1090 MCQs for 30 Sessions

📘 SESSION 1 – INTRODUCTION TO JAVA

🟢 EASY LEVEL MCQs (1–15)

1. Java was developed by?
A) Microsoft
B) Sun Microsystems
C) Oracle
D) IBM

✅ **Answer:** B

---

2. Java is a _______ language.
A) Low-level
B) Assembly
C) High-level
D) Machine-level

✅ **Answer:** C

---

3. Which of the following makes Java platform independent?
A) Compiler
B) JVM
C) JDK
D) JRE

✅ **Answer:** B

---

4. Java source files have extension:
A) `.class`
B) `.java`
C) `.exe`
D) `.jar`

✅ **Answer:** B

5. Which keyword is used to define a class?
A) struct
B) define
C) class
D) object

✅ **Answer:** C

6. Which method is the entry point of Java program?
A) start()
B) run()
C) main()
D) init()

✅ **Answer:** C

7. Which primitive data type is used to store true/false?
A) int
B) boolean
C) char
D) byte

✅ **Answer:** B

8. Size of `int` data type in Java?
A) 1 byte
B) 2 bytes
C) 4 bytes
D) 8 bytes

✅ **Answer:** C

9. Which of these is NOT a Java feature?
A) Object-oriented
B) Secure
C) Platform dependent
D) Portable

✅ **Answer:** C

10. Which command compiles Java code?
A) java
B) javac
C) jvm
D) jre

✅ **Answer:** B

---

11. JVM stands for?
A) Java Virtual Method
B) Java Variable Machine
C) Java Virtual Machine
D) Java Verified Machine

✅ **Answer:** C

---

12. Default value of boolean variable?
A) true
B) false
C) 0
D) null

✅ **Answer:** B

---

13. Which keyword is used to create object?
A) class
B) new
C) this
D) object

✅ **Answer:** B

---

14. Java is primarily used for:
A) Hardware programming
B) Web & Enterprise apps
C) OS kernel
D) Assembly coding

✅ **Answer:** B

---

15. Which data type stores single character?
A) String

B) char
C) byte
D) int

✅ **Answer:** B

🟡 MEDIUM LEVEL MCQs (16–30)

16. Which component loads class files into JVM?
A) JIT
B) Garbage Collector
C) Class Loader
D) Interpreter

✅ **Answer:** C

---

17. Java follows which memory model?
A) Stack only
B) Heap only
C) Stack + Heap
D) Cache

✅ **Answer:** C

---

18. Which of the following is NOT part of JVM?
A) Class Loader
B) Execution Engine
C) JDK
D) Runtime Data Area

✅ **Answer:** C

---

19. Size of `char` in Java is:
A) 1 byte
B) 2 bytes
C) 4 bytes
D) Platform dependent

✅ **Answer:** B

---

20. Which feature allows Java to run on multiple platforms?
A) Dynamic binding
B) Multithreading
C) Bytecode
D) Encapsulation

✅ **Answer:** C

---

21. Which data type has the largest range?
A) int
B) long
C) float
D) double

✅ **Answer:** B

---

22. JDK includes:
A) JVM only
B) JRE only
C) JRE + Development tools
D) Compiler only

✅ **Answer:** C

---

23. Which memory stores local variables?
A) Heap
B) Method area
C) Stack
D) Register

✅ **Answer:** C

---

24. Which keyword makes variable constant?
A) static
B) final
C) const
D) fixed

✅ **Answer:** B

---

25. Default value of int variable?
A) undefined

B) 0
C) null
D) garbage

✅ **Answer:** B

---

26. Java uses ______ typing.
A) Dynamic
B) Weak
C) Strong
D) Loose

✅ **Answer:** C

---

27. Which is true about Java bytecode?
A) Platform dependent
B) Interpreted by OS
C) Executed by JVM
D) Same as machine code

✅ **Answer:** C

---

28. Which primitive type is signed?
A) boolean
B) char
C) byte
D) none

✅ **Answer:** C

---

29. Which section stores bytecode?
A) Stack
B) Heap
C) Method Area
D) PC Register

✅ **Answer:** C

---

30. Java supports multiple inheritance using:
A) Classes
B) Objects

C) Interfaces
D) Constructors

✅ **Answer:** C

🔴 HARD LEVEL MCQs (31–45)

31. Which JVM component converts bytecode to machine code?
A) Interpreter
B) Class Loader
C) JIT Compiler
D) Garbage Collector

✅ **Answer:** C

---

32. Which primitive data type has no negative values?
A) int
B) byte
C) char
D) long

✅ **Answer:** C

---

33. Which memory is shared among all threads?
A) Stack
B) Heap
C) PC Register
D) Native stack

✅ **Answer:** B

---

34. Which is executed first when Java program runs?
A) main()
B) Static block
C) Constructor
D) Object creation

✅ **Answer:** B

---

35. Which statement about JVM is correct?
A) JVM is platform dependent

B) JVM is platform independent
C) JVM compiles code
D) JVM writes source code

✅ **Answer:** A

---

36. Which Java feature ensures security?
A) Pointers
B) Bytecode verification
C) Assembly code
D) Manual memory

✅ **Answer:** B

---

37. Why Java has no pointers?
A) To improve speed
B) To improve security
C) To save memory
D) To simplify syntax

✅ **Answer:** B

---

38. Which data type occupies least memory?
A) short
B) byte
C) char
D) boolean

✅ **Answer:** B

---

39. Which of these is NOT primitive?
A) int
B) float
C) String
D) char

✅ **Answer:** C

---

40. What happens if main() is not static?
A) Compile error
B) Runtime error

C) Program runs
D) JVM creates object automatically

✅ **Answer:** B

---

41. Java is called robust because of:
A) Pointers
B) Garbage collection
C) Assembly
D) Manual memory

✅ **Answer:** B

---

42. Which memory area stores constants?
A) Heap
B) Stack
C) Method Area
D) PC Register

✅ **Answer:** C

---

43. Which tool creates bytecode?
A) JVM
B) JRE
C) javac
D) java

✅ **Answer:** C

---

44. Java does not support operator overloading because:
A) Complexity
B) Security
C) Performance
D) Simplicity

✅ **Answer:** D

---

45. Which primitive type supports decimal values?
A) int
B) long
C) float
D) char

✅ **Answer:** C

🧩 CODE SNIPPET MCQs (10+)

Snippet 1

```
public class Test {
    public static void main(String[] args) {
        byte b = 127;
        b++;
        System.out.println(b);
    }
}
```

✅ **Output:** `-128` (overflow)

Snippet 2

```
int x;
System.out.println(x);
```

✅ **Answer:** Compile-time error (uninitialized variable)

Snippet 3

```
public class A {
    static {
        System.out.println("Hello");
    }
}
```

✅ **Answer:** Prints `Hello` when class loads

Snippet 4

```
System.out.println(10 + 20 + "Java");
```

✅ **Output:** `30Java`

Snippet 5

```
char c = 65;
System.out.println(c);
```

✅ **Output:** `A`

Snippet 6

```
float f = 10.5;
```

✅ **Answer:** Compile-time error (needs `10.5f`)

Snippet 7

```
boolean b = 1;
```

✅ **Answer:** Compile-time error

Snippet 8

```
long l = 1000;
int i = l;
```

✅ **Answer:** Compile-time error (narrowing)

Snippet 9

```
System.out.println('A' + 1);
```

✅ **Output:** `66`

Snippet 10

```
double d = 10 / 4;
System.out.println(d);
```

✅ **Output:** `2.0`

Snippet 11 (Bonus – HARD)

```
public class Test {
    public static void main(String[] args) {
        System.out.println(0.1 + 0.2 == 0.3);
    }
}
```

✅ **Output:** `false` (floating-point precision)

📘 SESSIONS 2 & 3 – BASIC PROGRAMMING CONCEPTS (JAVA)

🟢 EASY LEVEL MCQs (1–20)

1. Which of the following is NOT a Java token?
A) Keyword
B) Identifier
C) Operator
D) Pointer

✅ **Answer: D**

---

2. Which keyword is used to declare a variable constant?
A) static
B) final
C) const
D) fixed

✅ **Answer: B**

---

3. Which operator is used for logical AND?
A) &
B) &&
C) |
D) ||

✅ **Answer: B**

---

4. Which loop executes at least once?
A) for
B) while
C) do-while
D) foreach

✅ **Answer: C**

---

5. Index of first element in Java array is:
A) 0
B) 1
C) -1
D) Depends on size

✅ **Answer: A**

---

6. Which control statement is used to exit loop?
A) continue
B) exit
C) break
D) stop

✅ **Answer: C**

---

7. Which of these is a valid identifier?
A) 1num
B) num-1
C) num_1
D) num#

✅ **Answer: C**

---

8. Size of boolean array element is:
A) 1 bit

B) 1 byte
C) JVM dependent
D) 2 bytes

✅ Answer: C

---

9. Which operator increments value by 1?
A) +
B) ++
C) +=
D) --

✅ Answer: B

---

10. Which keyword skips current iteration?
A) break
B) continue
C) pass
D) exit

✅ Answer: B

---

11. Which array stores rows and columns?
A) 1-D
B) Jagged
C) Multi-dimensional
D) Linear

✅ Answer: C

---

12. Which keyword is used to return value from method?
A) break
B) exit
C) return
D) yield

✅ Answer: C

---

13. Java arrays are:
A) Static
B) Dynamic

C) Resizable
D) Fixed-size

✅ **Answer: D**

---

14. Which loop is best for known iterations?
A) while
B) do-while
C) for
D) infinite

✅ **Answer: C**

---

15. Which operator compares values?
A) =
B) ==
C) !=
D) Both B and C

✅ **Answer: D**

---

16. Which statement checks multiple conditions?
A) if
B) if-else
C) switch
D) loop

✅ **Answer: C**

---

17. Default value of int array elements?
A) garbage
B) null
C) 0
D) undefined

✅ **Answer: C**

---

18. Which operator has highest precedence?
A) +
B) *
C) ()
D) =

✅ **Answer: C**

---

19. Java supports which type of arrays?
A) Static
B) Dynamic
C) Both
D) None

✅ **Answer: A**

---

20. Which token represents constant value?
A) Identifier
B) Literal
C) Keyword
D) Operator

✅ **Answer: B**

🟡 MEDIUM LEVEL MCQs (21–40)

