CM1601 - Programming Fundamentals

Tokens in the Java Language

Keywords - Identifiers - Literals - Operators- Separators

Sem 2 - Week 2 |Cassim Farook

Learning Outcomes

- On completion of this lecture, students are expected to be able to:
- Recognise tokens used in writing a java program
- Identify primitive data types
- Define variables
- Identify how to output values
- Identify how to input values

Tokens used in the Java Language

- Tokens are the smallest elements of a program that are meaningful to the compiler. A Java program is a collection of tokens, comments and white spaces.
- There are five types of tokens.
  1. Keywords - Reserved words, not to be used for naming.
  2. Identifiers – Names used/written by developer
  3. Literals - Constant values of a data type
  4. Operators - specific operations such as mathematical symbols
  5. Separators - separate different parts of the code

1. Keywords (Reserved Words)

- Keywords are known as "reserved words" have pre-defined meanings in Java.
- These reserved words cannot be used for developer-defined identifiers.
- Reserved words use only lowercase letters.

primitive/built-in data types	int, double, char, boolean
scope/access control modifiers	public, private, protected
control statements	if, else, switch, while, for
built-in constants	true, false
Object-Oriented Keywords	class, extends, implements, new, this, super, final, static
exception handling keywords	try, catch, finally, throw, assert, throws

GeeksforGeeks (2018). *Java Keywords*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/java/java-keywords/ [Accessed: 2026/01/26]

1. Keywords (Reserved Words)

- There are about 60 reserved words total.

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

2. Identifiers (Defining names)

- Programmer-defined names that are given to various program elements.
    - Variables names
    - Methods names
    - Names given for arrays, classes, objects, packages, interfaces.
- Identifiers consist of alphabets, digits, underscore and dollar sign.
- Must not begin with a digit.
- Case sensitive.
- No white spaces.
- Can be of any length.

