MANOJ MORE

C-DAC MUMBAI

JAVA

DAY 2

AUGUST 2025

TYPE CASTING

- The process of converting the value of one data type (int, float, double, etc.) to another data type is known as typecasting.
- In Java, there are 13 types of type conversion. (https://docs.oracle.com/javase/specs/jls/se10/html/jls-5.html)
    - Widening Type Casting
    - Narrowing Type Casting

WIDENING TYPE CASTING

- In Widening Type Casting, Java automatically converts one data type to another data type.

```
int num = 10;
double data = num;
```

Implicit Type Casting

TYPE CONVERSION FROM INT TO STRING

```
int num = 10;
String data = String.valueOf(num);
```

TYPE CONVERSION FROM STRING TO INT

```
String data = "10";
int num = Integer.parseInt(data);
```

NumberFormatException

NARROWING TYPE CASTING

- In Narrowing Type Casting, we manually convert one data type into another using the parenthesis.

```
double num = 10.99;
int data = (int)num;
```

Explicit Type Casting

BOXING AND UNBOXING

IN JAVA

MANOJ MORE

2025

JAVA WRAPPER CLASS

- The wrapper classes in Java are used to convert primitive types (int, char, float, etc) into corresponding objects.

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

Advantages of Wrapper Classes

- In Java, sometimes we might need to use objects instead of primitive data types. For example, while working with collections.
- We can store the null value in wrapper objects.

JAVA AUTOBOXING - PRIMITIVE TYPE TO WRAPPER OBJECT

In autoboxing, the Java compiler automatically converts primitive types into their corresponding wrapper class objects

```
int a = 56;

// autoboxing
Integer aObj = a;
```

Autoboxing has a great advantage while working with Java collections.

```
List<Integer> list = new List<>();

//autoboxing
list.add(5);
list.add(6);

System.out.println("ArrayList: " + list);
```

JAVA UNBOXING - WRAPPER OBJECTS TO PRIMITIVE TYPES

In unboxing, the Java compiler automatically converts wrapper class objects into their corresponding primitive types.

```
// autoboxing
Integer aObj = 56;

// unboxing
int a = aObj;
```

Like autoboxing, unboxing can also be used with Java collections.

```
ArrayList<Integer> list = new ArrayList<>();

//autoboxing
list.add(5);
list.add(6);

System.out.println("ArrayList: " + list);

// unboxingint a = list.get(0);
System.out.println("Value at index 0: " + a);
```

Here, the get() method returns the object at index 0. However, due to unboxing, the object is automatically converted into the primitive type int and assigned to the variable a.

JAVA KEYWORDS

Method	Description
abstract	A non-access modifier.
assert	For debugging
boolean	A data type that can only store true or false values
break	Breaks out of a loop or a switch block
byte	A data type that can store whole numbers from -128 and 127
case	Marks a block of code in switch statements
catch	Catches exceptions generated by try statements
char	A data type that is used to store a single character
class	Defines a class
const	Defines a constant. Not in use - use final instead
continue	Continues to the next iteration of a loop
default	Specifies the default block of code in a switch statement
do	Used together with while to create a do-while loop
double	A data type that can store fractional numbers from 1.7e−308 to 1.7e+308

JAVA KEYWORDS

Method	Description
else	Used in conditional statements
enum	Declares an enumerated (unchangeable) type
extends	Extends a class (indicates that a class is inherited from another class)
final	A non-access modifier used for classes, attributes and methods, non-changeable
finally	Used with exceptions, a block of code that will be executed no matter of an exception or not
float	A data type that can store fractional numbers from 3.4e−038 to 3.4e+038
for	Create a for loop
goto	Not in use, and has no function
if	Makes a conditional statement
implements	Implements an interface
import	Used to import a package, class or interface
instanceof	Checks whether an object is an instance of a specific class or an interface
int	A data type that can store whole numbers from -2147483648 to 2147483647
interface	Used to declare a special type of class that only contains abstract methods

JAVA KEYWORDS

Method	Description
long	Store whole numbers from -9223372036854775808 to 9223372036854775808
native	Specifies that a method is not implemented in the same Java source file (but another language)
new	Creates new objects
package	Declares a package
private	An access modifier used for attributes, methods and constructors, making them only accessible within the declared class
protected	An access modifier used for attributes, methods and constructors - accessible in the same package and subclasses
public	An access modifier used for classes, attributes, methods and constructors, making them accessible by any other class
return	Finished the execution of a method, and can be used to return a value from a method
short	A data type that can store whole numbers from -32768 to 32767
static	A non-access modifier. Static methods/attributes can be accessed without creating an object of a class
strictfp	Obsolete. Restrict the precision and rounding of floating point calculations
super	Refers to superclass (parent) objects
switch	Selects one of many code blocks to be executed
synchronized	A non-access modifier, which specifies that methods can only be accessed by one thread at a time

