**Section 1: Java Basics (Q1–Q10)**

1. **Q:** Java के हो?
**A:** Java एक Object-Oriented, platform-independent, high-level programming language हो जसले "Write Once, Run Anywhere" concept प्रयोग गर्छ।
2. **Q:** JVM भनेको के हो?
**A:** JVM (Java Virtual Machine) Java bytecode चलाउने runtime environment हो।
3. **Q:** JDK र JRE बीच के फरक छ?
**A:**
	* **JDK**: Development tools + JRE।
	* **JRE**: JVM + Libraries, केवल run गर्नका लागि।
4. **Q:** Bytecode भनेको के हो?
**A:** Compiler द्वारा Java code बाट generated platform-independent instructions, जुन JVM मा चल्छ।
5. **Q:** Java किन platform-independent हो?
**A:** किनकि Java को compiled bytecode JVM मार्फत कुनै पनि OS मा चल्न सक्छ।
6. **Q:** Java को main method को signature के हो?
**A:** public static void main(String[] args)
7. **Q:** Variable र constant बीच के फरक छ?
**A:** Variable को value change हुन्छ; constant (final keyword प्रयोग गरेर) को value change हुँदैन।
8. **Q:** Wrapper class भनेको के हो?
**A:** Primitive data types लाई object मा wrap गर्ने classes (जस्तै: Integer, Double)।
9. **Q:** == र .equals() बीच के फरक छ?
**A:** == memory reference compare गर्छ; .equals() value compare गर्छ।
10. **Q:** final, finally, र finalize() बीच के फरक छ?
**A:**
	* **final**: variable, method, class मा constant/override रोक्ने।
	* **finally**: exception handling मा cleanup code राख्ने block।
	* **finalize()**: garbage collection अघि object cleanup गर्ने method।

**Section 2: OOP Concepts (Q11–Q20)**

1. **Q:** OOP को चार pillars के हुन्?
**A:** Encapsulation, Inheritance, Polymorphism, Abstraction।
2. **Q:** Encapsulation के हो?
**A:** Data र methods एउटै class भित्र राख्ने र direct बाहिरबाट access रोक्ने।
3. **Q:** Abstraction के हो?
**A:** Implementation details लुकाएर केवल आवश्यक जानकारी देखाउने।
4. **Q:** Inheritance का प्रकार के के हुन्?
**A:** Single, Multilevel, Hierarchical (Java मा multiple inheritance direct support हुँदैन)।
5. **Q:** Method overloading के हो?
**A:** एउटै method नाम, तर parameter list फरक।
6. **Q:** Method overriding के हो?
**A:** Parent class को method लाई child class मा redefine गर्ने।
7. **Q:** Interface र abstract class बीच के फरक छ?
**A:** Interface मा सबै methods abstract हुन्छन्; abstract class मा abstract र non-abstract दुवै हुन्छन्।
8. **Q:** super keyword के हो?
**A:** Parent class को methods, variables, constructors access गर्न प्रयोग हुन्छ।
9. **Q:** Constructor overloading के हो?
**A:** एउटै class मा फरक parameter list भएका multiple constructors बनाउने।
10. **Q:** this keyword को प्रयोग के हो?
**A:** Current object को reference access गर्न प्रयोग हुन्छ।

**Section 3: Strings & Collections (Q21–Q30)**

1. **Q:** String immutable किन हुन्छ?
**A:** Security, performance, thread-safety को लागि।
2. **Q:** StringBuilder र StringBuffer बीच के फरक छ?
**A:** StringBuilder faster तर non-synchronized; StringBuffer slower तर synchronized।
3. **Q:** String pool के हो?
**A:** Heap मा stored unique string literals को special memory area।
4. **Q:** ArrayList र LinkedList बीच के फरक छ?
**A:** ArrayList random access मा fast; LinkedList insert/delete मा fast।
5. **Q:** HashMap र Hashtable बीच के फरक छ?
**A:** HashMap non-synchronized; Hashtable synchronized।
6. **Q:** HashSet duplicate elements किन नलिन्छ?
**A:** किनकि यो hashing mechanism प्रयोग गर्छ।
7. **Q:** Comparable र Comparator बीच के फरक छ?
**A:** Comparable natural ordering define गर्छ; Comparator custom ordering define गर्छ।
8. **Q:** Fail-fast र fail-safe iterator बीच के फरक छ?
**A:** Fail-fast concurrent modification detect गर्छ र exception फ्याँक्छ; fail-safe copy मा iterate गर्छ।
9. **Q:** PriorityQueue के हो?
**A:** Queue जसले elements लाई natural वा custom order मा राख्छ।
10. **Q:** TreeMap र HashMap बीच के फरक छ?
**A:** TreeMap sorted हुन्छ; HashMap unordered हुन्छ।