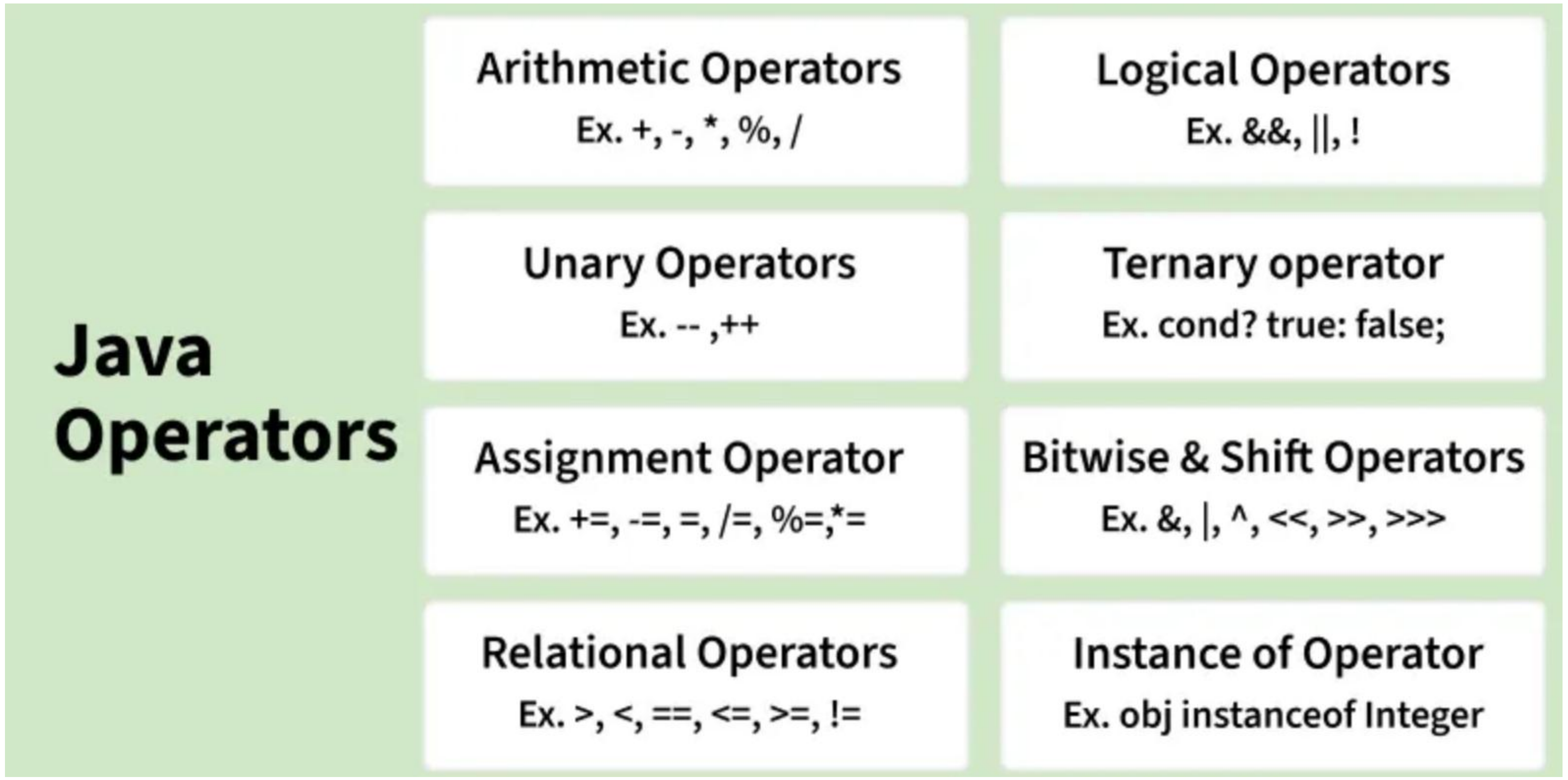
3. Literals

There are five basic types of literals in Java. Literals are constant values of a data type that a developer will write withing the code lines.

1. Integral literals
2. Floating point literals
3. Character literals
4. String literals
5. Boolean literals

4. Operators

- Special symbols that perform operations on variables or values.