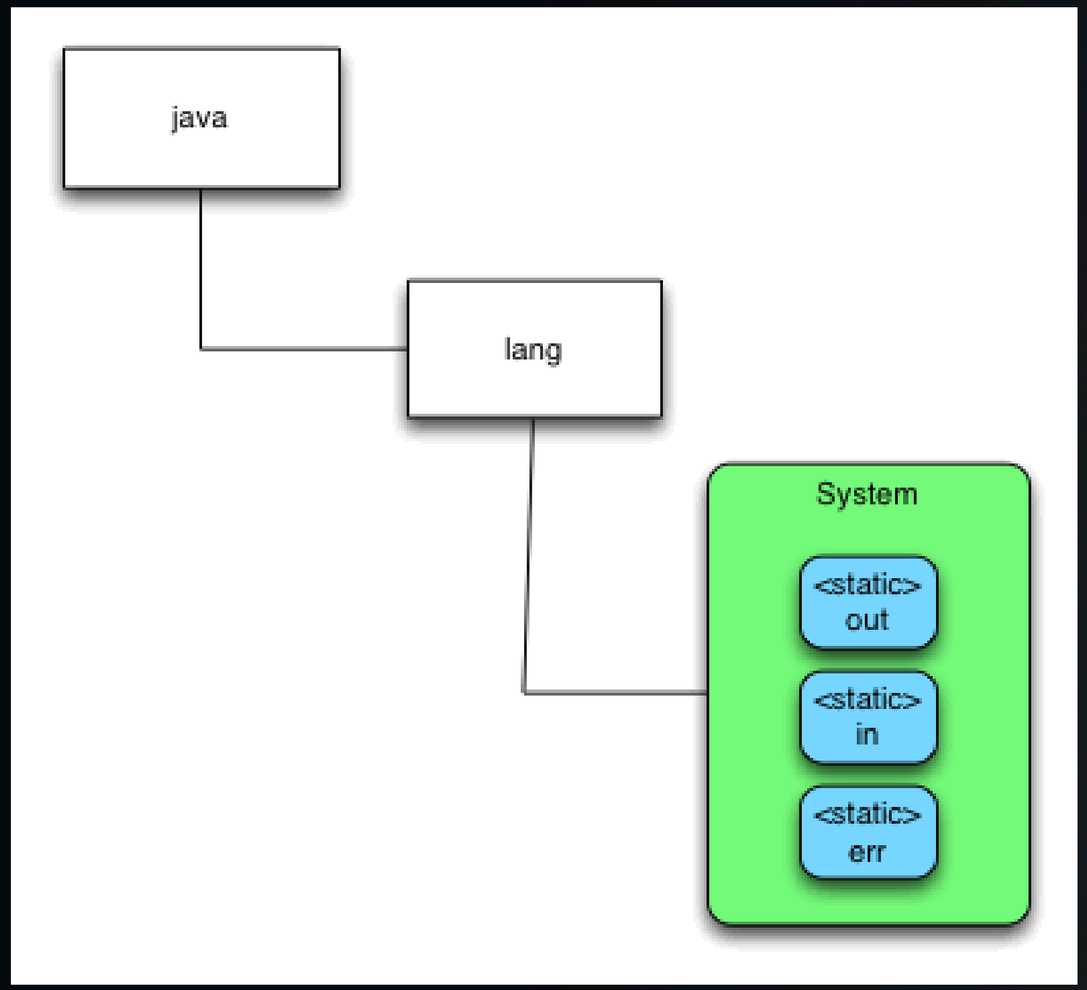
JAVA KEYWORDS

Method	Description
this	Refers to the current object in a method or constructor
throw	Creates a custom error
throws	Indicates what exceptions may be thrown by a method
transient	Used to ignore an attribute when serializing an object
try	Creates a try...catch statement
void	Specifies that a method should not have a return value
volatile	Indicates that an attribute is not cached thread-locally, and is always read from the "main memory"
while	Creates a while loop

True, False, and Null are not keywords, but they are literals and reserved words that cannot be used as identifiers.

SYSTEM CLASS IN JAVA

- System class in Java is present in java.lang package.
- This class is derived from the Object class.