21. Which casting happens automatically?
A) Narrowing
B) Explicit
C) Widening
D) Manual

✅ **Answer: C**

---

22. Which is invalid assignment?
A) int a = 10;
B) double d = 10;
C) int x = 10.5;
D) float f = 10;

✅ **Answer: C**

---

23. Which operator is ternary?
A) ?:
B) ++
C) &&
D) %=

✅ **Answer: A**

---

24. Which control structure replaces multiple if-else?
A) for
B) while
C) switch
D) continue

✅ **Answer: C**

---

25. Which loop is entry-controlled?
A) do-while
B) while
C) infinite
D) none

✅ **Answer: B**

---

26. Which array declaration is correct?
A) int a[5];
B) int[] a = new int[5];
C) int a = new int[5];
D) array int a;

✅ **Answer: B**

---

27. Jagged array means:
A) Same columns
B) Different columns
C) 1-D array
D) Sorted array

✅ **Answer: B**

---

28. Which operator performs bitwise AND?
A) &&
B) &
C) |
D) ^

✅ **Answer: B**

29. Which loop is infinite?
A) for(;;)
B) while(false)
C) do-while(false)
D) foreach

✅ Answer: A

30. Method without return value uses:
A) null
B) int
C) void
D) empty

✅ Answer: C

31. Which is NOT allowed in Java compared to C++?
A) Multiple inheritance via class
B) Operator overloading
C) Pointers
D) All

✅ Answer: D

32. Which is mandatory in switch-case?
A) default
B) break
C) case
D) colon

✅ Answer: C

33. Java arrays are stored in:
A) Stack
B) Heap
C) Method area
D) Register

✅ Answer: B

34. Which operator checks object reference?
A) ==
B) equals
C) instanceof
D) !=

✅ **Answer: C**

---

35. Data type compatibility means:
A) Same data type
B) Convertible types
C) Same size
D) Same memory

✅ **Answer: B**

---

36. Which keyword is used to define method?
A) method
B) function
C) return
D) return type

✅ **Answer: D**

---

37. Which statement skips rest code and goes to next loop?
A) break
B) continue
C) exit
D) skip

✅ **Answer: B**

---

38. Which is valid switch type?
A) int
B) char
C) String
D) All

✅ **Answer: D**

---

39. Which array type has rows of different length?
A) Rectangular

B) Square
C) Jagged
D) Linear

✅ Answer: C

---

40. Which operator associates right to left?
A) +
B) =
C) *
D) &&

✅ Answer: B

🔴 HARD LEVEL MCQs (41–60)

41. What happens if break is missing in switch?
A) Error
B) Loop ends
C) Fall-through
D) Program stops

✅ Answer: C

---

42. Which is invalid array declaration?
A) int[][] a;
B) int a[][];
C) int[] a[];
D) int[5] a;

✅ Answer: D

---

43. Which operator has lowest precedence?
A) ++
B) *
C) +
D) =

✅ Answer: D

---

44. Which loop is best for pattern printing?
A) while
B) do-while
C) nested for
D) switch

✅ **Answer: C**

---

45. Java does NOT support:
A) Function overloading
B) Operator overloading
C) Method overloading
D) Constructor overloading

✅ **Answer: B**

---

46. Which array index causes exception?
A) arr[0]
B) arr[length-1]
C) arr[length]
D) arr[1]

✅ **Answer: C**

---

47. Which exception occurs for invalid array index?
A) NullPointerException
B) ArithmeticException
C) ArrayIndexOutOfBoundsException
D) ClassCastException

✅ **Answer: C**

---

48. Which operator is evaluated first?
A) =
B) ++
C) ?:
D) &&

✅ **Answer: B**

---

49. Which control statement has no condition?
A) for(;;)

B) while
C) do-while
D) switch

✅ Answer: A

---

50. Which is true about Java vs C++?
A) Java uses pointers
B) C++ has garbage collection
C) Java has automatic memory management
D) Java supports multiple inheritance via class

✅ Answer: C

---

51. Which loop is fastest generally?
A) while
B) for
C) do-while
D) depends

✅ Answer: D

---

52. Which operator is short-circuit?
A) &
B) &&
C) |
D) ^

✅ Answer: B

---

53. Which array initialization is correct?
A) int a[] = {1,2,3};
B) int a = {1,2};
C) a[] = new int[5];
D) array a = new int[5];

✅ Answer: A

---

54. Which is NOT a Java token category?
A) Keyword
B) Operator

C) Comment
D) Literal

✅ **Answer: C**

---

55. Which control flow is selection-based?
A) for
B) while
C) if-else
D) continue

✅ **Answer: C**

---

56. Java does not allow:
A) Nested loops
B) Nested if
C) goto
D) switch

✅ **Answer: C**

---

57. Which keyword prevents inheritance?
A) private
B) final
C) static
D) protected

✅ **Answer: B**

---

58. Which array size is decided at runtime?
A) Static
B) Dynamic
C) Fixed
D) Constant

✅ **Answer: B**

---

59. Which loop condition is checked last?
A) for
B) while
C) do-while
D) foreach

✅ **Answer: C**

---

60. Java arrays are:
A) Value types
B) Reference types
C) Primitive
D) None

✅ **Answer: B**

🧩 CODE SNIPPET MCQs (20+)

Snippet 1

```
int a = 5;
System.out.println(a++ + ++a);
```

✅ **Output:** `12`

Snippet 2

```
for(int i=1;i<=3;i++)
    System.out.print("*");
```

✅ **Output:** `***`

Snippet 3

```
int[] a = new int[3];
System.out.println(a[1]);
```

✅ **Output:** `0`

Snippet 4

```
int x = 10;
if(x = 5)
```

```
    System.out.println("Hi");
```

✅ **Answer: Compile-time error**

Snippet 5

```
int i = 1;
while(i++ <= 3)
    System.out.print(i);
```

✅ **Output:** `234`

Snippet 6

```
int[][] a = {{1,2},{3,4,5}};
System.out.println(a[1].length);
```

✅ **Output:** `3`

Snippet 7

```
int x = 5;
System.out.println(x > 3 ? x : 3);
```

✅ **Output:** `5`

Snippet 8

```
for(int i=1;i<=3;i++) {
    for(int j=1;j<=i;j++)
        System.out.print("*");
    System.out.println();
}
```

✅ **Output:**

```
*
**
```

```
***
```

Snippet 9

```
int a = 10;
int b = 20;
a = a + b - (b = a);
System.out.println(a + " " + b);
```

✅ **Output:** `20 10`

Snippet 10

```
switch(2) {
    case 1: System.out.print("A");
    case 2: System.out.print("B");
    case 3: System.out.print("C");
}
```

✅ **Output:** `BC`

Snippet 11

```
int[] a = {1,2,3};
a[3] = 10;
```

✅ **Answer: Runtime Exception**

Snippet 12

```
int i = 0;
do {
    System.out.print(i);
    i++;
} while(i < 0);
```

✅ **Output:** `0`

Snippet 13

```
int x = 5;
System.out.println(++x * x++);
```

✅ **Output:** 36

Snippet 14

```
int[] a = new int[-5];
```

✅ **Answer: Runtime Exception (NegativeArraySizeException)**

Snippet 15

```
boolean b = true;
if(b = false)
    System.out.println("Yes");
else
    System.out.println("No");
```

✅ **Output:** No

Snippet 16

```
int i = 1;
for(; i <= 3; i++);
System.out.print(i);
```

✅ **Output:** 4

Snippet 17

```
char c = 'A';
System.out.println(c + 1);
```

✅ **Output:** `66`

Snippet 18

```
int x = 10;
System.out.println(x << 1);
```

✅ **Output:** `20`

Snippet 19

```
int[] a = null;
System.out.println(a.length);
```

✅ **Answer: NullPointerException**

Snippet 20

```
int i = 10;
while(i-- > 8)
    System.out.print(i);
```

✅ **Output:** `98`

📘 SESSION 4 – OBJECT ORIENTED PROGRAMMING CONCEPTS (JAVA)

🟢 EASY LEVEL MCQs (1–20)

1. OOP stands for:
A) Object Oriented Programming
B) Object Operated Programming
C) Open Object Programming
D) Optional Object Programming

✅ **Answer:** A

2. An object is:
A) Blueprint
B) Instance of class
C) Method
D) Variable

✅ **Answer:** B

3. A class is:
A) Object
B) Instance
C) Blueprint
D) Method

✅ **Answer:** C

4. Which keyword is used to create an object?
A) class
B) new
C) this
D) static

✅ **Answer:** B

5. Encapsulation means:
A) Code reuse
B) Hiding implementation
C) Multiple inheritance
D) Code duplication

✅ **Answer:** B

6. Which access modifier is most restrictive?
A) public
B) protected
C) default
D) private

✅ **Answer:** D

7. Which keyword is used to define a class?
A) struct
B) object
C) class
D) define

✅ **Answer:** C

---

8. Which of the following is NOT an OOP principle?
A) Encapsulation
B) Abstraction
C) Compilation
D) Inheritance

✅ **Answer:** C

---

9. Data members of class are usually declared as:
A) public
B) protected
C) private
D) static

✅ **Answer:** C

---

10. Which is used to access private data?
A) Constructor
B) Getter/Setter
C) Object
D) Method call

✅ **Answer:** B

---

11. Which keyword hides data?
A) final
B) static
C) private
D) this

✅ **Answer:** C

---

12. Which allows partial implementation?
A) Class

B) Object
C) Abstract class
D) Package

✅ **Answer:** C

---

13. Can we create object of abstract class?
A) Yes
B) No
C) Sometimes
D) Only in same package

✅ **Answer:** B

---

14. Which keyword is used to declare abstract method?
A) virtual
B) abstract
C) static
D) default

✅ **Answer:** B

---

15. Interface methods are by default:
A) private
B) protected
C) public abstract
D) static

✅ **Answer:** C

---

16. Abstraction focuses on:
A) How things work
B) What things do
C) Memory layout
D) Speed

✅ **Answer:** B

---

17. Which hides internal details from user?
A) Inheritance
B) Abstraction

C) Polymorphism
D) Compilation

✅ **Answer:** B

---

18. Which keyword prevents data modification?
A) static
B) final
C) private
D) protected

✅ **Answer:** B

---

19. Which is correct naming convention for class?
A) myclass
B) my_class
C) MyClass
D) MYCLASS

✅ **Answer:** C

---

20. Which principle improves security?
A) Encapsulation
B) Inheritance
C) Polymorphism
D) Overloading

✅ **Answer:** A

🟡 MEDIUM LEVEL MCQs (21–40)