Java Operators

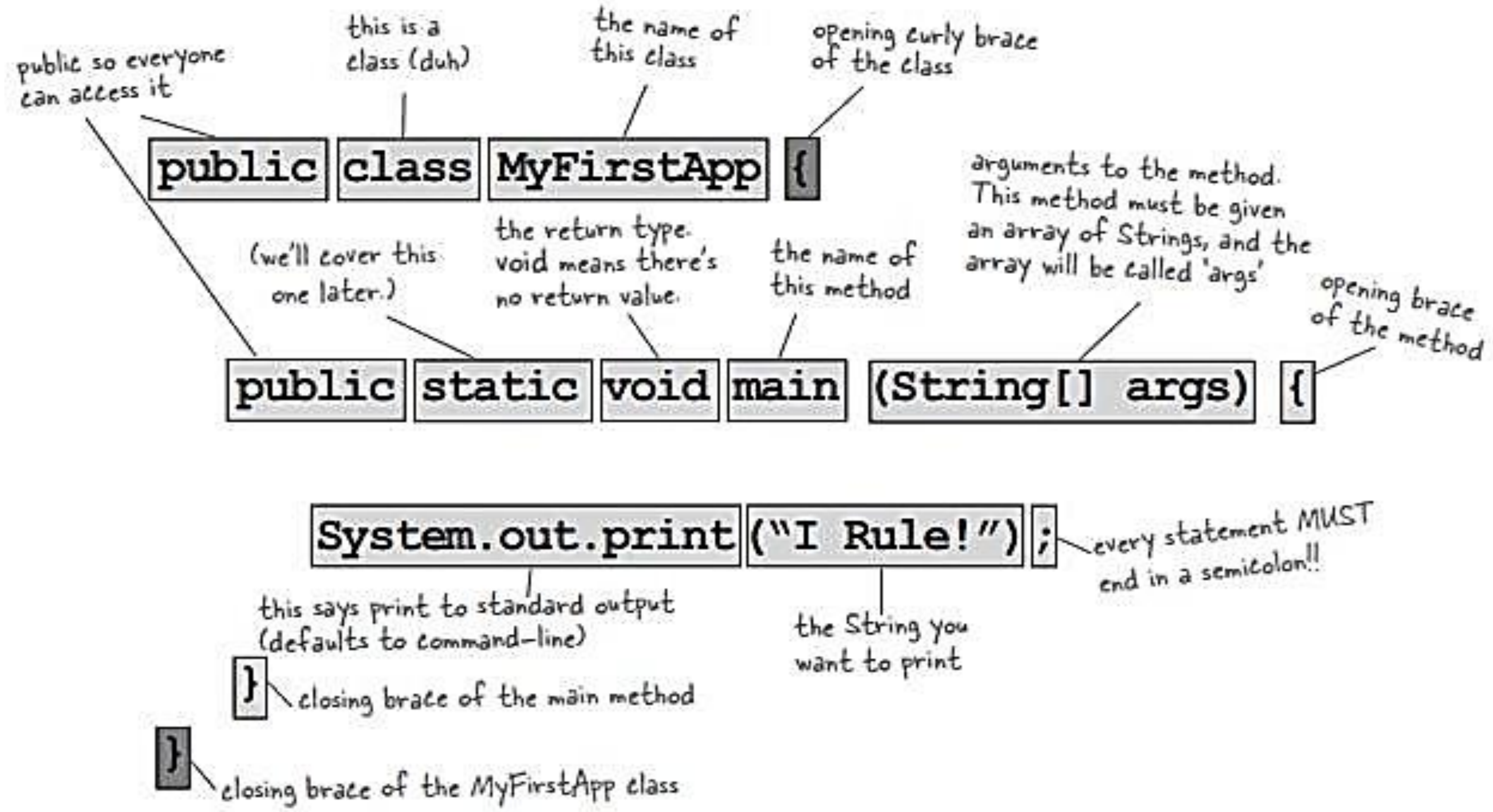
Arithmetic Operators Ex. +, -, *, %, /	**Logical Operators** Ex. &&, ||, !
Unary Operators Ex. -- ,++	**Ternary operator** Ex. cond? true: false;
Assignment Operator Ex. +=, -=, =, /=, %=,*=	**Bitwise & Shift Operators** Ex. &, |, ^, <<, >>, >>>
Relational Operators Ex. >, <, ==, <=, >=, !=	**Instance of Operator** Ex. obj instanceof Integer

GeeksforGeeks (2017). *Java Operators*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/java/operators-in-java/ [Date Accessed: 2026-01-26].

5. Separators

Round Brackets	( )	for method arguments
Curly Braces	{ }	for code blocks
Square Braces	[ ]	for defining lists
Comma	,	for separating elements
Period	.	for accessing instance methods/properties
Semi-colon	;	for terminating statements