JAVA OPERATORS

Java operators are special symbols that perform operations on variables or values.

Arithmetic Operators

* : Multiplication
/ : Division
% : Modulo
+ : Addition
- : Subtraction

Unary Operators

- , Negates the value.
+ , Indicates a positive value (automatically converts byte, char, or short to int).
++ , Increments by 1.
-- , Decrements by 1.
! , Inverts a boolean value.

JAVA OPERATORS

Java operators are special symbols that perform operations on variables or values.

Assignment Operator

+= , Add and assign.
-= , Subtract and assign.
*= , Multiply and assign.
/= , Divide and assign.
%= , Modulo and assign.

Relational Operators

== , Equal to.
!= , Not equal to.
< , Less than.
<= , Less than or equal to.
> , Greater than.
>= , Greater than or equal to.

JAVA OPERATORS

Java operators are special symbols that perform operations on variables or values.

Logical Operators

&&, Logical AND: returns true when both conditions are true.
||, Logical OR: returns true if at least one condition is true.
!, Logical NOT: returns true when a condition is false and vice-versa

Ternary operator

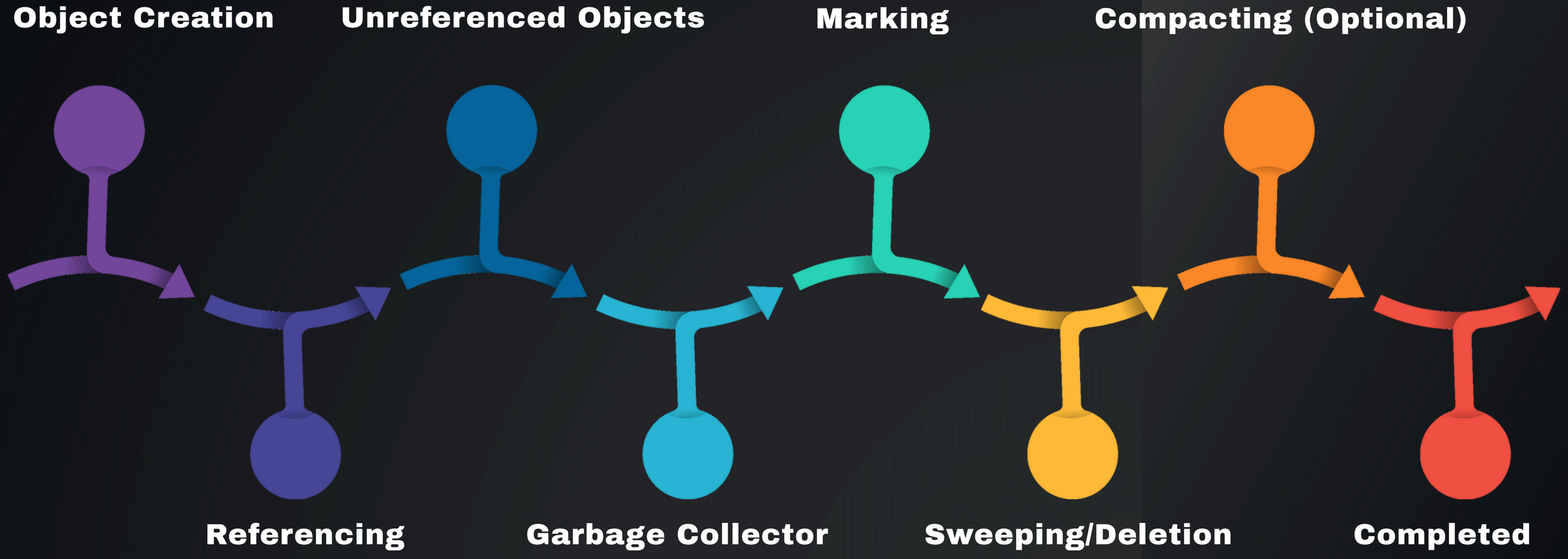
The Ternary Operator is a shorthand version of the if-else statement.
It has three operands and hence the name Ternary. The general format is,

condition ? if true : if false

LIFE CYCLE OF OBJECT IN JAVA

- **Step 1:** Creation of .class file
- **Step 2:** Loading .class file into memory
- **Step 3:** Looking for initialized static members of class
- **Step 4:** Allocation of memory for object and reference variable
- **Step 5:** Calling of the constructor of class
- **Step 6:** Removing of object and reference variable from memory

GARBAGE COLLECTION (GC) IN JAVA



CALLING THE GC

Using System.gc()

Using Runtime.getRuntime().gc()

The finalize() Method