21. Encapsulation is achieved using:
A) Methods only
B) Variables only
C) Access modifiers
D) Constructors

✅ **Answer:** C

---

22. Which member can be accessed outside class?
A) private

B) default
C) protected
D) public

✅ **Answer:** D

---

23. Which is mandatory in abstract class?
A) Abstract method
B) Concrete method
C) Object creation
D) Constructor

✅ **Answer:** D

---

24. Abstract class can contain:
A) Only abstract methods
B) Only concrete methods
C) Both
D) None

✅ **Answer:** C

---

25. Interface can contain variables that are:
A) private
B) protected
C) final static
D) non-static

✅ **Answer:** C

---

26. Which promotes loose coupling?
A) Encapsulation
B) Abstraction
C) Classes
D) Objects

✅ **Answer:** B

---

27. Which keyword is used to access current object?
A) this
B) super

C) new
D) self

✅ **Answer:** A

---

28. Can abstract class have constructor?
A) No
B) Yes
C) Only static
D) Only private

✅ **Answer:** B

---

29. Which is NOT allowed in interface?
A) Abstract methods
B) Static final variables
C) Instance variables
D) Default methods

✅ **Answer:** C

---

30. Which improves maintainability?
A) Hard coding
B) Encapsulation
C) Global variables
D) Public data

✅ **Answer:** B

---

31. Which hides complexity from user?
A) Encapsulation
B) Abstraction
C) Compilation
D) JVM

✅ **Answer:** B

---

32. Abstract method must be implemented by:
A) Same class
B) Any class
C) Concrete subclass
D) Interface

✅ **Answer:** C

---

33. Which is a real-world example of abstraction?
A) Engine design
B) Car steering
C) Car driving
D) Circuit diagram

✅ **Answer:** C

---

34. Can interface have constructor?
A) Yes
B) No
C) Only default
D) Only private

✅ **Answer:** B

---

35. Which is better for hiding implementation?
A) Class
B) Object
C) Interface
D) Package

✅ **Answer:** C

---

36. Which OOP principle binds data & methods?
A) Abstraction
B) Encapsulation
C) Inheritance
D) Polymorphism

✅ **Answer:** B

---

37. Which is valid abstract method?
A) void run(){}
B) abstract void run();
C) abstract void run(){}
D) void abstract run();

✅ **Answer:** B

38. Interface methods before Java 8 were:
A) private
B) static
C) abstract
D) default

✅ **Answer:** C

39. Which access modifier allows package-level access?
A) public
B) private
C) default
D) protected

✅ **Answer:** C

40. Which allows multiple abstraction?
A) Class
B) Object
C) Interface
D) Constructor

✅ **Answer:** C

🔴 HARD LEVEL MCQs (41–60)

41. Which OOP principle reduces dependency?
A) Encapsulation
B) Abstraction
C) Polymorphism
D) Inheritance

✅ **Answer:** B

42. Why fields are private in encapsulation?
A) Performance
B) Security
C) Readability
D) Compilation

✅ **Answer:** B

43. Which is invalid in abstract class?
A) Constructor
B) Static method
C) Final method
D) Object creation

✅ **Answer:** D

44. Which is true about interface variables?
A) Mutable
B) Instance based
C) Constants
D) Protected

✅ **Answer:** C

45. Abstract class supports:
A) Multiple inheritance
B) Partial abstraction
C) Full abstraction only
D) No abstraction

✅ **Answer:** B

46. Which keyword forces implementation?
A) final
B) abstract
C) private
D) static

✅ **Answer:** B

47. Which provides 100% abstraction?
A) Class
B) Abstract class
C) Interface
D) Object

✅ **Answer:** C

48. Can abstract method be static?
A) Yes
B) No
C) Sometimes
D) Only protected

✅ **Answer:** B

---

49. Why abstraction is important?
A) Faster code
B) Less memory
C) Reduced complexity
D) More syntax

✅ **Answer:** C

---

50. Which violates encapsulation?
A) Private fields
B) Public getters
C) Public fields
D) Access methods

✅ **Answer:** C

---

51. Which class member belongs to class not object?
A) Instance variable
B) Local variable
C) Static variable
D) Parameter

✅ **Answer:** C

---

52. Which hides data at compile-time?
A) Encapsulation
B) Abstraction
C) JVM
D) Constructor

✅ **Answer:** A

---

53. Which is correct about abstraction?
A) Implementation hiding

B) Data hiding
C) Object hiding
D) Memory hiding

✅ **Answer:** A

---

54. Which cannot be abstract?
A) Class
B) Method
C) Constructor
D) Interface

✅ **Answer:** C

---

55. Which improves code readability?
A) Abstraction
B) Global variables
C) Public fields
D) Hard coding

✅ **Answer:** A

---

56. Interface supports:
A) State
B) Implementation
C) Contract
D) Object creation

✅ **Answer:** C

---

57. Which method cannot be abstract?
A) Static
B) Public
C) Protected
D) Default

✅ **Answer:** A

---

58. Which OOP principle hides internal working?
A) Encapsulation
B) Abstraction

C) Both A & B
D) None

✅ **Answer:** C

---

59. Which keyword stops overriding?
A) abstract
B) private
C) final
D) static

✅ **Answer:** C

---

60. Best example of encapsulation:
A) Class with public fields
B) Class with private fields & public methods
C) Interface
D) Object

✅ **Answer:** B

🧩 CODE SNIPPET MCQs (20)

Snippet 1

```
class A {
    private int x = 10;
}
System.out.println(new A().x);
```

✅ **Answer:** Compile-time error

Snippet 2

```
class Test {
    int x;
}
System.out.println(new Test().x);
```

✅ **Output:** `0`

Snippet 3

```
class A {
    private int x;
    public int getX() { return x; }
}
```

✅ **Concept:** Encapsulation

Snippet 4

```
abstract class A {
    abstract void show();
}
```

✅ **Valid abstraction**

Snippet 5

```
abstract class A {
    A() { System.out.println("Constructor"); }
}
```

✅ **Valid (abstract class can have constructor)**

Snippet 6

```
interface I {
    int x = 10;
}
I.x = 20;
```

✅ **Answer:** Compile-time error

Snippet 7

```
interface I {
    void show();
}
```

✅ **Method is public abstract by default**

Snippet 8

```
class A {
    private int x;
}
class B {
    void show() {
        A a = new A();
        // a.x = 5;
    }
}
```

✅ **Encapsulation enforced**

Snippet 9

```
abstract class A {
    void show() {}
}
```

✅ **Valid (non-abstract method allowed)**

Snippet 10

```
abstract class A {
    abstract void show();
}
new A();
```

✅ **Compile-time error**

Snippet 11

```
interface I {
    static void show() {}
}
```

✅ **Valid (Java 8+)**

Snippet 12

```
class A {
    final int x = 10;
    void set() { /* x = 20; */ }
}
```

✅ **Compile-time error if uncommented**

Snippet 13

```
abstract class A {
    abstract void show();
}
class B extends A {}
```

✅ **Compile-time error (show not implemented)**

Snippet 14

```
class A {
    private int x;
    public void setX(int x) {
        this.x = x;
    }
}
```

✅ **Correct encapsulation**

Snippet 15

```
interface I {
    default void show() {
        System.out.println("Hello");
    }
}
```

✅ **Valid abstraction**

Snippet 16

```
abstract class A {
    abstract static void show();
}
```

✅ **Compile-time error**

Snippet 17

```
class A {
    int x;
    A(int x) {
        this.x = x;
    }
}
```

✅ **Constructor using `this`**

Snippet 18

```
interface I {
    private void show() {}
}
```

✅ **Valid (Java 9+)**

Snippet 19

```
abstract class A {
    final void show() {}
}
```

✅ **Valid**

Snippet 20

```
class A {
    public int x;
}
```

✅ **Violates encapsulation**

📘 SESSION 5 – STATIC & REFERENCE CONCEPTS (JAVA)

🟢 EASY LEVEL MCQs (1–15)

1. Static variable belongs to:
A) Object
B) Method
C) Class
D) Constructor

✅ **Answer:** C

---

2. Static variables are created:
A) Per object
B) Per method call
C) Once per class
D) Every time constructor runs

✅ **Answer:** C

---

3. Static members are accessed using:
A) Object reference
B) Class name
C) Constructor
D) this keyword

✅ **Answer:** B

---

4. Which keyword is used to declare static member?
A) final
B) this
C) static
D) const

✅ **Answer:** C

---

5. Primitive data types store:
A) Objects
B) Memory address
C) Actual values
D) Methods

✅ **Answer:** C

---

6. Reference data types store:
A) Values
B) Object data
C) Memory address
D) Stack memory

✅ **Answer:** C

---

7. Which is a reference data type?
A) int
B) float
C) char
D) String

✅ **Answer:** D

---

8. Static method can access:
A) Only instance variables
B) Only static variables
C) Both static & instance
D) Local variables only

✅ **Answer:** B

9. How many copies of static variable exist?
A) One per object
B) One per method
C) One per class
D) One per thread

✅ **Answer:** C

10. Reference variable refers to:
A) Class
B) Method
C) Object
D) Static block

✅ **Answer:** C

11. Which memory stores objects?
A) Stack
B) Heap
C) Method area
D) Register

✅ **Answer:** B

12. Default value of reference variable is:
A) 0
B) garbage
C) null
D) undefined

✅ **Answer:** C

13. Static method is called using:
A) Object
B) Reference
C) Class name
D) this

✅ **Answer:** C

14. Primitive variables store data in:
A) Heap
B) Stack
C) Method area
D) Cache

✅ **Answer:** B

---

15. Which keyword creates reference?
A) this
B) static
C) new
D) final

✅ **Answer:** C

🟡 MEDIUM LEVEL MCQs (16–30)

16. Static methods cannot use:
A) Static variables
B) Local variables
C) Instance variables
D) Parameters

✅ **Answer:** C

---

17. Reference variable stores:
A) Object
B) Object value
C) Address of object
D) Class data

✅ **Answer:** C

---

18. Which is correct about static block?
A) Executes after main
B) Executes before main
C) Executes per object
D) Executes per method

✅ **Answer:** B

---

19. Which access is preferred for static members?
A) Object name
B) Class name
C) Reference variable
D) this

✅ **Answer:** B

---

20. Which is NOT a reference type?
A) Array
B) Class
C) Interface
D) int

✅ **Answer:** D

---

21. Reference variables are stored in:
A) Heap
B) Stack
C) Method area
D) Register

✅ **Answer:** B

---

22. Static variable is initialized:
A) When object is created
B) When method is called
C) When class is loaded
D) When constructor runs