A simple Java application

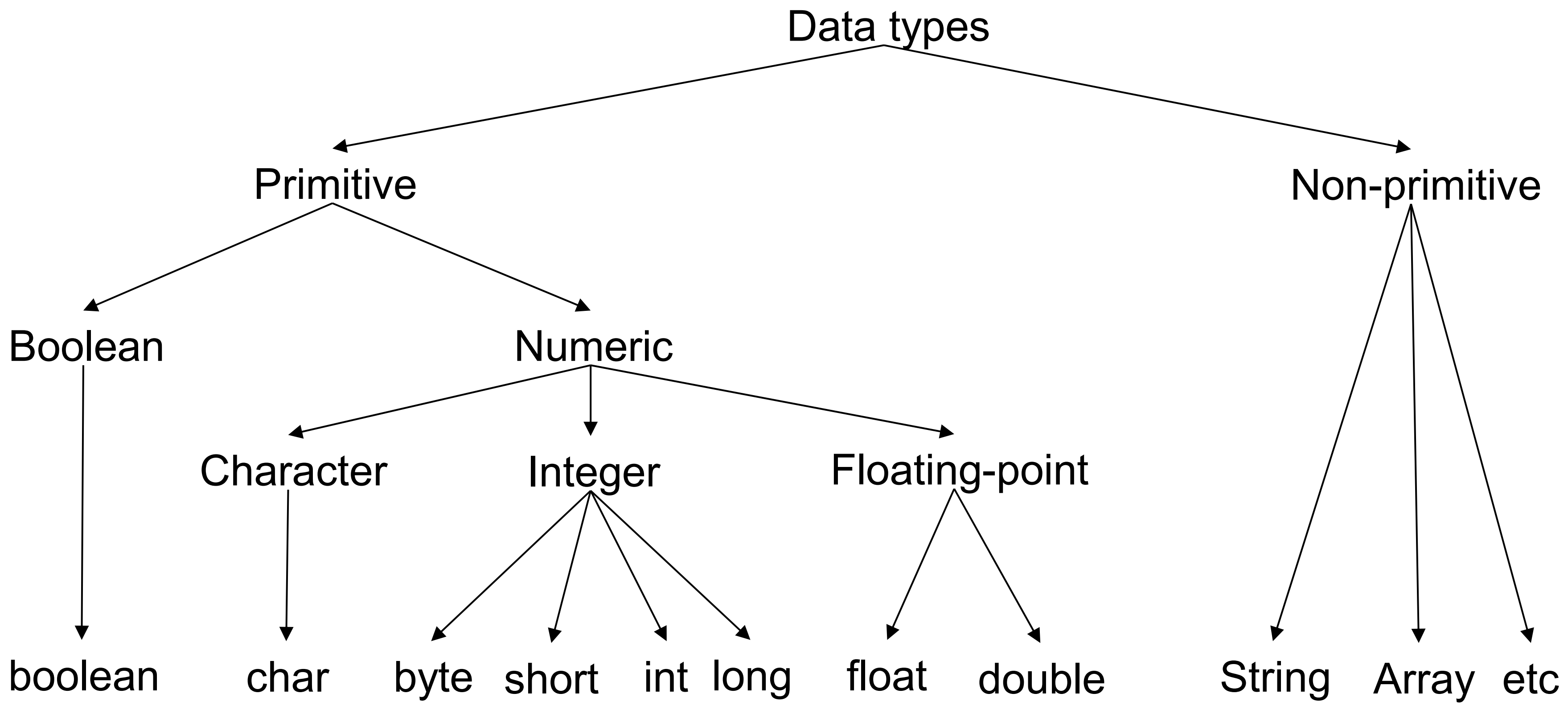


```
public class MyFirstApp {
    public static void main (String[] args) {
        System.out.print("I Rule!");
    }
}
```

public so everyone can access it
this is a class (duh)
the name of this class
opening curly brace of the class
(we'll cover this one later.)
the return type. void means there's no return value.
the name of this method
arguments to the method. This method must be given an array of Strings, and the array will be called 'args'
opening brace of the method
every statement MUST end in a semicolon!!
this says print to standard output (defaults to command-line)
the String you want to print
closing brace of the main method
closing brace of the MyFirstApp class

Data Types in Java

- Primitive Data type or Intrinsic or built-in data type
- Non-Primitive Data type or derived or reference data type



Primitive Data Types in Java

- There are 8 primitive types: 6 of those refer to numbers.
  - 4 for integers types,
  - 2 for floating-point types,

- Character type is used for characters in Unicode encoding.
- Boolean type for true or false values.

- **Java is a statically-typed language**, which means that **the type of a variable** must be **declared before it is** used.