**Section 4: Exception Handling (Q31–Q40)**

1. **Q:** Checked र Unchecked exception बीच के फरक छ?
**A:** Checked compile-time मा check हुन्छ; Unchecked runtime मा।
2. **Q:** try-catch-finally को flow के हो?
**A:** try → catch (यदि exception) → finally (सधैं execute)।
3. **Q:** throw र throws बीच के फरक छ?
**A:** throw single exception फ्याँक्छ; throws method declaration मा exceptions list गर्छ।
4. **Q:** Custom exception कसरी बनाउने?
**A:** Exception class extend गरेर।
5. **Q:** Multiple catch blocks कसरी काम गर्छ?
**A:** Exception को प्रकार अनुसार पहिलो matching catch block execute हुन्छ।
6. **Q:** finally block exception आए पनि execute हुन्छ?
**A:** हो, तर System.exit() प्रयोग भएमा हुँदैन।
7. **Q:** try-with-resources के हो?
**A:** Resources automatic close गर्ने try statement।
8. **Q:** Error र Exception बीच के फरक छ?
**A:** Error JVM ले handle गर्न सक्दैन (hardware failure जस्तो); Exception handle गर्न सकिन्छ।
9. **Q:** RuntimeException example दिनुहोस्।
**A:** NullPointerException, ArrayIndexOutOfBoundsException।
10. **Q:** SQLException के हो?
**A:** JDBC मा database सम्बन्धित exception।

**Section 5: Multithreading (Q41–Q50)**

1. **Q:** Thread के हो?
**A:** Program को smallest execution unit।
2. **Q:** Multithreading को लाभ के हो?
**A:** Faster execution, better CPU utilization।
3. **Q:** Thread बनाउनका दुई तरिका के हुन्?
**A:** Thread class extend गर्ने वा Runnable implement गर्ने।
4. **Q:** synchronized keyword को काम के हो?
**A:** Multiple threads बाट shared resource सुरक्षित गर्न।
5. **Q:** Deadlock के हो?
**A:** जब दुई वा बढी threads एकअर्काको resource को लागि अनन्त समय पर्खन्छन्।
6. **Q:** Thread-safe class को उदाहरण दिनुहोस्।
**A:** StringBuffer, Vector।
7. **Q:** volatile keyword के हो?
**A:** Variable को value सबै threads मा तुरुन्त update गर्न।
8. **Q:** sleep() र wait() बीच के फरक छ?
**A:** sleep() समय अनुसार रोक्छ; wait() notify/notifyAll बाट resume हुन्छ।
9. **Q:** Thread priority के हो?
**A:** Thread को execution preference level।
10. **Q:** Executor framework के हो?
**A:** Thread pool management को लागि utility classes को set।

**Section 6: Java 8 Features (Q51–Q60)**

1. **Q:** Lambda expression के हो?
**A:** Anonymous function Java मा functional style programming का लागि।
2. **Q:** Functional interface के हो?
**A:** Single abstract method भएको interface (जस्तै Runnable)।
3. **Q:** Streams API के हो?
**A:** Collections मा functional-style operations गर्न API।
4. **Q:** Optional class के हो?
**A:** Null values सुरक्षित रूपमा handle गर्न wrapper।
5. **Q:** Method reference के हो?
**A:** Existing method लाई lambda को shortcut मार्फत reference गर्ने।
6. **Q:** Default method के हो?
**A:** Interface भित्र method body भएको method।
7. **Q:** Predicate, Function, Consumer के हुन्?
**A:** Java 8 को functional interfaces।
8. **Q:** Stream filter र map बीच के फरक छ?
**A:** filter elements remove गर्छ; map transform गर्छ।
9. **Q:** Parallel stream के हो?
**A:** Stream को काम multi-core threads मा execute गर्ने।
10. **Q:** Instant र LocalDateTime के हुन्?
**A:** Java 8 Date-Time API का classes।

**Section 7: Memory Management (Q61–Q70)**

1. **Q:** Garbage collection के हो?
**A:** Unused objects हटाएर memory free गर्ने process।
2. **Q:** finalize() method को प्रयोग के हो?
**A:** GC अघि cleanup गर्न (deprecated in latest Java)।
3. **Q:** Heap र Stack memory बीच के फरक छ?
**A:** Heap मा objects; Stack मा method variables।
4. **Q:** Memory leak Java मा कसरी हुन्छ?
**A:** Objects reachable रहँदा जुन प्रयोगमा छैनन्।
5. **Q:** SoftReference र WeakReference बीच के फरक छ?
**A:** Weak references GC ले तुरुन्त हटाउँछ; Soft references memory critical हुँदा।
6. **Q:** PermGen र Metaspace बीच के फरक छ?
**A:** PermGen fixed size; Metaspace dynamically grow हुन्छ (Java 8+ मा)।
7. **Q:** OutOfMemoryError को कारण के हुन सक्छ?
**A:** Heap overflow, memory leak।
8. **Q:** StackOverflowError के हो?
**A:** Infinite recursion वा excessive stack usage।
9. **Q:** JVM tuning कसरी गरिन्छ?
**A:** JVM options जस्तै -Xmx, -Xms प्रयोग गरेर।
10. **Q:** ReferenceQueue के हो?
**A:** GC ले collected references queue मा राख्छ।

