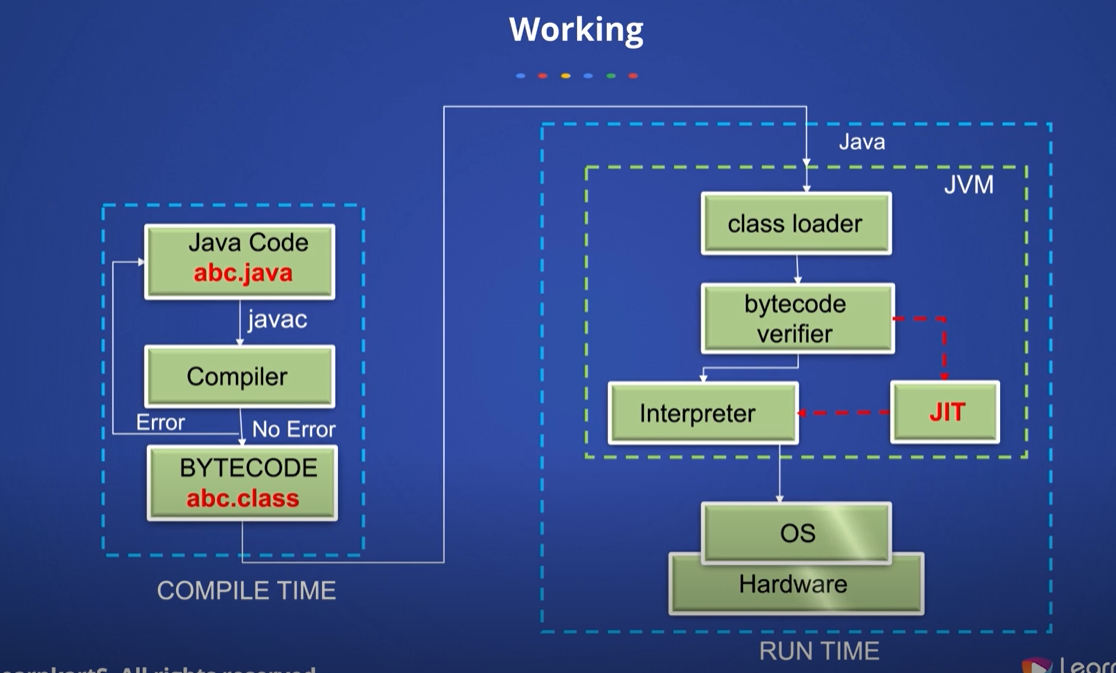
Java platform independent, very versatile

java is portable = we can tranfer code from one window to unix or any others through by

2 high level programming styles

1.procedural programming = break the code into multiple independent functions and make the code talk to each other

2.object-oriented programming = focus on data first rather than functionality



Modern compiler design

Java virtual machine (JVM)

Java runtime environment (JRM)

Java development toolkit (JDT)

OBJECT ORIENTED PRINCIPLES

1.polymorphism = one interface and many forms

Ex one guy can do many things ex can marry , can do work etc

Overloaded method = any binding happens during compile time

Overriding method = any biding happens during execution time

2.abstraction = only gives required details rather than how internal things works

3.Encapsulation = binding together data and functions related to the data

When some other class wants to talk, there is a way to prevent the access

so that only class function can access

4.inheritance= allow one class to inherit the properties or behaviors from another class

Single level inheritance

Multi level inheritance

Multiple inheritance

Hierarchical inheritance

Hybrid inheritance

5.class = class is a structure which consist of certain attributes.

6.object = anything that can be independently or uniquely identified by itself.