Primitive Data (Auto initialized)

Data Type	Default Value
byte	0
short	0
int	0
long	0L
float	0.0F
double	0.0D
char	'\u0000'
boolean	false

Primitive Data Types (size)

Type	Size	(from)	Range (to)
boolean	1 bit	-	-
char	16 bits Unicode	‘\u0000' (or 0)	'\uffff' (or 65,535 inclusive)
byte	1 byte	-128	127
short	2 bytes	-32,768	32, 767
int	4 bytes	-2,147, 483,648	2,147, 483,647
long	8 bytes	-9,223,372,036,854,775, 808	9,223,372,036, 854, 775,807
float	4 bytes	3.4e-038	3.4e+038
double	8 bytes	1.7e-308	1.7e+308

Non-Primitive Wrapper Class for Primitive Types

- They convert primitive data types into objects.
- Autoboxing - procedure of converting a primitive value into an object of the corresponding wrapper
- Contains type conversion methods

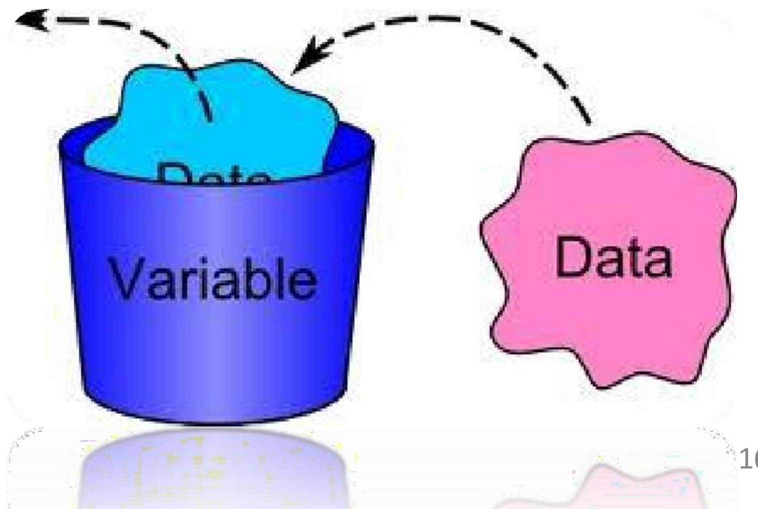
```
int x = 5;
float y = 3.14f;
long z=6000;

// autoboxing
Integer intobj = x;
Float floatobj=y;
Long longobj=z;
```

Primitive type	Wrapper Class
boolean	Boolean
byte	Byte
char	Character
float	Float
int	Integer
long	Long
short	Short
double	Double

Variables

- A variable that has locations in the memory that can hold values.
- Names should be descriptive to improve readability.
- Descriptive but not too long
- Can use standard short names for temporary "throwaway" variables:

  i, k, x, y, str



Variable declaration

- All variables have three important attributes.
    - Data type
    - Name (identifier)
    - Value
- Java is a Strongly-typed language. That means, every variable must be declared as a type.

Activity – What are valid variable names?

1. average
2. height
3. 123
4. pass_mark
5. (area)
6. Total
7. Mark
8. sum1
9. 25th%

Variable declaration and initialization

- When declaring a variable in Java, you need to specify:

  The type

  The name

  The value (optional) => initialization

```
            <data type> <variable name>;
            <data type> <variable name> = <value>;
```

  Declaration

```
   int number;
```

  Initialization

```
   number= 25;
```

  Declaration and initialization

```
   int number=25;
```

Variable declaration (creation):

- Examples
  - `int number = 30;`
  - `char letter = ‘a’;`
  - `boolean ready = true;`

- Always use informative variables names:
  - `int a = 10;char b = ‘a’;` //No Syntax errors but not informative
  - `int number = 10;`
  - `boolean response= true;` //Informative and meaningful

3.1 Integral literals

- byte, short, int, long

  Can be expressed using,

  - Decimal: Base 10, digits 0 - 9
  - Octal: Base 8, digits 0 – 7
  - Hexadecimal: Base 16, digits 0 - 9 & A – F

```
int decVal = 26; // number 26, decimal
int octVal = 032; // number 26, octal
int hexVal = 0x1a;// number 26, hexadecimal
```