**Section 8: File I/O (Q71–Q80)**

1. **Q:** FileReader र FileWriter बीच के फरक छ?
**A:** FileReader characters पढ्छ; FileWriter characters लेख्छ।
2. **Q:** BufferedReader को लाभ के हो?
**A:** Large data efficiently read गर्न।
3. **Q:** Serialization के हो?
**A:** Object लाई byte stream मा बदल्ने process।
4. **Q:** Deserialization के हो?
**A:** Byte stream लाई object मा बदल्ने process।
5. **Q:** transient keyword को प्रयोग के हो?
**A:** Serialization मा variable skip गर्न।
6. **Q:** Path र Files class के हुन्?
**A:** Java NIO मा file handling को लागि utility classes।
7. **Q:** RandomAccessFile के हो?
**A:** File को arbitrary position मा read/write गर्न।
8. **Q:** InputStream र Reader बीच के फरक छ?
**A:** InputStream bytes पढ्छ; Reader characters पढ्छ।
9. **Q:** OutputStream र Writer बीच के फरक छ?
**A:** OutputStream bytes लेख्छ; Writer characters लेख्छ।
10. **Q:** ObjectOutputStream के हो?
**A:** Objects लाई serialize गरेर file मा लेख्ने।

**Section 9: Advanced Java (Q81–Q90)**

1. **Q:** JDBC के हो?
**A:** Java Database Connectivity API।
2. **Q:** Statement र PreparedStatement बीच के फरक छ?
**A:** PreparedStatement precompiled हुन्छ, सुरक्षित र तेज।
3. **Q:** ResultSet के हो?
**A:** Database query result को table-like representation।
4. **Q:** Connection pooling के हो?
**A:** Database connections reuse गर्ने technique।
5. **Q:** Servlet के हो?
**A:** Java server-side program जसले HTTP requests handle गर्छ।
6. **Q:** JSP के हो?
**A:** Java Server Pages, HTML मा Java embed गर्ने।
7. **Q:** JPA के हो?
**A:** Java Persistence API, ORM को लागि।
8. **Q:** Hibernate के हो?
**A:** Popular ORM framework Java मा।
9. **Q:** RMI के हो?
**A:** Remote Method Invocation, दूरस्थ object call गर्न।
10. **Q:** JMS के हो?
**A:** Java Message Service, asynchronous communication को लागि।

**Section 10: Miscellaneous (Q91–Q100)**

1. **Q:** Enum के हो?
**A:** Fixed set of constants define गर्ने type।
2. **Q:** Annotation के हो?
**A:** Metadata provide गर्ने special tag।
3. **Q:** Reflection API के हो?
**A:** Runtime मा class को जानकारी पत्ता लगाउने API।
4. **Q:** Marker interface के हो?
**A:** बिना method भएको interface (जस्तै Serializable)।
5. **Q:** Singleton pattern के हो?
**A:** केवल एउटै object बनाउने design pattern।
6. **Q:** Factory pattern के हो?
**A:** Object creation encapsulate गर्ने pattern।
7. **Q:** Dependency Injection के हो?
**A:** Object को dependency बाहिरबाट provide गर्ने।
8. **Q:** Immutable class कसरी बनाउने?
**A:** final class, final fields, बिना setter।
9. **Q:** equals() र hashCode() override किन गर्नुपर्छ?
**A:** Hash-based collections मा सही comparison र storage को लागि।
10. **Q:** Java मा best practices के हुन्?
**A:** Naming conventions, code reusability, exception handling, memory management, thread-safety।

