

## Unit-1

\* who developed Java and in which year?

Ans: Java was developed by James Gosling at Sun Microsystems in 1995.

\* what was Java originally called?

Ans: Oak.

\* what are the key features of Java?

Ans: Java is simple, object-oriented, platform-independent, secure, robust.

\* what is the difference b/w JDK, JRE and JVM?

Ans: i. JDK (Java Development Kit):- Tools for developing Java program

ii. JRE (Java Runtime Environment):- Environment to run Java programs.

iii. JVM (Java Virtual Machine):- Converts bytecode into machine code

\* what is Bytecode?

Ans: Bytecode is an intermediate code generated by the Java compiler which is platform independent and executed by the JVM.

\* what is platform independence in Java?

Ans: Java program can run on any operating system with a JVM, making it platform-independent.

\* Why is the main() method important in Java?

Ans: The main() method is the entry point of a Java application

\* List all the primitive data types in Java?  
Ans:- byte, short, int, long, float, double, char, boolean.

\* what is the default value of boolean variable?

Ans:- false.

\* what is the difference b/w int and Integer?

Ans:- int is a primitive data type, Integer is a wrapper class for int.

\* what are auto-boxing and unboxing?

Ans:- Auto-boxing :- Converting a primitive to a wrapper.

i. Unboxing :- Converting a wrapper to a primitive.

\* what is type Casting?

Ans:- Converting one data type into another.

\* what are identifiers in Java?

Ans:- Identifiers are the names given to classes, methods, variables etc..

\* Explain the word Static keyword.

Ans:- It is used to define class-level variables and methods which can be accessed without creating an object.

\* what are access modifiers?

Ans:- private : (class only)

default (same package)

protected (same package + subclass)

public (anywhere)

\* what are wrapper Class in Java?

Ans: A class that converts primitive types into objects

\* what are bitwise operators?

Ans: &(AND), |(OR), ^(XOR), ~(NOT), <<(leftshift), >>(right shift).

\* Difference b/w  $\text{==}$  and  $=$

Ans:  $\text{==}$  → checks equality  
 $=$  → assignment operator.

\* what is operator precedence?

Ans: The order in which operators are evaluated in an expression.

\* what is the difference b/w "if-else" and "switch".

Ans if-else → handles conditions.

switch → handles multiple constant values more cleanly.

## Unit - 2

\* what is an array ?

Ans:- A container object that holds multiple values of the same data type .

\* How do you declare and initialize an array ?

Ans:- `int[] arr = new int[5];` → 1D

`int[][] Matrix = new int[3][3]` → 2D.

\*

\* what is an enumeration(enum)?

Ans:- A special class that defines a collection of constants.

\* what is a class ?

Ans:- A blue print for creating objects , it defines properties and behaviors.

\* what is an object ?

Ans:- An instance of class.

\* what is a constructor ?

Ans:- A special method used to initialize objects.

\* what is method overloading ?

Ans:- Having multiple methods with the same name but different parameters.

\* what is constructor Overloading?

Ans: Defining multiple constructor in the same class with different parameters lists.

\* what is the use of this keyword?

Ans: it refers to the current object.

\* what are initializer blocks?

Ans: Blocks of code inside a class that are executed when an object is created, before the constructor.

\* Difference b/w string and String builders?

Ans: String is immutable (Cannot be changed after creation).

StringBuilder is mutable (Can be modified).

\* How do you use String Builder?

Ans  
StringBuilder sb = new StringBuilder("Hello");  
sb.append("world");  
System.out.println(sb);

## Unit -3

\* what is inheritance ?  
(Child/Subclass)

Ans:- inheritance allows one class to acquire the properties and methods of another class (parent /superclass)

\* what is method overriding?

Ans:- when a subclass provides a specific implementation of a method already defined in its superclass.

\* what is the use of the Super keyword?

Ans:- "Super" is used to refer to the immediate parent class object and constructor.