The prefix 0 indicates octal, whereas 0x indicates hexadecimal

3.2 Floating point literals

- float, double
- Can be expressed even in scientific notation.
- double is the default data type, not float, therefore, special character is optional.

```
double d1 = 123.4;
double d2 = 1.234e2; //same value

float f1 = 123.4f;   //special suffix needed
```

3.3 Character Literals

- char
  - Any Unicode (UTF-16) character.
  - Use 'single quotes' for `char` literals.

```
    'A'       'x'       '3'       '?'       ' '
```

‘H’ is a `char` constant.

“H” is a String that happens to only contain a single character--it is not a `char`.

3.4 String Literals

- Sequence of characters separated by double quotes.
- **“green”** **“Washington, D.C. 2005”** **“270-32-3456”** **“2*(I+3)/j”**
- Declaration:

  String name = “Alan”;
- Find the length of a string : **int length = name.length();**
- Convert to upper case: **String upper_case = name.toUpperCase();**
- Convert to lower case: **String lower_case = name.toLowerCase();**
- Concatenation: **String twice = name.concat(name);**
- Numbers and strings: be careful when concatenating numbers and strings.
- Special characters: You can print special characters using the escape character “\”.

3.4 String Literal Escape Sequences

Character	Escape Sequence
Backspace	`\b`
Tab	`\t`
New line (line feed)	`\n`
Form feed	`\f`
Carriage return	`\r`
Quotation quote (“)	`\”`
Single quote (‘)	`\’`
Backslash (\)	`\\`

Non printing characters, Always begins with a backward slash

3.5 Boolean Literals: boolean

- A Boolean is a non-numerical primitive data type.
- A Boolean can only have two values: true or false(logical values)
- Declaration and initialization:

```
boolean istrue  = true;
boolean isfalse = false;
```

- A Boolean variable declared with no initialisation will have a default value of false.

```
boolean myboolean; // this variable is false
```

Output Methods

- To print an output, we use the class System: System.out
- System.out.println(); System.out.print(); System.out.printf();

```
System.out.println(3+4);    // 7
```

We need to be careful when printing numbers and characters together:

System.out.print("00" + 3 + 4) //concatenation 007 or 0034 ?

- print() method - Print on the same line contagiously
- println() method - Print a line & move to next line
- printf() method - Print with variable embeddings

Input methods

- When you type a value in a program, to retrieve it, you can use the in object of the System package:
  System.in
- After getting that value, you must first store it somewhere.
- One of the classes you can use is called Scanner.
- Before using the Scanner class, you must import the java.util.Scanner package into the program.
- You can then get values interactively through keyboard

Input methods

- This would be done by writing the following in the top section of the file:
  - **import java.util.Scanner;**
- To use the **Scanner** class to retrieve a value form input stream, use the following formula:
  - **Scanner scanner = new Scanner(System.in);**
- Afters declaring a **Scanner** class, its variable is ready to receive the value.
  - **String myline = scanner.nextLine()**
- If you need to read word by word use next().
  - **String word1 = scanner.next ()**

Input methods

We use inputs to input data in our program (e.g., keyboard).

To input using the keyboard, we use the class System: System.in

We also need to import a Scanner from the package java.util.

Example:

Import util

Create a Scanner

```
import java.util.*;

public class Hello {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Insert your name:");
        String name = input.next();
        String surname = input.next();
        System.out.println("Hello: " + name + “ ” + surname);
    }
}
```

Use Scanner

Use Scanner again

Input methods

- When you want to read a specific data value from the user, you can use the type that you need.

```
String word = input.next();

String sentence = input.nextLine();

int integer = input.nextInt()

double height = input.nextDouble();
```

Summary

- Java program is a collection of tokens, comments and white space.
- There are eight primitive data types: 6 of those refer to numbers.
- There are five basic types of literals in Java.
- Variable Declaration, Assignment and Initialization is important.