**Section 11: String Handling & Logic (Q101–Q120)**

1. String मा uppercase र lowercase count कसरी गर्ने? → Character iterate गरेर ASCII वा Character.isUpperCase() / isLowerCase() प्रयोग गर्ने।
2. String लाई toggle case कसरी गर्ने? → प्रत्येक character को case उल्टाउने logic प्रयोग गर्ने।
3. Built-in reverse method बिना String उल्टाउने तरिका? → Loop backwards गरेर नयाँ string बनाउने।
4. String मा special characters हटाएर केवल अंक र अक्षर राख्ने? → replaceAll("[^a-zA-Z0-9]", "") प्रयोग गर्ने।
5. String मा केवल alphabet र space छन् कि छैनन् check गर्ने? → matches("[a-zA-Z ]+") प्रयोग गर्ने।
6. String rotation check गर्ने तरिका? → एक string लाई double गरेर contains() check गर्ने।
7. Recursion प्रयोग गरेर palindrome check गर्ने? → First र last character compare गर्दै recursion call गर्ने।
8. प्रत्येक शब्दको पहिलो अक्षर uppercase बनाउने? → Split गरेर capitalize गर्ने।
9. String मा केवल पहिलो occurrence replace गर्ने? → replaceFirst() प्रयोग गर्ने।
10. Duplicate words find गर्ने? → Map प्रयोग गरेर word frequency निकाल्ने।
11. String लाई title case बनाउने? → Split गरेर पहिलो अक्षर capitalize गर्ने।
12. Word reverse गर्ने तर punctuation position same राख्ने? → Non-letter characters skip गर्ने।
13. String लाई char array मा बदल्ने तर space skip गर्ने? → replace(" ", "").toCharArray() प्रयोग गर्ने।
14. Comma separated values लाई array मा बदल्ने? → split(",") प्रयोग गर्ने।
15. String को पहिलो र अन्तिम character swap गर्ने? → StringBuilder प्रयोग गरेर।
16. Character को ASCII value print गर्ने? → (int) char cast गर्ने।
17. दुई characters को position swap गर्ने? → Char array मा बदल्ने र swap गर्ने।
18. Palindrome subsequence count गर्ने? → Recursion वा dynamic programming प्रयोग गर्ने।
19. Balanced brackets check गर्ने? → Stack प्रयोग गर्ने।
20. String लाई Morse code मा बदल्ने? → Map बनाएर mapping गर्ने।

**Section 12: Number & Math Problems (Q121–Q140)**

1. Bitwise method प्रयोग गरेर prime check गर्ने? → √n सम्म loop गर्ने।
2. Next prime number find गर्ने? → N+1 बाट सुरु गरेर prime check गर्ने।
3. Recursion बाट palindrome number check गर्ने? → उल्टो number recursion बाट बनाउने।
4. Recursion बाट factorial निकाल्ने? → Base case 1, अन्यथा n \* factorial(n-1)।
5. Iterative बाट factorial निकाल्ने? → Loop multiply गर्ने।
6. Recursion बाट Fibonacci निकाल्ने? → Base case 0 र 1, अन्यथा f(n-1)+f(n-2)।
7. Iterative बाट Fibonacci निकाल्ने? → Variables update गर्दै loop गर्ने।
8. Recursion बाट prime factors निकाल्ने? → Smallest divisor निकाल्दै recursion।
9. Power of two check गर्ने? → (n & (n-1)) == 0 र n > 0।
10. Recursion बाट Armstrong number check गर्ने? → Digits को power sum recursion बाट गर्ने।
11. Reverse number निकाल्ने? → Loop गरेर remainder \* 10 + digit गर्ने।
12. Recursion बाट Strong number check गर्ने? → Factorial sum recursion बाट गर्ने।
13. Perfect square check गर्ने? → sqrt(n) को integer square बराबर छ कि छैन हेर्ने।
14. Spy number के हो? → Digits को sum == product।
15. Happy number के हो? → Digits को square sum अन्ततः 1 हुने।
16. Duck number के हो? → Leading zero नभएको तर zero भएको number।
17. Buzz number के हो? → 7 बाट divisible वा अन्त्यमा 7 हुने।
18. Evil number के हो? → Binary मा 1 को count even हुने।
19. Magic number के हो? → Digits को sum recursive गर्दा 1 हुने।
20. Disarium number के हो? → प्रत्येक digit आफ्नो position को power को sum == number।

**Section 13: Arrays & Collections (Q141–Q160)**

1. Even index sum निकाल्ने? → i += 2 loop गर्ने।
2. Odd index sum निकाल्ने? → i=1 बाट सुरु गरेर i += 2।
3. Pair sum == target find गर्ने? → HashSet प्रयोग गर्ने।
4. Triplet sum == target find गर्ने? → Sorting + two pointers।
5. Median निकाल्ने? → Sort गरेर middle element।
6. Mode निकाल्ने? → Frequency highest element।
7. Average निकाल्ने? → Sum / length।
8. Palindrome elements मात्र print गर्ने? → Palindrome check गरेर।
9. Prime elements मात्र print गर्ने? → Prime check गरेर।
10. Merge sorted arrays बिना extra space गर्ने? → Gap method।
11. kth largest element निकाल्ने? → PriorityQueue वा quickselect।
12. Duplicate remove गर्ने? → LinkedHashSet प्रयोग गर्ने।
13. Maximum subarray sum निकाल्ने? → Kadane’s algorithm।
14. Minimum subarray sum निकाल्ने? → Kadane inverse logic।
15. Cyclic right rotation गर्ने? → Reverse method प्रयोग गर्ने।
16. Cyclic left rotation गर्ने? → Reverse method प्रयोग गर्ने।
17. Zero elements अन्त्यमा सार्ने तर order same राख्ने? → Stable shifting method।
18. Frequency अनुसार sort गर्ने? → Map + custom comparator।
19. First non-repeating element निकाल्ने? → Map count गरेर।
20. Leaders find गर्ने? → Rightmost देखि max track गर्ने।