✅ **Answer:** C

---

23. Which can be static?
A) Local variable
B) Instance variable
C) Method
D) Parameter

✅ **Answer:** C

---

24. Which statement is true?
A) Static variable depends on object

B) Reference variable depends on class
C) Static variable shared by all objects
D) Reference variable shared by class

✅ **Answer:** C

---

25. Which data type comparison checks address?
A) Primitive
B) Reference
C) Boolean
D) char

✅ **Answer:** B

---

26. Which operator compares reference addresses?
A) equals()
B) ==
C) !=
D) instanceof

✅ **Answer:** B

---

27. Static methods are bound at:
A) Runtime
B) Compile time
C) Object creation
D) JVM shutdown

✅ **Answer:** B

---

28. Which is better for constants?
A) Reference variable
B) Instance variable
C) Static final variable
D) Local variable

✅ **Answer:** C

---

29. Primitive data types are passed by:
A) Reference
B) Value

C) Object
D) Address

✅ **Answer:** B

---

30. Reference types are passed by:
A) Value
B) Address
C) Reference value
D) Copy

✅ **Answer:** C

🔴 HARD LEVEL MCQs (31–45)

31. Which is loaded first?
A) Object
B) Constructor
C) Static block
D) main method

✅ **Answer:** C

---

32. Which causes NullPointerException?
A) Uninitialized static variable
B) Uninitialized local variable
C) Accessing null reference
D) Accessing static variable

✅ **Answer:** C

---

33. Which memory area stores static variables?
A) Heap
B) Stack
C) Method area
D) Register

✅ **Answer:** C

---

34. Which statement is false?
A) Static variables are shared

B) Reference variables point to objects
C) Primitive variables store address
D) Static methods belong to class

✅ **Answer:** C

---

35. Which can change object state?
A) Static variable
B) Reference variable
C) Final variable
D) Constant

✅ **Answer:** B

---

36. Which is true about reference equality?
A) Checks value
B) Checks content
C) Checks address
D) Checks data type

✅ **Answer:** C

---

37. Static methods support:
A) Overriding
B) Dynamic binding
C) Method hiding
D) Runtime polymorphism

✅ **Answer:** C

---

38. Which is shared across objects?
A) Reference variable
B) Instance variable
C) Static variable
D) Local variable

✅ **Answer:** C

---

39. Which cannot be static?
A) Nested class
B) Constructor

C) Method
D) Variable

✅ **Answer:** B

---

40. Which is safer for shared data?
A) Instance variable
B) Local variable
C) Static variable
D) Reference variable

✅ **Answer:** C

---

41. What happens when reference is set to null?
A) Object destroyed
B) Memory freed immediately
C) Eligible for GC
D) JVM crash

✅ **Answer:** C

---

42. Which improves memory efficiency?
A) Instance variables
B) Static variables
C) Reference variables
D) Local variables

✅ **Answer:** B

---

43. Which best describes reference variable?
A) Holds value
B) Holds object
C) Holds address of object
D) Holds class

✅ **Answer:** C

---

44. Which comparison is content-based?
A) ==
B) !=
C) equals()
D) instanceof

✅ **Answer:** C

---

45. Which difference is correct?
A) Static variable → per object
B) Reference variable → per class
C) Static variable → shared
D) Reference variable → shared

✅ **Answer:** C

🧩 CODE SNIPPET MCQs (15)

Snippet 1

```
class A {
    static int x = 10;
}
class Test {
    public static void main(String[] args) {
        System.out.println(A.x);
    }
}
```

✅ **Output:** `10`

Snippet 2

```
class A {
    static int x = 10;
}
class B {
    public static void main(String[] args) {
        A a1 = new A();
        A a2 = new A();
        a1.x = 20;
        System.out.println(a2.x);
    }
}
```

✅ **Output:** `20`

Snippet 3

```
class Test {
    static int x;
    public static void main(String[] args) {
        System.out.println(x);
    }
}
```

✅ **Output:** `0`

Snippet 4

```
class A {
    int x = 10;
}
public class Test {
    public static void main(String[] args) {
        A a1 = new A();
        A a2 = a1;
        a2.x = 20;
        System.out.println(a1.x);
    }
}
```

✅ **Output:** `20`

Snippet 5

```
class Test {
    static void show() {
        // System.out.println(x);
    }
    int x = 10;
}
```

✅ **Answer:** Compile-time error if uncommented

Snippet 6

```
class Test {
    static {
        System.out.println("Static Block");
    }
    public static void main(String[] args) {}
}
```

✅ **Output:** `Static Block`

Snippet 7

```
class Test {
    public static void main(String[] args) {
        int x = 10;
        int y = x;
        y = 20;
        System.out.println(x);
    }
}
```

✅ **Output:** `10`

Snippet 8

```
class A {
    int x = 10;
}
class Test {
    public static void main(String[] args) {
        A a = null;
        System.out.println(a.x);
    }
}
```

✅ **Answer:** NullPointerException

Snippet 9

```
class Test {
    static int x = 10;
    static void show() {
        System.out.println(x);
    }
}
```

✅ **Valid static access**

Snippet 10

```
class Test {
    public static void main(String[] args) {
        String s = null;
        System.out.println(s.length());
    }
}
```

✅ **Answer:** NullPointerException

Snippet 11

```
class A {
    static int x = 10;
}
class Test {
    public static void main(String[] args) {
        A.x++;
        System.out.println(A.x);
    }
}
```

✅ **Output:** `11`

Snippet 12

```
class Test {
    static int x = 10;
```

```java
    static int y = x + 5;
}
```

✅ **Valid**

Snippet 13

```java
class Test {
    static void show() {
        System.out.println("Hello");
    }
    public static void main(String[] args) {
        show();
    }
}
```

✅ **Output:** `Hello`

Snippet 14

```java
class Test {
    int x = 10;
    static int y = 20;
}
```

✅ **x → instance, y → static**

Snippet 15

```java
class Test {
    public static void main(String[] args) {
        Test t = new Test();
        System.out.println(t);
    }
}
```

✅ **Output:** Reference value (hashcode format)

📘 SESSION 6 – CONSTRUCTORS, REFERENCES & MEMORY (JAVA)

🟢 EASY LEVEL MCQs (1–15)

1. Constructor name must be same as:
A) Method
B) Variable
C) Class
D) Object

✅ **Answer:** C

---

2. Constructors are used to:
A) Destroy objects
B) Initialize objects
C) Call methods
D) Allocate stack memory

✅ **Answer:** B

---

3. Constructors do not have:
A) Parameters
B) Access modifier
C) Return type
D) Body

✅ **Answer:** C

---

4. Default constructor is provided by:
A) Programmer
B) JVM
C) Compiler
D) JRE

✅ **Answer:** C

---

5. Object is created using:
A) this
B) super
C) new
D) static

✅ **Answer:** C

---

6. Reference variable stores:
A) Object
B) Value
C) Address of object
D) Class

✅ **Answer:** C

---

7. Objects are stored in:
A) Stack
B) Heap
C) Method area
D) Register

✅ **Answer:** B

---

8. Local variables are stored in:
A) Heap
B) Stack
C) Method area
D) Cache

✅ **Answer:** B

---

9. Java supports:
A) Pass by reference
B) Pass by value
C) Both
D) None

✅ **Answer:** B

---

10. Instance variables get default values from:
A) Constructor

B) Programmer
C) JVM
D) Compiler

✅ **Answer:** C

---

11. Reference variable default value is:
A) 0
B) garbage
C) null
D) undefined

✅ **Answer:** C

---

12. Constructor is invoked when:
A) Class loads
B) Method is called
C) Object is created
D) Program ends

✅ **Answer:** C

---

13. Stack memory is used for:
A) Objects
B) Class variables
C) Method calls
D) Static blocks

✅ **Answer:** C

---

14. Heap memory is shared among:
A) Threads
B) Methods
C) Objects
D) Local variables

✅ **Answer:** A

---

15. Which keyword refers to current object?
A) this
B) super

C) new
D) static

✅ **Answer:** A

🟡 MEDIUM LEVEL MCQs (16–30)

16. Parameterized constructor is used to:
A) Allocate memory
B) Initialize object with values
C) Call methods
D) Load class

✅ **Answer:** B

---

17. Which memory stores reference variables?
A) Heap
B) Stack
C) Method area
D) Register

✅ **Answer:** B

---

18. What happens when reference is reassigned?
A) Old object destroyed
B) JVM crash
C) Reference points to new object
D) Memory freed immediately

✅ **Answer:** C

---

19. Garbage collection occurs when:
A) Reference becomes null
B) Object has no reference
C) Method ends
D) Constructor ends

✅ **Answer:** B

---

20. Passing object to method actually passes:
A) Object

B) Address copy
C) Reference value
D) Heap location

✅ **Answer:** C

---

21. Which variable lifetime is method-bound?
A) Instance
B) Static
C) Local
D) Reference

✅ **Answer:** C

---

22. Constructor overloading means:
A) Same constructor
B) Different names
C) Same name, different parameters
D) Different classes

✅ **Answer:** C

---

23. Which is stored in stack during method call?
A) Object
B) Reference
C) Static variable
D) Class

✅ **Answer:** B

---

24. Which statement is true?
A) Objects live in stack
B) Reference variables live in heap
C) Objects live in heap
D) Methods live in heap

✅ **Answer:** C

---

25. If reference is passed to method and modified, what happens?
A) Object unchanged
B) Reference changes only inside method

C) Object state changes
D) JVM error

✅ **Answer:** C

---

26. Which is initialized first?
A) Constructor
B) Instance variables
C) Method call
D) Reference

✅ **Answer:** B

---

27. Constructor chaining is achieved using:
A) super
B) this
C) new
D) static

✅ **Answer:** B

---

28. Which is NOT allowed in constructor?
A) Access modifier
B) Parameters
C) Return type
D) this keyword

✅ **Answer:** C

---

29. Which causes memory leak?
A) Null reference
B) Unused object with reference
C) Garbage collection
D) Stack variable

✅ **Answer:** B

---

30. Heap memory allocation is:
A) Static
B) Compile time
C) Runtime
D) Load time

✅ **Answer:** C

🔴 HARD LEVEL MCQs (31–45)

31. Which is passed to method in Java?
A) Object
B) Address
C) Copy of reference
D) Heap pointer

✅ **Answer:** C

---

32. Which statement explains Java pass by value?
A) Object copied
B) Reference copied
C) Memory shared
D) Address passed directly

