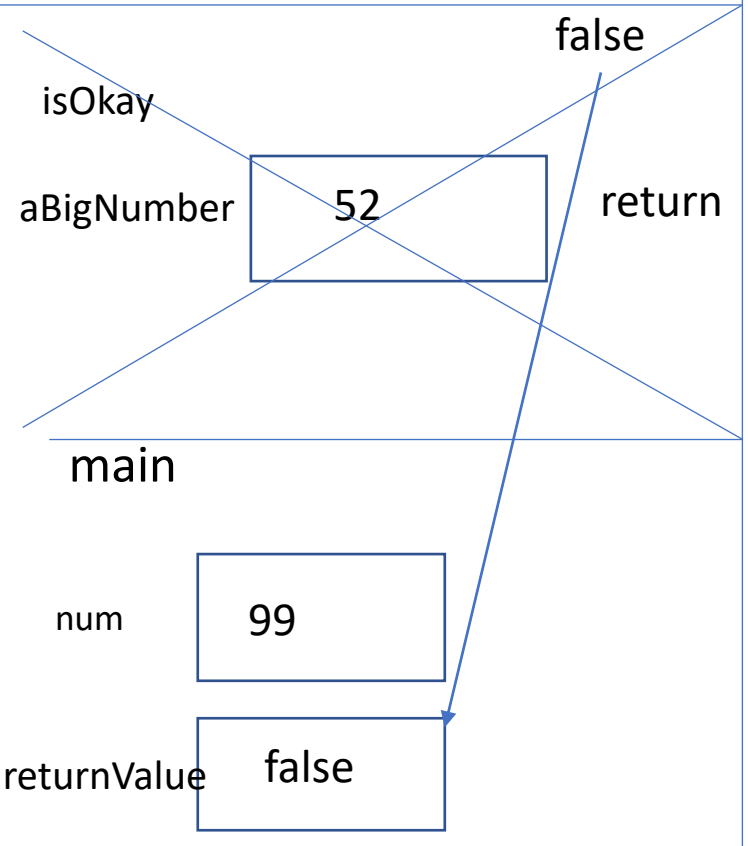
Static Heap

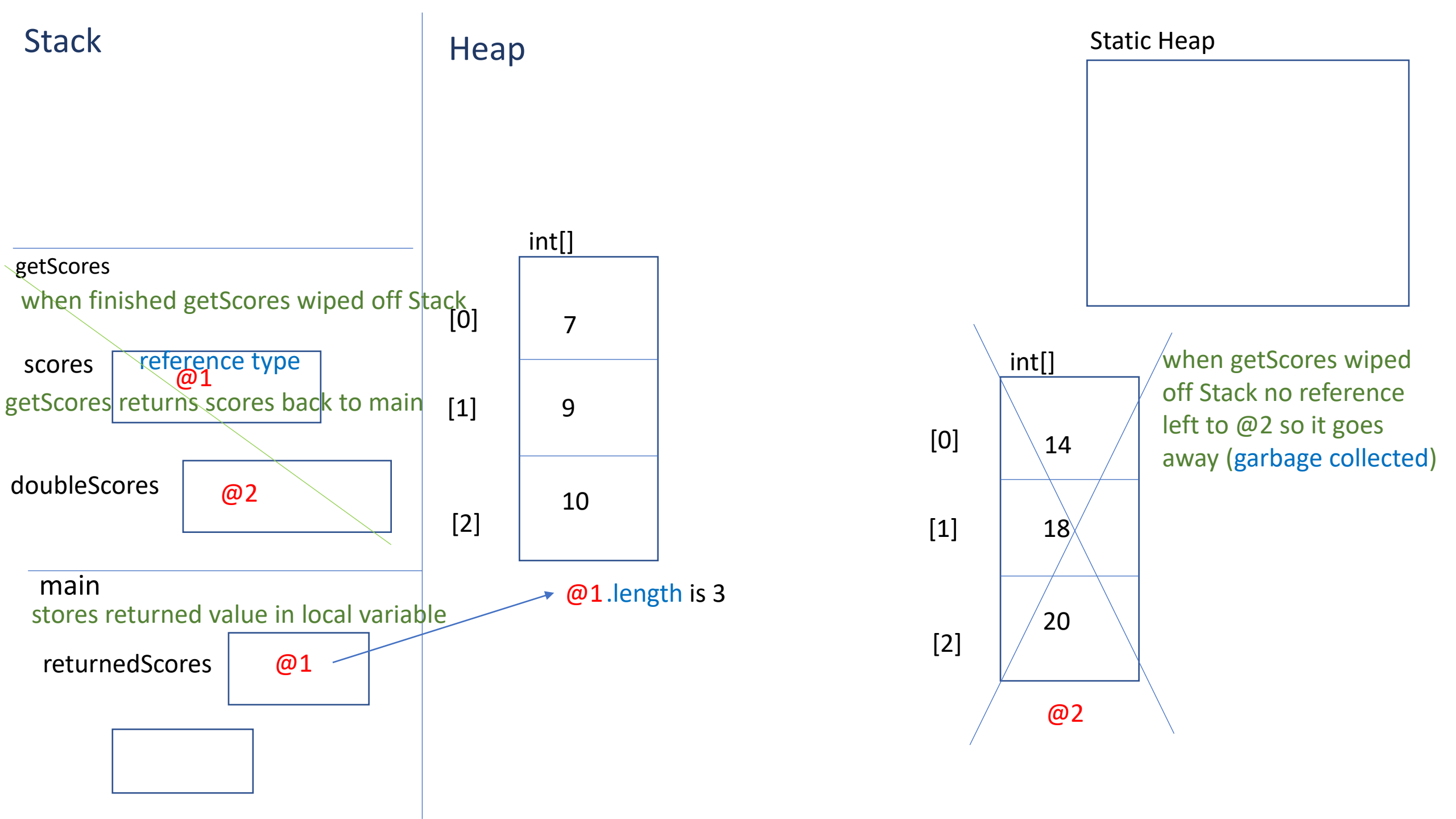
main

Stack

Heap

Static Heap




Stack
Heap
Static Heap
getScores
when finished getScores wiped off Stack
scores
reference type
@1
getScores returns scores back to main
doubleScores
@2
main
stores returned value in local variable
returnedScores
@1
int[]
[0]
7
[1]
9
[2]
10
@1.length is 3
int[]
[0]
14
[1]
18
[2]
20
@2
when getScores wiped off Stack no reference left to @2 so it goes away (garbage collected)

Stack

main

first @1

second @1

Heap

boolean[]

[0]	false
[1]	true
[2]	true
[3]	false

@1

Static Heap