**Section 14: File, Date & Miscellaneous (Q161–Q180)**

1. File मा append गर्ने? → FileWriter(file, true)।
2. CSV file पढ्ने? → BufferedReader र split(",")।
3. CSV file लेख्ने? → FileWriter प्रयोग गर्ने।
4. Text file मा specific word search गर्ने? → contains() check गरेर।
5. Text file मा word replace गर्ने? → Read → replaceAll() → write back।
6. Folder मा सबै files list गर्ने? → File.listFiles()।
7. केवल .txt files filter गर्ने? → FilenameFilter प्रयोग गर्ने।
8. Current time milliseconds मा पाउने? → System.currentTimeMillis()।
9. Program execution time measure गर्ने? → Start र end time को अन्तर।
10. Current working directory पाउने? → System.getProperty("user.dir")।
11. ZIP file बनाउने? → ZipOutputStream।
12. ZIP file extract गर्ने? → ZipInputStream।
13. PDF file पढ्ने? → Apache PDFBox / iText।
14. Excel file पढ्ने? → Apache POI।
15. Excel file लेख्ने? → Apache POI।
16. Image file size find गर्ने? → File.length()।
17. Image resize गर्ने? → BufferedImage प्रयोग गर्ने।
18. Image format change गर्ने? → ImageIO.write()।
19. File permissions check गर्ने? → canRead(), canWrite(), canExecute()।
20. Hidden file check गर्ने? → isHidden() method।

**Section 15: Advanced Problem Solving (Q181–Q200)**

1. String permutations generate गर्ने? → Recursion प्रयोग गर्ने।
2. Combinations generate गर्ने? → Backtracking।
3. Sudoku solver बनाउने? → Backtracking।
4. N-Queens solve गर्ने? → Backtracking।
5. Maze solver बनाउने? → DFS/BFS।
6. Graph cycle detection गर्ने? → DFS + visited array।
7. Bellman-Ford shortest path algorithm? → Edges relax repeatedly।
8. Prim’s algorithm MST निकाल्ने? → Priority queue प्रयोग गर्ने।
9. Kruskal’s algorithm MST निकाल्ने? → Disjoint set प्रयोग गर्ने।
10. Topological sort गर्ने? → DFS वा Kahn’s algorithm।
11. BST insert गर्ने? → Recursion।
12. BST delete गर्ने? → तीन case handle गर्ने।
13. Heap sort गर्ने? → Heapify + extract max।
14. Counting sort गर्ने? → Frequency array।
15. Radix sort गर्ने? → Digit-wise counting sort।
16. Shell sort गर्ने? → Gap reduce गर्दै insertion sort।
17. Bitwise AND, OR, XOR उदाहरण? → a & b, a | b, a ^ b।
18. XOR swap without temp? → a = a ^ b; b = a ^ b; a = a ^ b।
19. Count set bits in integer? → n & (n-1) method।
20. Fast exponentiation power निकाल्ने? → Divide & conquer method।

**Section 16: Extra Simple Core Java Q&A**

1. **Q:** Java को file extension के हो?
**A:** .java source code, .class compiled bytecode।
2. **Q:** Java को founder को नाम के हो?
**A:** James Gosling (Sun Microsystems)।
3. **Q:** Java पहिलो पटक कहिले release भयो?
**A:** 1995 मा।
4. **Q:** Java को current LTS version (2025 अनुसार) के हो?
**A:** Java 21 (Sept 2023 release)।
5. **Q:** Java को default package कुन हो?
**A:** java.lang।
6. **Q:** Package भनेको के हो?
**A:** Classes र interfaces को logically grouping।
7. **Q:** Import statement को syntax के हो?
**A:** import packageName.className; वा import packageName.\*;
8. **Q:** Access modifiers का प्रकार के के हुन्?
**A:** public, protected, default (package-private), private।
9. **Q:** Non-access modifiers का उदाहरण के हुन्?
**A:** static, final, abstract, synchronized, transient, volatile।
10. **Q:** Java मा default value of int के हो?
**A:** 0।
11. **Q:** Java मा default value of boolean के हो?
**A:** false।
12. **Q:** Static variable कसरी define गर्ने?
**A:** static dataType varName;
13. **Q:** Static method कसरी call गर्ने?
**A:** ClassName.methodName(); (object बिना)।
14. **Q:** Java मा main method को parameter के हो?
**A:** String array (String[] args)।
15. **Q:** System.out.println को मतलब के हो?
**A:** System class → out (PrintStream object) → println method।
16. **Q:** Java मा comments का प्रकार के के हुन्?
**A:** Single-line (//), Multi-line (/\* ... \*/), Documentation (/\*\* ... \*/)।
17. **Q:** Variable naming rules Java मा के हुन्?
**A:** Start with letter, $, or \_, spaces allowed छैन, case-sensitive।
18. **Q:** Java case-sensitive language हो?
**A:** हो।
19. **Q:** Local variable को default value हुन्छ?
**A:** हुँदैन, initialize गर्नुपर्छ।
20. **Q:** Instance variable भनेको के हो?
**A:** Class भित्र declare गरिएको तर method बाहिरको variable।