✅ **Answer:** B

---

33. When does object become eligible for GC?
A) Reference reassigned
B) Method ends
C) Constructor ends
D) Class unloads

✅ **Answer:** A

---

34. Which memory handles recursion?
A) Heap
B) Stack
C) Method area
D) Cache

✅ **Answer:** B

---

35. Which variable is destroyed first?
A) Instance
B) Static
C) Local
D) Object

✅ **Answer:** C

---

36. Which best describes stack memory?
A) Shared
B) Thread-safe
C) Slower
D) Stores objects

✅ **Answer:** B

---

37. Which is faster to access?
A) Heap
B) Stack
C) Method area
D) Disk

✅ **Answer:** B

---

38. Which is true about constructor call order?
A) Child → Parent
B) Parent → Child
C) Random
D) Parallel

✅ **Answer:** B

---

39. Which reference reassignment causes GC eligibility?
A) Two references → one object
B) One reference → two objects
C) Reference → null
D) All

✅ **Answer:** D

---

40. Which is false?
A) Objects stored in heap
B) References stored in stack
C) Static stored in stack
D) Methods stored in stack

✅ **Answer:** C

41. Which memory overflows in deep recursion?
A) Heap
B) Stack
C) Method area
D) Cache

✅ **Answer:** B

42. Which constructor is executed first?
A) Parameterized
B) Default
C) No-arg
D) Based on object

✅ **Answer:** D

43. Which statement is correct?
A) Java has true pass by reference
B) Java modifies original reference
C) Java passes copy of reference
D) Java copies object

✅ **Answer:** C

44. Which causes StackOverflowError?
A) Infinite object creation
B) Infinite recursion
C) Memory leak
D) GC failure

✅ **Answer:** B

45. Which memory is cleaned automatically?
A) Stack
B) Heap
C) Method area
D) Both A & B

✅ **Answer:** D

🧩 CODE SNIPPET MCQs (15)

Snippet 1

```
class Test {
    int x;
}
public class Main {
    public static void main(String[] args) {
        Test t = new Test();
        System.out.println(t.x);
    }
}
```

✅ **Output:** `0`

Snippet 2

```
class Test {
    Test() {
        System.out.println("Constructor");
    }
}
public class Main {
    public static void main(String[] args) {
        new Test();
    }
}
```

✅ **Output:** `Constructor`

Snippet 3

```
class A {
    int x = 10;
}
class Main {
    static void change(A a) {
        a.x = 20;
    }
    public static void main(String[] args) {
        A obj = new A();
        change(obj);
        System.out.println(obj.x);
    }
}
```

✅ **Output:** `20`

Snippet 4

```
class A {
    int x = 10;
}
public class Main {
    public static void main(String[] args) {
        A a1 = new A();
        A a2 = a1;
        a2.x = 30;
        System.out.println(a1.x);
    }
}
```

✅ **Output:** `30`

Snippet 5

```
class A {
    A(int x) {}
}
new A();
```

✅ **Answer:** Compile-time error

Snippet 6

```
class A {
    int x;
}
```

```
class B {
    A a = new A();
}
```

✅ **Valid reference initialization**

Snippet 7

```
class Test {
    static int x;
    int y;
}
public class Main {
    public static void main(String[] args) {
        Test t = new Test();
        System.out.println(t.x + " " + t.y);
    }
}
```

✅ **Output:** `0 0`

Snippet 8

```
class A {
    A() {
        this(10);
    }
    A(int x) {
        System.out.println(x);
    }
}
```

✅ **Output:** `10`

Snippet 9

```
class Test {
    public static void main(String[] args) {
        Test t = null;
        t.toString();
    }
}
```

✅ **Answer:** NullPointerException

Snippet 10

```
class A {
    int x;
}
public class Main {
    static void change(A a) {
        a = new A();
        a.x = 50;
    }
    public static void main(String[] args) {
        A obj = new A();
        change(obj);
        System.out.println(obj.x);
    }
}
```

✅ **Output:** `0`

Snippet 11

```
class A {
    int x = 10;
}
class Main {
    public static void main(String[] args) {
        A a = new A();
        a = null;
        System.out.println(a.x);
    }
}
```

✅ **Answer:** NullPointerException

Snippet 12

```
class Test {
    void show() {
        int x = 10;
    }
}
```

✅ **x stored in stack**

Snippet 13

```
class Test {
    static void show() {
        Test t = new Test();
    }
}
```

✅ **Object in heap, reference in stack**

Snippet 14

```
class Emp {
    int id;
    String name;
}
```

✅ **Reference variables inside object**

Snippet 15

```
class A {
    A() {
        System.out.println("A");
    }
}
class Main {
    public static void main(String[] args) {
        A a1 = new A();
        A a2 = new A();
    }
}
```

✅ **Output:**

```
A
A
```

📘 SESSION 7 – INHERITANCE, POLYMORPHISM & RELATIONSHIPS (JAVA)

🟢 EASY LEVEL MCQs (1–20)

1. Inheritance represents:
A) HAS-A relationship
B) USES-A relationship
C) IS-A relationship
D) PART-OF relationship

✅ **Answer:** C

---

2. Which keyword is used for inheritance in Java?
A) implements
B) extends
C) inherits
D) super

✅ **Answer:** B

---

3. Java supports which inheritance directly?
A) Multiple
B) Single
C) Hybrid
D) Circular

✅ **Answer:** B

4. Which is an example of single inheritance?
A) A → B → C
B) A ← B → C
C) A → B
D) A ↔ B

✅ **Answer:** C

5. Multilevel inheritance means:
A) One parent, many children
B) Many parents, one child
C) Chain of inheritance
D) No inheritance

✅ **Answer:** C

6. Hierarchical inheritance means:
A) One class inherits many
B) Many classes inherit one
C) Chain inheritance
D) Interface-based

✅ **Answer:** B

7. Method overloading is:
A) Runtime binding
B) Same method name, same params
C) Same method name, different params
D) Different method name

✅ **Answer:** C

8. Method overriding occurs in:
A) Same class
B) Different class without inheritance
C) Parent–child relationship
D) Interface only

✅ **Answer:** C

9. Compile-time polymorphism is achieved using:
A) Overriding
B) Inheritance
C) Overloading
D) Interfaces

✅ **Answer:** C

---

10. Runtime polymorphism is achieved using:
A) Overloading
B) Overriding
C) Static methods
D) Constructors

✅ **Answer:** B

---

11. Which keyword refers to parent class object?
A) this
B) new
C) parent
D) super

✅ **Answer:** D

---

12. Which keyword refers to current class object?
A) this
B) super
C) self
D) object

✅ **Answer:** A

---

13. Association represents:
A) IS-A
B) PART-OF
C) USES-A
D) NONE

✅ **Answer:** C

---

14. Aggregation is a:
A) Strong HAS-A

B) Weak HAS-A
C) IS-A
D) USES-A

✅ **Answer:** B

---

15. Composition is:
A) Weak HAS-A
B) Temporary association
C) Strong HAS-A
D) IS-A

✅ **Answer:** C

---

16. Child class object creation calls:
A) Child constructor only
B) Parent constructor only
C) Parent then child constructor
D) Random order

✅ **Answer:** C

---

17. Which method cannot be overridden?
A) public
B) protected
C) final
D) abstract

✅ **Answer:** C

---

18. Which access modifier allows overriding?
A) private
B) default
C) protected
D) static

✅ **Answer:** C

---

19. Overridden method must have:
A) Different name
B) Same name & signature

C) Different return type
D) Static keyword

✅ **Answer:** B

---

20. Which relationship uses constructor injection?
A) Inheritance
B) Association
C) Aggregation
D) Composition

✅ **Answer:** D

🟡 MEDIUM LEVEL MCQs (21–40)

21. Which inheritance is NOT supported using classes?
A) Single
B) Multilevel
C) Hierarchical
D) Multiple

✅ **Answer:** D

---

22. Multiple inheritance is supported using:
A) Abstract class
B) Interface
C) Object
D) Constructor

✅ **Answer:** B

---

23. Which method is bound at runtime?
A) static
B) private
C) final
D) overridden

✅ **Answer:** D

---

24. Which rule is correct for overriding?
A) Access level must be same or broader

B) Access can be reduced
C) Return type must differ
D) Method must be static

✅ **Answer:** A

---

25. Which keyword prevents inheritance?
A) private
B) final
C) static
D) protected

✅ **Answer:** B

---

26. Association lifetime is:
A) Same as object
B) Short-lived
C) Independent
D) Compile-time

✅ **Answer:** C

---

27. Which relationship is destroyed when parent is destroyed?
A) Association
B) Aggregation
C) Composition
D) Inheritance

✅ **Answer:** C

---

28. Which method cannot be overridden?
A) static
B) protected
C) public
D) abstract

✅ **Answer:** A

---

29. Which keyword is used to call parent constructor?
A) this
B) super

C) parent
D) base

✅ **Answer:** B

---

30. Which polymorphism resolves at compile time?
A) Overriding
B) Dynamic binding
C) Overloading
D) Method dispatch

✅ **Answer:** C

---

31. Which is true about `super()`?
A) Optional always
B) Must be first statement
C) Can be anywhere
D) Cannot be used

✅ **Answer:** B

---

32. Which relationship promotes reusability most?
A) Association
B) Aggregation
C) Composition
D) Inheritance

✅ **Answer:** D

---

33. Which binding is used in overriding?
A) Static binding
B) Early binding
C) Dynamic binding
D) Compile-time

✅ **Answer:** C

---

34. Which of the following supports runtime polymorphism?
A) Reference type of parent
B) Object of parent
C) Static methods
D) Constructors

✅ **Answer:** A

---

35. Can constructors be overridden?
A) Yes
B) No
C) Sometimes
D) Only abstract

✅ **Answer:** B

---

36. Overloaded methods differ by:
A) Return type only
B) Access modifier only
C) Parameter list
D) Method body

✅ **Answer:** C

---

37. Which causes method hiding?
A) Overriding static methods
B) Overloading
C) Abstract method
D) Constructor

✅ **Answer:** A

---

38. Which is true about hierarchical inheritance?
A) One child many parents
B) One parent many children
C) Chain structure
D) Cyclic

✅ **Answer:** B

---

39. Which relationship is tightly coupled?
A) Association
B) Aggregation
C) Composition
D) Interface

✅ **Answer:** C

40. Which method call is resolved using object type?
A) Static
B) Private
C) Overridden
D) Final

✅ **Answer:** C

🔴 HARD LEVEL MCQs (41–60)