Basic elements of Java

1.Unicode Characters = any character which is used in code that has an internal representation

Lowest value=\u0000 highest value=\uFFFF

2.Variables = variable is allocation of a memory location with a name

Syntax = data type variable [ = value] [, variable [ = value] …] ;

Local variable: variable which is specific to a method

Instance variable: while creating an object, there is a data that data part is called instance variable

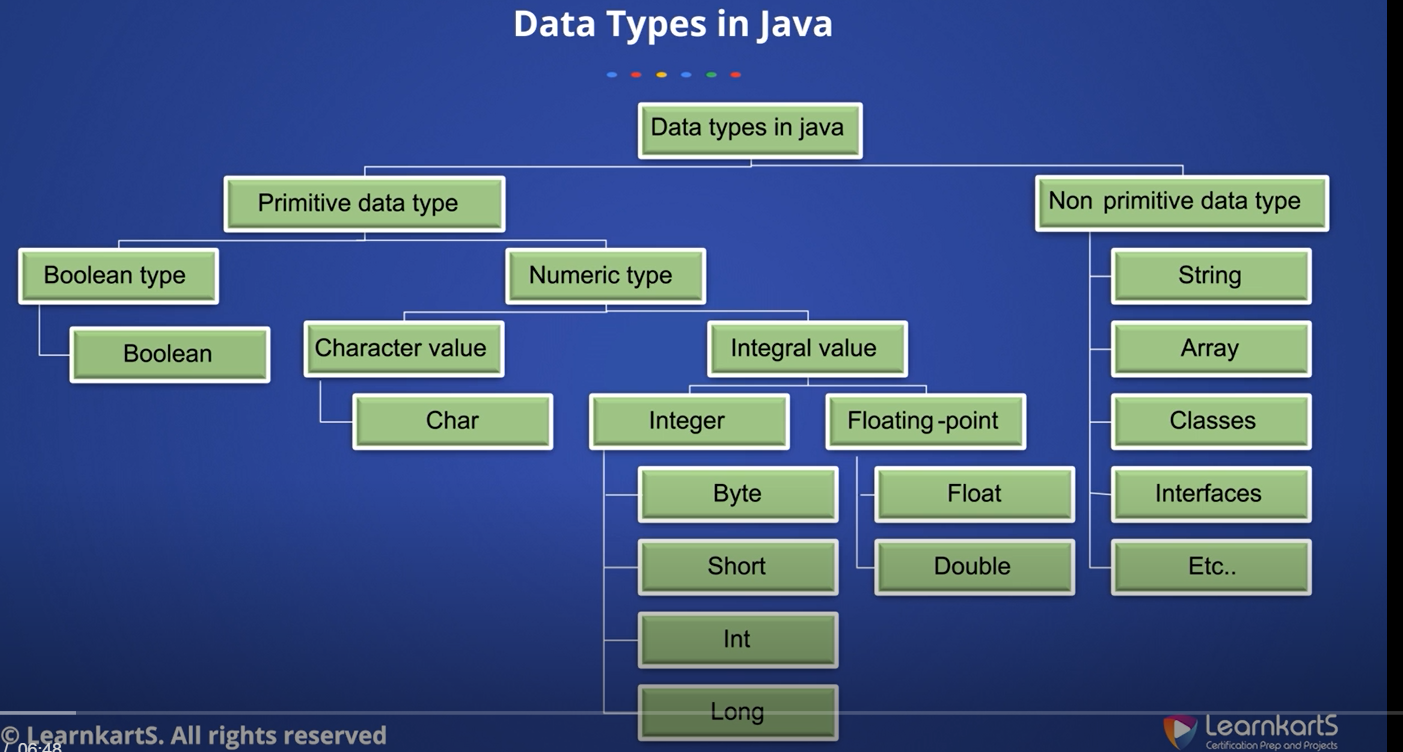
Static variable: common variable across multiple instances of same class