**Section 17: Keywords, Operators & Basics (Q221–Q250)**

1. **Q:** Java मा total reserved keywords कति छन्?
**A:** 50+ (Java version अनुसार फरक)।
2. **Q:** break keyword को काम के हो?
**A:** Loop वा switch terminate गर्ने।
3. **Q:** continue keyword को काम के हो?
**A:** Current iteration skip गरेर next iteration मा जाने।
4. **Q:** return keyword को काम के हो?
**A:** Method बाट value फर्काउने वा exit गर्ने।
5. **Q:** Arithmetic operators Java मा के हुन्?
**A:** +, -, \*, /, %।
6. **Q:** Relational operators Java मा के हुन्?
**A:** ==, !=, >, <, >=, <=।
7. **Q:** Logical operators Java मा के हुन्?
**A:** &&, ||, !।
8. **Q:** Assignment operators Java मा के हुन्?
**A:** =, +=, -=, \*=, /=, %=।
9. **Q:** Unary operators Java मा के हुन्?
**A:** ++, --, +, -।
10. **Q:** Ternary operator को syntax के हो?
**A:** condition ? valueIfTrue : valueIfFalse।
11. **Q:** Type casting का प्रकार Java मा के हुन्?
**A:** Implicit (widening) र Explicit (narrowing)।
12. **Q:** instanceof keyword को प्रयोग के हो?
**A:** Object को type check गर्ने।
13. **Q:** Bitwise operators Java मा के हुन्?
**A:** &, |, ^, ~, <<, >>, >>>।
14. **Q:** Shift operators Java मा के हुन्?
**A:** << (left shift), >> (right shift), >>> (unsigned right shift)।
15. **Q:** Pre-increment र post-increment बीच के फरक छ?
**A:** Pre (++i) increment पहिले; Post (i++) increment पछि।
16. **Q:** Wrapper classes किन प्रयोग हुन्छन्?
**A:** Primitive types लाई objects मा wrap गर्न।
17. **Q:** Autoboxing के हो?
**A:** Primitive → Wrapper automatic conversion।
18. **Q:** Unboxing के हो?
**A:** Wrapper → Primitive automatic conversion।
19. **Q:** Default constructor भनेको के हो?
**A:** बिना parameter भएको constructor, compiler द्वारा provide।
20. **Q:** Parameterized constructor भनेको के हो?
**A:** Parameters लिने constructor।
21. **Q:** Overloaded constructor भनेको के हो?
**A:** एउटै class मा multiple constructors, फरक parameters।
22. **Q:** Static block के हो?
**A:** Class load हुँदा एक पटक चल्ने block।
23. **Q:** Initialization block के हो?
**A:** Object create हुँदा चल्ने block (instance initializer)।
24. **Q:** Garbage collector manually कसरी run गर्ने?
**A:** System.gc();
25. **Q:** null literal को मतलब के हो?
**A:** Reference variable कुनै object point नगर्ने।
26. **Q:** this() call को प्रयोग के हो?
**A:** Same class को अर्को constructor call गर्न।
27. **Q:** super() call को प्रयोग के हो?
**A:** Parent class को constructor call गर्न।
28. **Q:** Method signature Java मा के हो?
**A:** Method name + parameter list।
29. **Q:** Variable arguments (varargs) कसरी define गर्ने?
**A:** methodName(int... numbers)।
30. **Q:** Marker annotation के हो?
**A:** बिना method भएको annotation (जस्तै: @Override, @Deprecated)।