41. Which is mandatory for runtime polymorphism?
A) Method overloading
B) Method overriding
C) Static methods
D) Constructors

✅ **Answer:** B

---

42. Which breaks inheritance chain?
A) super
B) final class
C) abstract method
D) protected

✅ **Answer:** B

---

43. Which is correct overriding rule?
A) Return type must differ
B) Access can be reduced
C) Checked exception must be same or narrower
D) Method must be static

✅ **Answer:** C

---

44. Which relationship is best for reuse without ownership?
A) Inheritance
B) Composition
C) Aggregation
D) Polymorphism

✅ **Answer:** C

45. Which is true about constructor chaining?
A) Child constructor runs first
B) Parent constructor runs first
C) Random order
D) Parallel

✅ **Answer:** B

46. Which cannot be inherited?
A) public methods
B) protected methods
C) private methods
D) default methods

✅ **Answer:** C

47. Which polymorphism uses method signature?
A) Runtime
B) Dynamic
C) Compile-time
D) Late binding

✅ **Answer:** C

48. Which scenario leads to ambiguity in C++ but not Java?
A) Multiple inheritance
B) Single inheritance
C) Hierarchical
D) Composition

✅ **Answer:** A

49. Which keyword resolves variable hiding?
A) this
B) super
C) static
D) final

✅ **Answer:** B

50. Which method call depends on reference type?
A) Overridden
B) Static
C) Instance
D) Abstract

✅ **Answer:** B

---

51. Which is NOT polymorphism?
A) Overloading
B) Overriding
C) Dynamic binding
D) Encapsulation

✅ **Answer:** D

---

52. Which relationship implies ownership?
A) Association
B) Aggregation
C) Composition
D) Interface

✅ **Answer:** C

---

53. Which binding occurs at runtime?
A) Static
B) Early
C) Dynamic
D) Compile-time

✅ **Answer:** C

---

54. Can we override a method with broader access?
A) No
B) Yes
C) Sometimes
D) Only abstract

✅ **Answer:** B

---

55. Which method is chosen at runtime?
A) Overloaded

B) Static
C) Overridden
D) Final

✅ **Answer:** C

---

56. Which keyword avoids name conflict?
A) static
B) this
C) super
D) final

✅ **Answer:** C

---

57. Which type of inheritance forms a tree?
A) Single
B) Multilevel
C) Hierarchical
D) Multiple

✅ **Answer:** C

---

58. Which relationship ensures strong lifecycle dependency?
A) Association
B) Aggregation
C) Composition
D) Inheritance

✅ **Answer:** C

---

59. Which method cannot participate in polymorphism?
A) public
B) static
C) protected
D) abstract

✅ **Answer:** B

---

60. Best design principle for flexibility is:
A) Tight coupling
B) Composition over inheritance

C) Global variables
D) Deep inheritance

✅ **Answer:** B

🧩 CODE SNIPPET MCQs (20)

Snippet 1 – Single Inheritance

```
class A {
    void show() { System.out.print("A"); }
}
class B extends A {}
```

✅ **Valid single inheritance**

Snippet 2 – Multilevel Inheritance

```
class A {}
class B extends A {}
class C extends B {}
```

✅ **Valid multilevel inheritance**

Snippet 3 – Hierarchical Inheritance

```
class A {}
class B extends A {}
class C extends A {}
```

✅ **Hierarchical inheritance**

Snippet 4 – Overloading

```
class Test {
    void show(int a) {}
    void show(double a) {}
}
```

✅ **Compile-time polymorphism**

Snippet 5 – Overriding

```
class A {
    void show() { System.out.print("A"); }
}
class B extends A {
    void show() { System.out.print("B"); }
}
```

✅ **Runtime polymorphism**

Snippet 6 – Runtime Dispatch

```
A obj = new B();
obj.show();
```

✅ **Output:** `B`

Snippet 7 – Static Method Hiding

```
class A {
    static void show() { System.out.print("A"); }
}
class B extends A {
    static void show() { System.out.print("B"); }
}
A a = new B();
a.show();
```

✅ **Output:** `A`

Snippet 8 – super keyword

```
class A {
    int x = 10;
}
class B extends A {
    int x = 20;
    void show() {
        System.out.print(super.x);
    }
}
```

✅ **Output:** `10`

Snippet 9 – this keyword

```
class A {
    int x;
    A(int x) {
        this.x = x;
    }
}
```

✅ **Correct use of this**

Snippet 10 – Constructor chaining

```
class A {
    A() { System.out.print("A"); }
}
class B extends A {
    B() { System.out.print("B"); }
}
new B();
```

✅ **Output:** `AB`

Snippet 11 – Association

```
class Car {}
class Driver {
    Car c;
}
```

✅ **Association**

Snippet 12 – Aggregation

```
class Engine {}
class Car {
    Engine e;
    Car(Engine e) { this.e = e; }
}
```

✅ **Aggregation**

Snippet 13 – Composition

```
class Engine {}
class Car {
    Engine e = new Engine();
}
```

✅ **Composition**

Snippet 14 – Invalid Override

```
class A {
    final void show() {}
}
class B extends A {
    void show() {}
}
```

✅ **Compile-time error**

Snippet 15 – Overload vs Override

```
class A {
    void show(int a) {}
}
```

```
class B extends A {
    void show(double a) {}
}
```

✅ **Overloading, not overriding**

Snippet 16 – super()

```
class A {
    A(int x) {}
}
class B extends A {
    B() { super(10); }
}
```

✅ **Valid constructor chaining**

Snippet 17 – Polymorphic reference

```
A obj;
obj = new B();
```

✅ **Valid**

Snippet 18 – Method selection

```
class A {
    void show() {}
}
class B extends A {
    void show() {}
}
A a = new B();
```

✅ **Method chosen at runtime**

Snippet 19 – Illegal inheritance

```
final class A {}
class B extends A {}
```

✅ **Compile-time error**

Snippet 20 – Encapsulated Employee

```
class Employee {
    private int id;
    public int getId() { return id; }
}
```

✅ **Encapsulation + inheritance-ready**

📘 SESSION 8 – CASTING, ABSTRACTION & INTERFACES (JAVA)

🟢 EASY LEVEL MCQs (1–15)

1. Upcasting means:
A) Child reference to parent object
B) Parent reference to child object
C) Explicit casting
D) Invalid casting

✅ **Answer:** A

---

2. Upcasting is:
A) Explicit
B) Implicit
C) Unsafe
D) Not allowed

✅ **Answer:** B

---

3. Downcasting means:
A) Parent reference to child object

B) Child reference to parent object
C) Interface to class
D) Object to primitive

✅ **Answer:** A

---

4. Downcasting requires:
A) No casting
B) Constructor
C) Explicit cast
D) JVM option

✅ **Answer:** C

---

5. Which keyword checks object type before downcasting?
A) this
B) super
C) instanceof
D) new

✅ **Answer:** C

---

6. Abstract class is declared using:
A) final
B) static
C) abstract
D) interface

✅ **Answer:** C

---

7. Abstract method has:
A) Body
B) Parameters only
C) No body
D) Static body

✅ **Answer:** C

---

8. Can we create object of abstract class?
A) Yes
B) No

C) Sometimes
D) Only in same package

✅ **Answer:** B

---

9. Interface is declared using:
A) class
B) abstract
C) interface
D) implements

✅ **Answer:** C

---

10. Interface supports:
A) Single inheritance
B) Multiple inheritance
C) No inheritance
D) Hybrid inheritance

✅ **Answer:** B

---

11. Interface methods are by default:
A) private
B) protected
C) public abstract
D) static

✅ **Answer:** C

---

12. Interface variables are by default:
A) private
B) instance
C) public static final
D) local

✅ **Answer:** C

---

13. Abstract class can have:
A) Only abstract methods
B) Only concrete methods
C) Both
D) None

✅ **Answer:** C

---

14. Which keyword is used to implement interface?
A) extends
B) implements
C) inherits
D) super

✅ **Answer:** B

---

15. Multiple inheritance in Java is achieved using:
A) Classes
B) Abstract classes
C) Interfaces
D) Objects

✅ **Answer:** C

🟡 MEDIUM LEVEL MCQs (16–30)

16. Which casting is always safe?
A) Downcasting
B) Upcasting
C) Explicit casting
D) Object casting

✅ **Answer:** B

---

17. Which causes ClassCastException?
A) Safe upcasting
B) Invalid downcasting
C) Abstract method call
D) Interface implementation

✅ **Answer:** B

---

18. Which is true about abstract class?
A) Cannot have constructor
B) Cannot have variables
C) Can have constructor
D) Cannot have methods

✅ **Answer:** C

---

19. Interface can contain (Java 8+):
A) Only abstract methods
B) Default methods
C) Constructors
D) Instance variables

✅ **Answer:** B

---

20. Which reference supports runtime polymorphism?
A) Child reference → child object
B) Parent reference → child object
C) Parent reference → parent object
D) Child reference → parent object

✅ **Answer:** B

---

21. Which keyword prevents invalid downcasting?
A) this
B) super
C) instanceof
D) new

✅ **Answer:** C

---

22. Abstract class provides:
A) Full abstraction
B) Partial abstraction
C) No abstraction
D) Multiple inheritance

✅ **Answer:** B

---

23. Interface provides:
A) Partial abstraction
B) Full abstraction
C) No abstraction
D) Data hiding

✅ **Answer:** B

24. A class implementing interface must:
A) Override all methods
B) Override some methods
C) Override none
D) Be abstract always

✅ **Answer:** A

25. Which can extend another interface?
A) Class
B) Object
C) Interface
D) Method

✅ **Answer:** C

26. Which is valid?
A) class A implements B, C
B) interface A implements B
C) class A extends B, C
D) interface A extends B, C

✅ **Answer:** D

27. Abstract method cannot be:
A) public
B) protected
C) private
D) default

✅ **Answer:** C

28. Which supports multiple inheritance of behavior?
A) Class
B) Abstract class
C) Interface (default methods)
D) Object

✅ **Answer:** C

29. Upcasting helps in:
A) Compile-time binding
B) Runtime polymorphism
C) Memory allocation
D) Garbage collection

✅ **Answer:** B

---

30. Which is NOT allowed in interface?
A) Static method
B) Default method
C) Instance variable
D) Abstract method

✅ **Answer:** C

🔴 HARD LEVEL MCQs (31–45)

31. Which statement about casting is correct?
A) Upcasting needs explicit cast
B) Downcasting is always safe
C) Downcasting may fail at runtime
D) Casting changes object type