- \* To call the parent class method
- \* To call the parent class constructor

\* what is the object class in java?

Ans:- It is the root class of all Java classes. Every class inherits from object.

\* what is the "final" keyword used for?

Ans:- Final variable:- Value cannot be changed

Final method:- Cannot be overridden

Final class :- Cannot be inherited.

\* what is the "instanceof" operator?

Ans:- It checks whether an object is an instance of a specific class or subclass.

\* what is abstract class?

Ans:- A class declared with the "abstract" key word which may contain abstract method (methods without a body).

\* Can we create an object of an abstract class?

Ans:- No, abstract classes cannot be instantiated directly.

\* what is an abstract method?

Ans:- A method without a body, declared in an abstract class.

\* what is an interface in java?

Ans:- An interface is a reference type in java that can contain abstract method and constants.

\* what are static and default methods in interface?

Ans:- Default method:- A method with a body inside the interface using the "default" keyword.

Static method:- A method inside the interface with the "static" keyword, accessible without an instance.

## Unit -4 .

\* What is a nested class ?

Ans:- A class defined within another class.

\* What are the types of nested classes?

Ans:- \* Static nested class

\* Non-static <sup>(inner)</sup> ~~nested~~ class

\* Local class (inside a method)

\* Anonymous class.

\* What is a static nested class ?

Ans:- A nested class that is declared static and can be accessed without an object of the outer class.

\* What is a non-static inner class ?

Ans:- A class inside another class that requires an instance of the outer class.

\* What is a local class ?

Ans:- A class defined within a method.

\* What is an anonymous class ?

Ans:- A class with no name defined and instantiated at the same time, usually for overriding methods quickly.

\* What is a functional interface ?

Ans:- An interface with only one abstract method

\* what are Lambda Expression?

Ans: A shorthand way of writing anonymous classes for functional interface.

\* How do you work with dates in java?

Ans: using classes like "java.util.Date", "java.util.Calendar", and "java.time" package

\* How to get current date?

Ans: import java.time.LocalDate

```
LocalDate date = LocalDate.now();
```

\* what is an expression in java?

Ans: An event that disrupts the normal flow of the program.

\* what is exception propagation?

Ans: Passing an exception from one method to another until it is ~~not~~ handled.

\* what is the purpose of try-catch-finally blocks?

Ans: try :- Code that might throw an exception.

Catch:- code to handle the exception.

finally:- code that execute regardless of exception

\* what is the difference b/w throw and throws?

Ans: throw :- used to actually throw an exception.

throws:- used in method Signatures to declare possible exceptions.

\* what is try-with-resources?

Ans: A feature that automatically closes resources after use.

\* How do you create a custom ~~exception~~ exception?

Ans: By extending the "Exception" class.

\* what are assertion?

Ans: A debugging aid to test assumption during development.

\* checked & unchecked exception?

Ans: checked :- must be handle using try-catch, throw..

unchecked :- occurs at a runtime and are not required to be handled.

\* what is an exception?

Ans: An event that occurs during the execution of a program that, disrupts the normal flow of instructions.

## Unit - 5.

\* what is I/O in java?

Ans:- I/O (Input and output) in java refers to reading data from

Source (like files, keyboard) and writing data to destination  
(like file, console).

\* what are Java I/O streams?

Ans:- Streams represent sequence of data. Java uses streams to perform input and output operations.

i. InputStream and Reader classes for reading.

ii. OutputStream and writer classes for writing.

\* what is the difference b/w byte streams and character stream?

Ans:- i. Byte streams (InputStream/OutputStream) handle raw binary data.

ii. Character streams (Reader/Writer) handle character data (unicode).

\* How do you read a file in java?

Ans:- BufferedReader br = new BufferedReader(new FileReader("File.txt"));

String line = br.readLine();

br.close();

\* How do we write to a file in java?

Ans:- BufferedWriter bw = new BufferedWriter(new FileWriter("File.txt"));

bw.write ("Hello world");

bw.close();

\* what is Serialization ?