**Section 18: Java Basics & Syntax (Q251–Q280)**

1. **Q:** Java मा comments लेख्ने ३ तरिका के हुन्?
 **A:** // (single-line), /\* ... \*/ (multi-line), /\*\* ... \*/ (documentation)।
2. **Q:** Java मा class को नामको convention के हो?
 **A:** PascalCase (जस्तै: MyClass, StudentInfo)।
3. **Q:** Java मा method को नामको convention के हो?
 **A:** camelCase (जस्तै: getName, calculateSum)।
4. **Q:** Java मा constant define गर्ने syntax के हो?
 **A:** final dataType CONSTANT\_NAME = value;।
5. **Q:** Java मा main method बिना program चल्छ?
 **A:** Normally हुँदैन, तर static blocks प्रयोग गरेर JVM चालु हुँदा चलाउन सकिन्छ (mainless approach deprecated छ)।
6. **Q:** Java मा single-line string output गर्न कुन method प्रयोग हुन्छ?
 **A:** System.out.print()।
7. **Q:** Java मा output पछि newline दिन कुन method प्रयोग हुन्छ?
 **A:** System.out.println()।
8. **Q:** Java मा formatted output कसरी गर्ने?
 **A:** System.out.printf() वा String.format() प्रयोग गरेर।
9. **Q:** Java मा variable declare मात्र गरेर initialize नगरेको भए local variables को default value हुन्छ?
 **A:** हुँदैन, compile-time error आउँछ।
10. **Q:** Java मा escape sequence \n के गर्छ?
 **A:** Newline दिन्छ।
11. **Q:** Escape sequence \t के गर्छ?
 **A:** Tab space दिन्छ।
12. **Q:** Java मा character literal कसरी लेख्ने?
 **A:** Single quotes भित्र (जस्तै: 'A')।
13. **Q:** Java मा boolean literal का value के हुन्?
 **A:** true र false।
14. **Q:** Java मा string literal कसरी define गर्ने?
 **A:** Double quotes भित्र (जस्तै: "Hello")।
15. **Q:** Java मा method बिना object कसरी चल्छ?
 **A:** Static block मा code राखेर।
16. **Q:** Java मा infinite loop कसरी लेख्ने?
 **A:** while(true) { ... } वा for(;;) { ... }।
17. **Q:** Java मा block scope के हो?
 **A:** {} भित्रको variable केवल त्यही block भित्र मात्र उपलब्ध हुन्छ।
18. **Q:** Java मा array को index सुरु कहाँबाट हुन्छ?
 **A:** 0 बाट।
19. **Q:** Java मा array को length कसरी पाउने?
 **A:** arrayName.length।
20. **Q:** Java मा arrayको last element को index के हुन्छ?
 **A:** length - 1।

**Section 19: Java Keywords & Small Concepts (Q281–Q310)**

1. **Q:** Java मा goto keyword हुन्छ?
 **A:** छैन, reserved keyword हो तर प्रयोग हुँदैन।
2. **Q:** Java मा const keyword प्रयोग हुन्छ?
 **A:** छैन, reserved keyword हो तर प्रयोग हुँदैन।
3. **Q:** Java मा native keyword के हो?
 **A:** JNI मार्फत native code (C/C++) call गर्न प्रयोग।
4. **Q:** Java मा strictfp keyword के हो?
 **A:** Floating-point calculation IEEE standard अनुसार गर्न बाध्य पार्ने।
5. **Q:** Java मा assert keyword के हो?
 **A:** Testing वा debugging का लागि condition validate गर्ने।
6. **Q:** Java मा synchronized method के हो?
 **A:** Multiple threads बाट एकैपटक access रोक्ने method।
7. **Q:** Java मा volatile variable के हो?
 **A:** Variable को value सबै threads मा तुरुन्त अपडेट हुने।
8. **Q:** Java मा transient keyword के हो?
 **A:** Serialization मा variable skip गर्न।
9. **Q:** Java मा default keyword कहिले प्रयोग हुन्छ?
 **A:** Interface default methods वा switch-case मा।
10. **Q:** Java मा enum को full form के हो?
 **A:** Enumeration।
11. **Q:** Java मा enum को constant naming convention के हो?
 **A:** All uppercase letters (जस्तै: MONDAY, TUESDAY)।
12. **Q:** Java मा switch statement मा string प्रयोग गर्न मिल्छ?
 **A:** हो, Java 7 देखि।
13. **Q:** Java मा switch-case मा duplicate case हुन्छ?
 **A:** हुँदैन, compile error।
14. **Q:** Java मा switch-case बिना break के हुन्छ?
 **A:** Fall-through behavior हुन्छ।
15. **Q:** Java मा continue statement label संग प्रयोग गर्न मिल्छ?
 **A:** हो, nested loops मा।
16. **Q:** Java मा method parameter final बनाउँदा के हुन्छ?
 **A:** Method भित्र parameter को value change गर्न सकिँदैन।
17. **Q:** Java मा super keyword constructor call मा कहिले हुन्छ?
 **A:** Child constructor को पहिलो line मा।
18. **Q:** Java मा multiple public classes एउटै file मा राख्न मिल्छ?
 **A:** हुँदैन, केवल एक मात्र public class allowed।
19. **Q:** Java मा main method private भए के हुन्छ?
 **A:** JVM call गर्न सक्दैन, compile हुन्छ तर run गर्दा error।
20. **Q:** Java मा static import के हो?
 **A:** Static members लाई class name बिना direct access गर्न दिने।