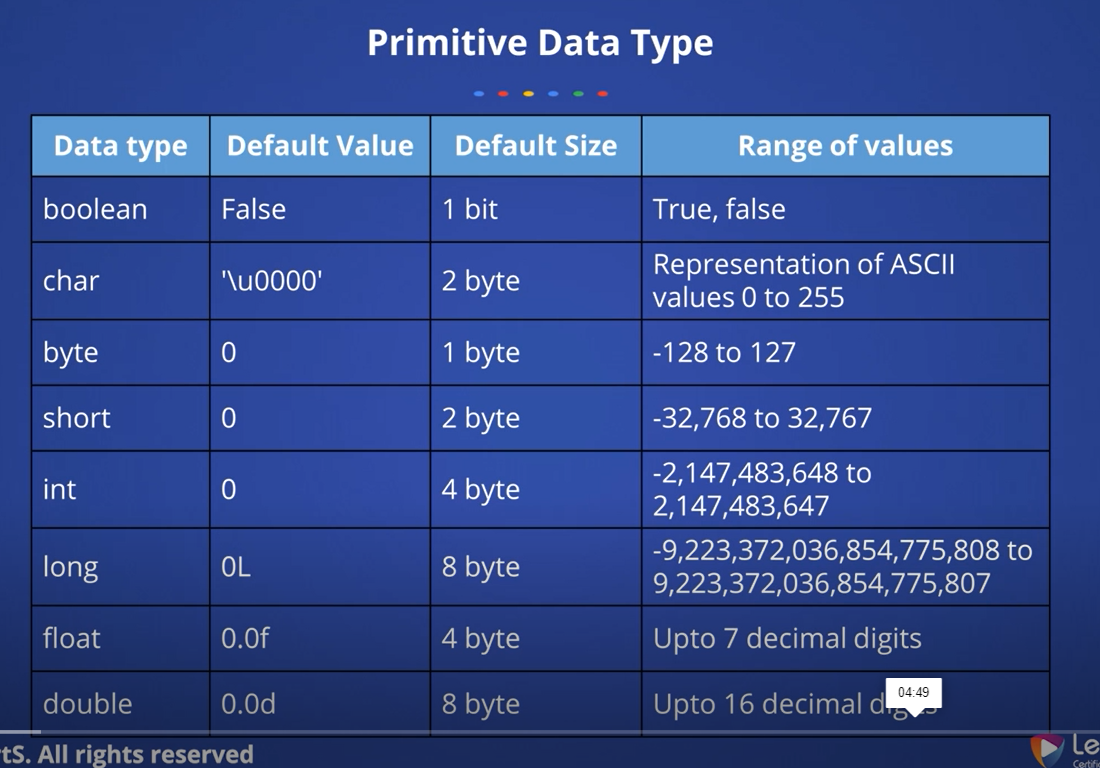
These cannot declare inside method

Types of data types in java

Primitive data type

Non Primitive data type





Constants = value cannot be change once it is been assigned

Static final 🡺datatype identifier\_name=value;

Set of constant values use enum

Literals: is a source code representation of a fixed value

Conventions: set of values to follow as best practice

Operators

Unary operators: require only one operand constant / variable

E.g.: i++, ++i, -1

Arithmetic operators

Shift operators = used to shift the bits of a number left or right

Relation operators = equal to, greater than

Bitwise operators = checks both condition in and decide

M1=60= 0011 1100

M2=13= 0000 1101

M1&M2 =0000 1100 = AND OPERATOR = BOTH SHOULD BE 1

M1IM2 = 0011 1101 = OR OPERATOR = ATLEAST ONE SHOULD BE 1

M1^M2=

~M

LOGICAL OPERATORS = USED FOR LOGIC OPERATIONS 🡺 checks one condition in and decides

&&

ll

!