Ans:- Converting an object into a byte stream to save it to a file or send over a network.

\* How do you Serialize an object ?

Ans:- The class must implement "Serializable" interface.

• use " ObjectOutputStream "

\* Deserialization - is a process of converting a file (InputStream) back into Java object

\* what are generics in Java ?

Ans:- Generics enable types (class and methods) to operate on objects of various types while providing compile-time type safety.

\* what is the type inference diamond operator "<>" ?

Ans:- It allows the compiler to infer the types automatically without repeating it.

Ex:- Box<Integer> box = new Box<>();

(No need to write new Box<Integer>();).

\* what are bounded types in generics ?

Ans:- Restricting the generic type to a subclass or interface.

\* what are wild cards in generics ?

Ans:- Represents unknown types in generic using "?" .

Types of wildcards:-

i) unbounded wildcard: <?>

→ Accepts Type (or) any of its  
Subclass → child class

ii) upper bounded wildcard: <? extends Type> .

→ Accepts Type (or) any

iii) Lower bounded wildcard: <? Super Type> . of its Superclass.

↓ Parent class

## Unit - 6

\* what is the java Collections Framework?

Ans:- A set of classes and interfaces that store and manipulate groups of data as a single unit.

\* How do you create a collection using Generics?

Ans:- `ArrayList<String> list = new ArrayList<>();`

`list.add("Hello");`

`S.O.P(list);`

Generics ensure type safety (only String values can be added).

\* How do you implement an ArrayList?

Ans:- `import java.util.ArrayList;`

\* ArrayList is a resizable array.

\* It allows duplicate elements and maintains insertion order.

\* How do you implement a TreeSet using "Comparable"?

Ans:- `import java.util.TreeSet;`

`TreeSet<Integer> set = new TreeSet<>();`

`set.add(20);`

`set.add(10);`

`set.add(30);`

`S.O.P(set);`

\* Here, elements are sorted because "Integer" implements "Comparable".

\* Comparable - is used when we want a class to define its own natural sorting order.

\* Comparator - is used when we want to sort objects in a different or custom order.

\* ArrayList - is a dynamic array that grows automatically when elements are added.

\* TreeSet → elements are stored in sorted order.

↳ By default it uses natural sorting (using Comparable).

\* HashMap - Stores key-value pair

↳ key must be unique

```
HashMap<Integer, String> map = new HashMap<>();
```

```
map.put(1, "Apple");
```

```
map.put(2, "Orange");
```

```
map.put(3, "Mango");
```

```
s.o.p(map);
```

\* what is Deque?

Ans → Allows inserting and removing elements from both ends

```
import java.util.ArrayDeque;
```

```
ArrayDeque<String> deque = new ArrayDeque<>();
```

```
deque.addFirst("First");
```

```
deque.addLast("Last");
```

```
s.o.p(deque);
```

\* what is JDBC?

Ans: JDBC (Java Database Connectivity) is an API that helps Java programs connect to and interact with a database (like MySQL etc.)

\* JDBC Drivers?

Ans: Special programs that connects Java code with the database

Types of Drivers:-

Type 1 - JDBC-ODBC bridge

Type 2 - Native API drive

Type 3 - Network protocol driver

Type 4 - Thin drive (pure java, most used today).

\* CRUD operation using JDBC.

Ans: CRUD = Create, Read, Update, Delete - basic database operation

\* Connecting to Non-Conventional Database.

Ans: • Normal JDBC is used with relational database like MySQL.

• Non-conventional database = NoSQL.

• We use special libraries/drivers to connect Java with these databases.

Eg: To connect Java with "MongoDB", we use "MongoDB" Java Driver.

\* what is an Instance Variable?

Ans: Instance variables are variables that belong to an object of a class.

\* Instanceof ?

Ans: The Instanceof operator checks:

"Is this object an object of this class?"

It returns true (or) false.