**Section 20: Java Memory & Miscellaneous (Q311–Q350)**

1. **Q:** Java मा heap memory मा के राखिन्छ?
 **A:** Objects र instance variables।
2. **Q:** Java मा stack memory मा के राखिन्छ?
 **A:** Local variables र method calls।
3. **Q:** JVM का main components के के हुन्?
 **A:** Class loader, runtime memory area, execution engine, garbage collector।
4. **Q:** Class loader का प्रकार के हुन्?
 **A:** Bootstrap, Extension, System (Application) class loader।
5. **Q:** Java मा class loading lazy हुन्छ?
 **A:** हो, जब first time reference हुन्छ।
6. **Q:** Java मा method overloading runtime मा हुन्छ?
 **A:** होइन, compile-time मा हुन्छ।
7. **Q:** Method overriding compile-time हुन्छ?
 **A:** होइन, runtime मा हुन्छ।
8. **Q:** Garbage collector explicitly call गर्दा guarantee हुन्छ?
 **A:** हुँदैन, केवल request मात्र पठाइन्छ।
9. **Q:** finalize() method automatic call हुन्छ?
 **A:** हो, GC द्वारा object हटाउनु अघि (deprecated)।
10. **Q:** JVM को default stack size change गर्न सकिन्छ?
 **A:** हो, JVM options -Xss प्रयोग गरेर।
11. **Q:** Heap size set कसरी गर्ने?
 **A:** JVM options -Xms (initial) र -Xmx (max)।
12. **Q:** Java मा OutOfMemoryError कहिले आउँछ?
 **A:** Heap मा space नभएर।
13. **Q:** StackOverflowError कहिले आउँछ?
 **A:** Recursive calls ले stack भरिँदा।
14. **Q:** Java मा String pool heap को कुन भागमा हुन्छ?
 **A:** Heap मा special area (interned strings)।
15. **Q:** Interned string के हो?
 **A:** String pool मा राखिएको unique literal।
16. **Q:** Java मा infinite recursion के गर्छ?
 **A:** StackOverflowError फ्याँक्छ।
17. **Q:** Immutable object के हो?
 **A:** जसको state change गर्न मिल्दैन।
18. **Q:** Mutable object के हो?
 **A:** जसको state change गर्न मिल्छ।
19. **Q:** Marker interface के हो?
 **A:** बिना methods भएको interface (जस्तै: Serializable)।
20. **Q:** Serializable interface को काम के हो?
 **A:** Object लाई byte stream मा convert गर्न।
21. **Q:** Externalizable interface के हो?
 **A:** Serialization custom control गर्न दिने।
22. **Q:** readObject() method कहाँ define हुन्छ?
 **A:** ObjectInputStream class मा।
23. **Q:** writeObject() method कहाँ define हुन्छ?
 **A:** ObjectOutputStream class मा।
24. **Q:** java.lang.Object class को parent class के हो?
 **A:** कुनै छैन, यो सबैको parent हो।
25. **Q:** Object class का केही methods नाम बताउनुहोस्।
 **A:** toString(), equals(), hashCode(), clone(), wait(), notify(), notifyAll()।
26. **Q:** equals() र == को main difference के हो?
 **A:** equals() value compare गर्छ; == reference compare गर्छ।
27. **Q:** hashCode() method को काम के हो?
 **A:** Object को hash value फर्काउने।
28. **Q:** toString() method को default behavior के हो?
 **A:** ClassName@HexHashCode string फर्काउने।
29. **Q:** Java मा wait() method synchronized block बाहिर call गर्न मिल्छ?
 **A:** हुँदैन, IllegalMonitorStateException आउँछ।
30. **Q:** notify() र notifyAll() बीच के फरक छ?
 **A:** notify() only one thread wake गर्छ; notifyAll() सबैलाई।
31. **Q:** Thread.sleep() र Object.wait() बीच फरक?
 **A:** sleep() lock release गर्दैन; wait() lock release गर्छ।
32. **Q:** Runnable interface मा कति abstract methods हुन्छ?
 **A:** 1 (run())।
33. **Q:** Callable interface मा कति abstract methods हुन्छ?
 **A:** 1 (call()), र value फर्काउँछ।
34. **Q:** Thread class र Runnable बीच फरक?
 **A:** Thread extend; Runnable implement।
35. **Q:** Daemon thread के हो?
 **A:** Background मा चल्ने low-priority thread।
36. **Q:** Daemon thread कसरी set गर्ने?
 **A:** thread.setDaemon(true) call गरेर start गर्नु अघि।
37. **Q:** Java मा join() method के गर्छ?
 **A:** एक thread अर्को thread समाप्त हुन पर्खन्छ।
38. **Q:** yield() method के गर्छ?
 **A:** Current thread CPU release गरेर ready state मा जान्छ।
39. **Q:** Thread priority को default value के हो?
 **A:** 5 (range 1–10)।
40. **Q:** Java मा TimerTask के हो?
 **A:** java.util.Timer द्वारा scheduled tasks execute गर्ने class।