Ternary Operators= has three operands

Condition? if true: if false

Assignment Operator

Any operator which assigns a value on right hand side to left hand side operator

Type Casting

Converting any primitive datatype into another

Widening = smaller datatype to a bigger datatype

Byte -> Short -> Int -> Long -> float - > Double

Narrowing = bigger datatype to a smaller datatype

Conditional Statement =

If statement 🡺1 possibility

If-else statement 🡺2 possibility

If-else-if ladder🡺 more than 2 possibilities

Nested if statement 🡺 for every step we have check possibility

Loops = any entity that repeats set of statements again and again

For = finite/ infinite

While = finite and infinite

Do-while = execute attest once

Access specifier

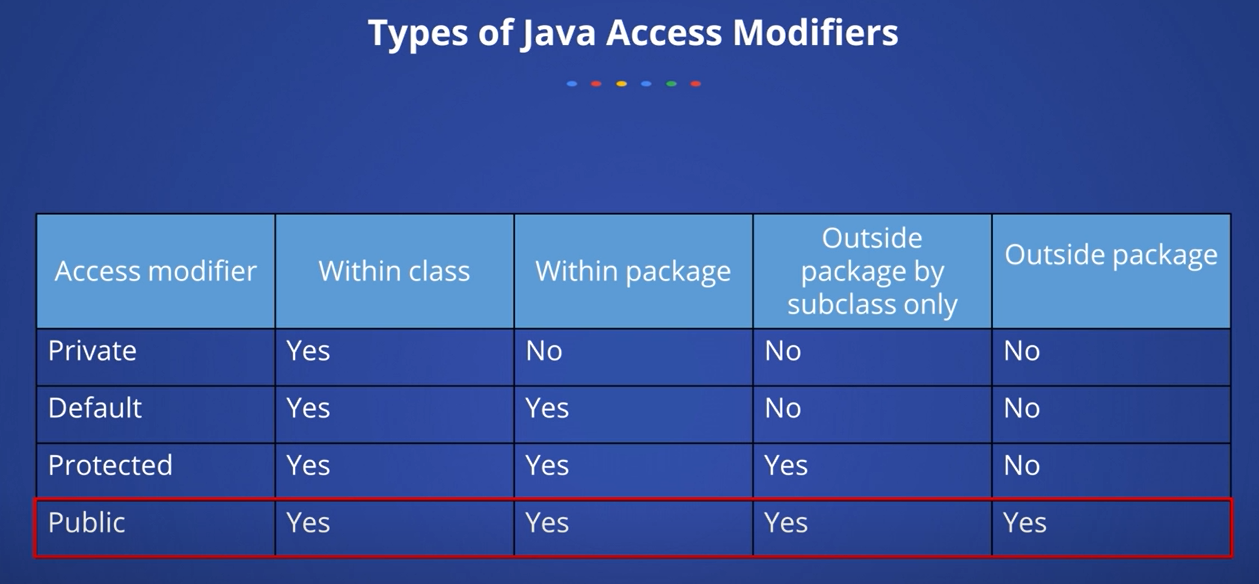
Accessibility or scope of a field, method, constructor or class

Public = access level everywhere

Private = access level is only within the class

Protected = outside package by subclass only through child class

Default = access within the package



Null

This = this represents the class which is getting called

This means current object

Instance of= checks whether a particular variable or operand (constant/variable) is it a part of the data type checking

Destructor = Destructor helps to de- allocate memory

Garbage collector = any value in the memory which is not referenced by any object or variable or any identifier

String = represents a sequence of characters

Arrays= collection of similar types of elements which has contiguous memory location

Single dimension 🡺datatype [] arr;

Multidimension 🡺 datatype [] [] arr;

For each loop: only be used for finite number of elements

for (datatype variable: array | collection) { }

Method overloading 🡺which has same method name

🡺 work🡺by changing the data type & by changing number of parameters

Call by value 🡺taking a copy of data and dealing with it & don’t have impact on original values

Call by reference 🡺 directly dealing with memory address where value is stored & impact on original values

Types of parameters

Formal parameter = which is declared in method like a, b

Actual parameter= which is declared real values

Variable argument’s function or methods which is going to receive any number arguments using …

Object Type Data casting

Primitive🡺 one datatype to another

Reference 🡺one object adress to another

Upcasting 🡺child to parent

Downcasting🡺parent to child

Method overriding

From superclass redeclaring same method in subclass

Method in parent class used to override 🡺overridden method

Method in child class 🡺overriding

Lambda Expression🡺 class interface = (a,b) ->(a+b)

Static variable and method belongs to class

Abstract Classes 🡺no code implementation 🡺 abstract before 🡺only extend we can’t instantiate

Inner class 🡺 class inside a class

Exception handing 🡺 nothing but error

Try block🡺 catch 🡺 finally

throw 🡺 we can throw exception which is catched by catch block

final🡺can’t change the value, methods, classes

Multithreading🡺 Multithreading is a process of running several threads at the same time.

Parrell execution of tasks

Thread vs process 🡺is light weight shared memory vs is heavy weight individual memory

2 ways of creating thread🡺We can extend Thread and also implement runnable

Thread scheduling is nothing but which one to run first

Wrapper classes🡺converting primitive to object

String 🡺 immutable🡺can’t change value

String🡺muttable🡺can change value