SDET UNIVERSITY

Basics of Programming

Primitive

Numerical

character

integer

floating-point

Boolean

boolean

Non-Primitive

String

Array

...

Ranges of Primitive Data Types

Type	Size in Bytes	Min Range	Max Range	
byte	1	-2^{7}	$2^{7}-1$	
short	2	-2^{15}	$2^{15}-1$	*Integers*
int	4	-2^{31}	$2^{31}-1$	
long	8	-2^{63}	$2^{63}-1$	
char	2	0	$2^{16}-1$	
float	4	-2^{31}	$2^{31}-1$	*Decimals*
double	8	-2^{63}	$2^{63}-1$	
boolean	1*	0 “FALSE”	1 “TRUE”	

**JVM Specific; typically 1 byte*

Keywords / Reserved Words

```
abstract
assert
boolean
break
byte
case
catch
char
class
const
continue
default
do
double
else
enum
extends
final
finally
float
for
goto
if
implements
import
instanceof
int
interface
long
native
new
package
private
protected
public
return
short
static
strictfp
super
switch
synchronized
this
throw
throws
transient
try
void
volatile
while
```

Local Variables

Declared and available only within a block

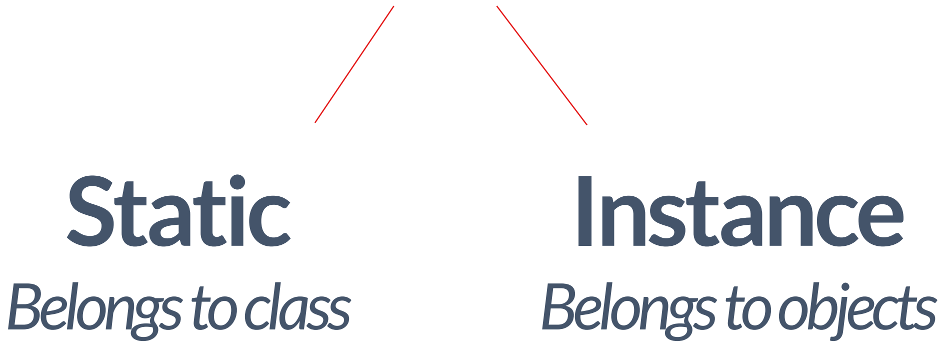
cannot use access specifier

ex: method, loop, if-statement, etc

Class Variables

Declared at class-level

Takes access specifier



Static

Belongs to class

Instance

Belongs to objects

Relational Operators

Relational operators compare to like types and evaluate to `boolean` values

< … *"is less than"*

> … *"is greater than"*

<= … *"is less than or equal to"*

=> … *"is greater than or equal to"*

== … *"is equal to"*

!= … *"is not equal to"*

Conditional Logical Operators

&& … *“and”*

|| … *“or”*

! … *“not, opposite of"*

exp1	exp2	(exp1 && exp1)	(exp1 || exp1)	(!exp1)
true	*true*	true	true	false
true	*false*	false	true	false
false	*true*	false	true	true
false	*false*	false	false	true

Logical operators are binary operators that require boolean values

`if` Statement

If-Statements test the condition and returns `boolean` value.

Syntax:

```
if (condition) {
  statements
}
else if (condition) {
  …
}
else {  …  }
```

`switch` Statement

Switch statements test equivalence value of the same variable

Syntax:

```
var variable;
switch (variable) {
 case expression: statements;
 case expression: statements;
 default: statements;
}
```

`do / while` Loops

Do and While Loops execute block when condition evaluates to True

Syntax:

```
while (condition) {
   statements
}

do {
   statements
} while (condition)
```

What’s the difference between while and do Loops?

for Loop

For-Loops are enhanced structure to iterate through set of statements

Syntax:

```
for (initialization; condition; iteration) {
   statements
}
```

Initialization: used to initialize an index variable and is executed only once (at beginning of loop)

Condition: tests the condition before executing code

Iteration: actions changed for each iteration (generally increment or decrement to index variable)

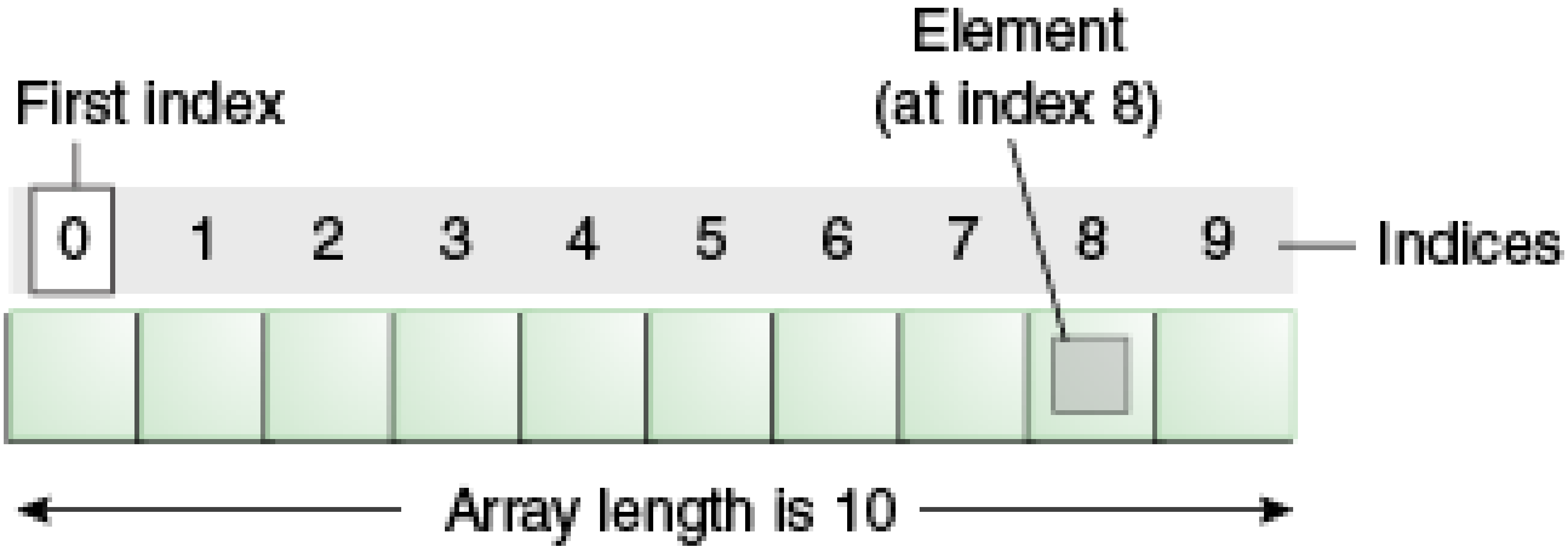
ARRAYS

What is an Array?

Definition: ordering by index of similar types of elements

Arrays hold multiple values of same data type into one variable, with each value identified by an index

ARRAYS


Element
(at index 8)
First index
0
1
2
3
4
5
6
7
8
9
Indices
Array length is 10

How do you Define an Array?

```
int[] arrayOfIntegers;
String []arrayOfStrings;
Double arrayOfDoubles[];
```

How do you Instantiate an Array?

```
arrayName = new dataType[size];
int myArray[] = new int[5];
```

How do you Initialize an Array?

```
arrayName[index] = value;
myArray[0] = “Java”;
myArray[1] = “Oracle”;
…
```

Declare + Instantiate + Initialize an Array?

```
int myArray[] = new Numbers{1,2,3};
```