✅ **Answer:** C

---

32. Which combination avoids ClassCastException?
A) Direct downcast
B) instanceof + downcast
C) Explicit cast only
D) Abstract reference

✅ **Answer:** B

---

33. Which reference determines method call at runtime?
A) Reference type
B) Object type
C) Class type
D) Interface type

✅ **Answer:** B

---

34. Which method cannot be abstract?
A) static
B) protected
C) public
D) default

✅ **Answer:** A

---

35. Interface variables are:
A) Mutable
B) Instance-based
C) Constants
D) Private

✅ **Answer:** C

---

36. Which supports constructor?
A) Interface
B) Abstract class
C) Both
D) None

✅ **Answer:** B

---

37. Which is true for interface inheritance?
A) Single only
B) Multiple allowed
C) No inheritance
D) Runtime only

✅ **Answer:** B

---

38. Which casting is checked at runtime?
A) Primitive casting
B) Upcasting
C) Downcasting
D) Compile-time casting

✅ **Answer:** C

---

39. Which enables flexible design?
A) Tight coupling

B) Downcasting everywhere
C) Programming to interface
D) Concrete classes only

✅ **Answer:** C

---

40. Which breaks abstraction?
A) Using interface
B) Using abstract class
C) Accessing concrete class directly
D) Upcasting

✅ **Answer:** C

---

41. Which can implement multiple interfaces?
A) Interface
B) Abstract class
C) Concrete class
D) All

✅ **Answer:** C

---

42. Which is resolved at compile time?
A) Overriding
B) Downcasting
C) Overloading
D) Dynamic dispatch

✅ **Answer:** C

---

43. Which design uses upcasting heavily?
A) Utility classes
B) Polymorphic APIs
C) Static helpers
D) Singleton

✅ **Answer:** B

---

44. Which causes ambiguity without interfaces?
A) Single inheritance
B) Multilevel inheritance

C) Multiple inheritance of classes
D) Hierarchical inheritance

✅ **Answer:** C

---

45. Best practice for abstraction is:
A) Use concrete classes
B) Use abstract classes only
C) Program to interfaces
D) Avoid casting

✅ **Answer:** C

🧩 CODE SNIPPET MCQs (15)

Snippet 1 – Upcasting

```
class A {}
class B extends A {}
A a = new B();
```

✅ **Valid upcasting**

Snippet 2 – Downcasting

```
A a = new B();
B b = (B) a;
```

✅ **Valid downcasting**

Snippet 3 – Invalid Downcasting

```
A a = new A();
B b = (B) a;
```

✅ **Runtime Exception:** ClassCastException

Snippet 4 – instanceof check

```
if(a instanceof B) {
    B b = (B) a;
}
```

✅ **Safe downcasting**

Snippet 5 – Abstract class

```
abstract class A {
    abstract void show();
}
```

✅ **Valid**

Snippet 6 – Abstract + concrete

```
abstract class A {
    void display() {}
}
```

✅ **Valid**

Snippet 7 – Interface

```
interface I {
    void show();
}
```

✅ **public abstract method**

Snippet 8 – Implement interface

```
class A implements I {
    public void show() {}
}
```

✅ **Valid**

Snippet 9 – Multiple interfaces

```
interface A {}
interface B {}
class C implements A, B {}
```

✅ **Valid multiple inheritance**

Snippet 10 – Interface upcasting

```
I obj = new A();
```

✅ **Valid**

Snippet 11 – Runtime polymorphism

```
A a = new B();
a.show();
```

✅ **Method of B executed**

Snippet 12 – Abstract reference

```
A a;
```

✅ **Allowed**

Snippet 13 – Interface variables

```
interface I {
    int x = 10;
}
```

✅ **public static final**

Snippet 14 – Default method

```
interface I {
    default void show() {
        System.out.println("Hello");
    }
}
```

✅ **Valid (Java 8+)**

Snippet 15 – Casting does not change object

```
A a = new B();
A a2 = (A) a;
```

✅ **Same object, different reference**

📘 SESSIONS 9 & 10 – FINAL, FUNCTIONAL INTERFACES, LAMBDA, INNER CLASSES & ENUM (JAVA)

🟢 EASY LEVEL MCQs (1–20)

1. `final` variable means:
A) Variable can be modified
B) Variable cannot be modified
C) Variable is static
D) Variable is private

✅ **Answer:** B

---

2. `final` method cannot be:
A) Accessed
B) Inherited
C) Overridden
D) Overloaded

✅ **Answer:** C

---

3. `final` class cannot be:
A) Instantiated
B) Extended
C) Used
D) Compiled

✅ **Answer:** B

---

4. Functional interface contains:
A) No abstract methods
B) One abstract method
C) Two abstract methods
D) Only default methods

✅ **Answer:** B

---

5. Which annotation marks functional interface?
A) @Override
B) @Functional
C) @FunctionalInterface
D) @Lambda

✅ **Answer:** C

---

6. Lambda expression is used to implement:
A) Class
B) Abstract class
C) Functional interface
D) Enum

✅ **Answer:** C

---

7. Which symbol is used in lambda expression?
A) ::
B) =>
C) ->
D) ==

✅ **Answer:** C

---

8. Interface default methods were introduced in:
A) Java 5
B) Java 6
C) Java 7
D) Java 8

✅ **Answer:** D

---

9. Static methods in interface are:
A) Inherited
B) Overridden
C) Hidden
D) Not inherited

✅ **Answer:** D

---

10. Inner class means:
A) Class inside package
B) Class inside method
C) Class inside another class
D) Class inside interface

✅ **Answer:** C

---

11. Which inner class can access static members only?
A) Regular inner
B) Anonymous
C) Static inner
D) Method local

✅ **Answer:** C

---

12. Enum is used to represent:
A) Objects
B) Constants
C) Interfaces
D) Methods

✅ **Answer:** B

---

13. Enum constants are:
A) Objects

B) Variables
C) Methods
D) Classes

✅ **Answer:** A

---

14. Array of objects stores:
A) Objects directly
B) References to objects
C) Methods
D) Classes

✅ **Answer:** B

---

15. When array of objects is created, how many objects are created initially?
A) Array + all objects
B) Only objects
C) Only array object
D) None

✅ **Answer:** C

---

16. Anonymous class is used to:
A) Reuse code
B) Create object with class definition
C) Create named class
D) Extend enum

✅ **Answer:** B

---

17. Lambda expression reduces:
A) Performance
B) Readability
C) Boilerplate code
D) Memory

✅ **Answer:** C

---

18. Enum implicitly extends:
A) Object
B) Enum

C) Class
D) Interface

✅ **Answer:** B

---

19. Functional interface can have:
A) Only one method
B) One abstract + many default methods
C) Only static methods
D) No methods

✅ **Answer:** B

---

20. Which inner class has no name?
A) Static inner
B) Regular inner
C) Anonymous
D) Method local

✅ **Answer:** C

🟡 MEDIUM LEVEL MCQs (21–40)

21. `final` reference variable means:
A) Object cannot change
B) Reference cannot change
C) Both cannot change
D) None

✅ **Answer:** B

---

22. Which can be `final`?
A) Class
B) Method
C) Variable
D) All

✅ **Answer:** D

---

23. Default methods in interface help to:
A) Achieve abstraction

B) Avoid breaking old implementations
C) Improve performance
D) Replace abstract class

✅ **Answer:** B

---

24. Interface private methods were introduced in:
A) Java 7
B) Java 8
C) Java 9
D) Java 11

✅ **Answer:** C

---

25. Lambda expression can access:
A) Only static variables
B) Only instance variables
C) Final or effectively final variables
D) No variables

✅ **Answer:** C

---

26. Which inner class needs object of outer class?
A) Static inner
B) Anonymous
C) Regular inner
D) Enum

✅ **Answer:** C

---

27. Method-local inner class scope is:
A) Whole class
B) Whole package
C) Only method
D) Whole JVM

✅ **Answer:** C

---

28. Which is NOT allowed in enum?
A) Constructors
B) Methods

C) Variables
D) Inheritance

✅ **Answer:** D

---

29. Which enum constructor is allowed?
A) public
B) protected
C) private
D) All

✅ **Answer:** C

---

30. Functional interface allows:
A) Multiple inheritance
B) Lambda implementation
C) Runtime polymorphism
D) Constructor overriding

✅ **Answer:** B

---

31. Lambda expression is converted into:
A) Class file
B) Object
C) Functional interface instance
D) Method

✅ **Answer:** C

---

32. Anonymous class is created using:
A) extends
B) implements
C) new keyword
D) lambda

✅ **Answer:** C

---

33. Static inner class can access:
A) All outer members
B) Only instance members
C) Only static members
D) Nothing

✅ **Answer:** C

---

34. Which enum method returns all constants?
A) valueOf()
B) values()
C) ordinal()
D) name()

✅ **Answer:** B

---

35. Which enum method returns index?
A) name()
B) valueOf()
C) ordinal()
D) compareTo()

✅ **Answer:** C

---

36. How many abstract methods allowed in functional interface?
A) 0
B) 1
C) 2
D) Unlimited

✅ **Answer:** B

---

37. Lambda expression supports:
A) Inheritance
B) Polymorphism
C) Functional programming
D) Serialization

✅ **Answer:** C

---

38. Which interface method cannot be overridden?
A) default
B) static
C) abstract
D) private

✅ **Answer:** B

39. Inner classes increase:
A) Performance
B) Security
C) Readability
D) Encapsulation

✅ **Answer:** D

40. Array of objects memory allocation happens in:
A) Stack
B) Heap
C) Method area
D) Register

✅ **Answer:** B

🔴 HARD LEVEL MCQs (41–60)

41. `final` method ensures:
A) Method hiding
B) Method overriding
C) Method binding
D) Method cannot be overridden

✅ **Answer:** D

42. Which breaks lambda compatibility?
A) Adding default method
B) Adding static method
C) Adding abstract method
D) Adding private method

✅ **Answer:** C

43. Which inner class can declare static members?
A) Regular
B) Method local
C) Anonymous
D) Static inner

✅ **Answer:** D

44. Lambda expressions are:
A) Objects
B) Methods
C) Anonymous functions
D) Threads

✅ **Answer:** C

45. Which is preferred over anonymous class?
A) Enum
B) Lambda expression
C) Abstract class
D) Static method

✅ **Answer:** B

46. Enum constants are created:
A) At runtime
B) At compile time
C) At class loading time
D) When accessed

✅ **Answer:** C

47. Which inner class increases memory usage least?
A) Regular inner
B) Static inner
C) Anonymous
D) Method local

✅ **Answer:** B

48. Functional interface may contain:
A) Static + default + private methods
B) Only abstract
C) Only static
D) Only default

✅ **Answer:** A

49. Lambda expressions cannot:
A) Throw exception
B) Access instance variable
C) Have multiple statements
D) Change local variable

✅ **Answer:** D

---

50. Which is NOT true about enum?
A) Enum can implement interfaces
B) Enum can extend class
C) Enum constants are objects
D) Enum can have methods

✅ **Answer:** B

---

51. Anonymous class is useful when:
A) Class is reused
B) One-time implementation needed
C) Performance critical
D) Multiple inheritance

✅ **Answer:** B

---

52. Which provides best encapsulation?
A) Enum
B) Static inner class
C) Anonymous class
D) Lambda

✅ **Answer:** B

---

53. Lambda expression is syntactic sugar for:
A) Abstract class
B) Interface implementation
C) Method overriding
D) Constructor

✅ **Answer:** B

---

54. Which cannot be final?
A) Constructor

B) Method
C) Class
D) Variable

✅ **Answer:** A

---

55. Enum constructor is called:
A) Explicitly
B) By JVM
C) By programmer
D) By main method

✅ **Answer:** B

---

56. Which inner class has access to local variables?
A) Static inner
B) Regular inner
C) Method local
D) Enum

✅ **Answer:** C

---

57. Functional interface ensures:
A) Multiple abstraction
B) Single responsibility
C) Single abstract behavior
D) No abstraction

✅ **Answer:** C

---

58. Lambda expression reduces:
A) Runtime overhead
B) Lines of code
C) Memory leaks
D) CPU cycles

✅ **Answer:** B

---

59. Enum comparison should use:
A) equals()
B) ==

C) compareTo()
D) hashCode()

✅ **Answer:** B

---

60. Best replacement for anonymous class (Java 8+)?
A) Abstract class
B) Static inner class
C) Lambda expression
D) Enum

✅ **Answer:** C

🧩 CODE SNIPPET MCQs (20)

Snippet 1 – final variable

```
final int x = 10;
x = 20;
```

✅ **Compile-time error**

Snippet 2 – final reference

```
final int[] a = {1,2};
a[0] = 5;
```

✅ **Valid**

Snippet 3 – final method

```
class A {
    final void show() {}
}
class B extends A {
    void show() {}
}
```

✅ **Compile-time error**

Snippet 4 – Functional interface

```
@FunctionalInterface
interface I {
    void show();
}
```

✅ **Valid**

Snippet 5 – Lambda expression

```
I obj = () -> System.out.println("Hello");
```

✅ **Valid lambda**

Snippet 6 – Anonymous class

```
I obj = new I() {
    public void show() {
        System.out.println("Hi");
    }
};
```

✅ **Valid**

Snippet 7 – Lambda vs anonymous

```
Runnable r = () -> System.out.println("Run");
```

✅ **Preferred lambda**

Snippet 8 – Default method

```
interface I {
    default void show() {}
}
```

✅ **Java 8 feature**

Snippet 9 – Static interface method

```
interface I {
    static void show() {}
}
```

✅ **Not inherited**

Snippet 10 – Inner class

```
class A {
    class B {}
}
```

✅ **Regular inner class**

Snippet 11 – Static inner class

```
class A {
    static class B {}
}
```

✅ **Static inner class**

Snippet 12 – Method local inner class

```
void m() {
    class A {}
}
```

✅ **Valid**

Snippet 13 – Anonymous inner class

```
Thread t = new Thread() {
    public void run() {}
};
```

✅ **Anonymous inner class**

Snippet 14 – Enum

```
enum Day { MON, TUE, WED }
```

✅ **Enum constants**

Snippet 15 – Enum usage

```
Day d = Day.MON;
```

✅ **Valid**

Snippet 16 – Enum methods

```
Day[] days = Day.values();
```

✅ **All constants**

Snippet 17 – Array of objects

```
Employee[] e = new Employee[3];
```

✅ **Only array object created**

Snippet 18 – Objects in array

```
e[0] = new Employee();
```

✅ **Employee object created**

Snippet 19 – Lambda with parameter

```
interface Calc {
    int add(int a, int b);
}
Calc c = (a,b) -> a + b;
```

✅ **Valid**

Snippet 20 – Effectively final

```
int x = 10;
Runnable r = () -> System.out.println(x);
```

✅ **Valid (effectively final)**

📘 SESSION 11 – ACCESS MODIFIERS, PACKAGES & CONSTRUCTOR CHAINING (JAVA)

🟢 EASY LEVEL MCQs (1–15)

1. Which access modifier gives widest access?
A) private
B) default
C) protected
D) public

✅ **Answer:** D

---

2. Which access modifier gives package-level access only?
A) public
B) private
C) protected
D) default

✅ **Answer:** D

3. Which access modifier restricts access within same class only?
A) public
B) protected
C) default
D) private

✅ **Answer:** D

4. Which access modifier allows subclass access outside package?
A) default
B) private
C) protected
D) public

✅ **Answer:** C

5. Package in Java is used to:
A) Store variables
B) Store objects
C) Group related classes
D) Compile programs

✅ **Answer:** C

6. Which keyword defines a package?
A) import
B) package
C) include
D) namespace

✅ **Answer:** B

7. Package statement must be written:
A) Anywhere
B) At the end
C) At the top
D) Inside class

✅ **Answer:** C

8. Which keyword is used to access classes from package?
A) package
B) import
C) include
D) use

✅ **Answer:** B

---

9. Static import is used to:
A) Import classes
B) Import packages
C) Import static members
D) Import constructors

✅ **Answer:** C

---

10. Constructor chaining means:
A) Calling methods
B) Calling multiple constructors
C) Calling same constructor repeatedly
D) Calling static blocks

✅ **Answer:** B

---

11. Which keyword is used for constructor chaining in same class?
A) super
B) this
C) new
D) static

✅ **Answer:** B

---

12. Which keyword is used to call parent constructor?
A) this
B) parent
C) super
D) extends

✅ **Answer:** C

---

13. Protected members are accessible within:
A) Same class only

B) Same package only
C) Same package + subclass
D) Anywhere

✅ **Answer:** C

---

14. Default access modifier keyword is:
A) default
B) none
C) package
D) internal

✅ **Answer:** B

---

15. Which is valid static import?
A) import static java.lang.Math;
B) import java.lang.Math._;_
C) import static java.lang.Math.;
D) import Math.static.*;

✅ **Answer:** C

🟡 MEDIUM LEVEL MCQs (16–30)

16. Which members are NOT inherited?
A) public
B) protected
C) default
D) private

✅ **Answer:** D

---

17. Which access modifier is more restrictive than protected?
A) public
B) private
C) default
D) none

✅ **Answer:** C

---

18. Which statement is true about protected access?
A) Accessible anywhere
B) Accessible only in same class
C) Accessible in subclass outside package
D) Accessible only via object

✅ **Answer:** C

---

19. Can constructors be protected?
A) No
B) Yes
C) Only default
D) Only public

✅ **Answer:** B

---

20. Static import allows access to static members:
A) With class name
B) Without class name
C) With object
D) With constructor

✅ **Answer:** B

---

21. Which statement is correct?
A) Multiple package statements allowed
B) Only one package statement allowed
C) Package statement optional
D) Package statement after import

✅ **Answer:** B

---

22. Which is correct order in Java file?
A) import → package → class
B) class → package → import
C) package → import → class
D) import → class → package

✅ **Answer:** C

---

23. Which constructor is called implicitly if not specified?
A) Parameterized

B) Copy
C) Default
D) Private

✅ **Answer:** C

---

24. Can static import import non-static members?
A) Yes
B) No
C) Sometimes
D) Only variables

✅ **Answer:** B

---

25. Protected members outside package can be accessed using:
A) Object reference
B) Class name
C) Subclass reference
D) Any reference

✅ **Answer:** C

---

26. Which constructor chaining rule is correct?
A) this() can be anywhere
B) super() can be after statements
C) this()/super() must be first
D) Both optional always

✅ **Answer:** C

---

27. Which package is imported by default?
A) java.io
B) java.util
C) java.lang
D) java.sql

✅ **Answer:** C

---

28. Which is NOT allowed?
A) public class
B) protected constructor

C) private top-level class
D) default access class

✅ **Answer:** C

---

29. Can a class belong to multiple packages?
A) Yes
B) No
C) Sometimes
D) Only static

✅ **Answer:** B

---

30. Which access modifier supports encapsulation best?
A) public
B) protected
C) default
D) private

✅ **Answer:** D

🔴 HARD LEVEL MCQs (31–45)

31. Protected member outside package is accessed via:
A) Parent class object
B) Child class object
C) Parent reference
D) Static access

✅ **Answer:** B

---

32. Which breaks constructor chaining?
A) Calling super() first
B) Calling this() first
C) Using both this() & super()
D) Missing constructor

✅ **Answer:** C

---

33. Which is true about default access?
A) Package-private

B) Same as protected
C) Same as public
D) Same as private

✅ **Answer:** A

---

34. Which statement about static import is correct?
A) Improves performance
B) Reduces readability sometimes
C) Mandatory for static methods
D) Replaces inheritance

✅ **Answer:** B

---

35. Which is valid?
A) this(); super();
B) super(); this();
C) this();
D) Both A & B

✅ **Answer:** C

---

36. Protected constructor is mainly used for:
A) Object creation restriction
B) Inheritance control
C) Encapsulation
D) Polymorphism

✅ **Answer:** B

---

37. Which access modifier allows widest subclass access?
A) private
B) default
C) protected
D) public

✅ **Answer:** D

---

38. Which is correct about package visibility?
A) default > protected
B) protected > default

C) both same
D) unrelated

✅ **Answer:** B

---

39. Static import ambiguity occurs when:
A) Same method names imported
B) Multiple classes imported
C) Same package imported
D) Constructors imported

✅ **Answer:** A

---

40. Which cannot be accessed outside package even by subclass?
A) public
B) protected
C) default
D) static

✅ **Answer:** C

---

41. Which is false?
A) private members not inherited
B) protected inherited
C) default inherited across packages
D) public inherited everywhere

✅ **Answer:** C

---

42. Which is correct for constructor chaining across packages?
A) default constructor works
B) private constructor works
C) public/protected constructor works
D) static constructor works

✅ **Answer:** C

---

43. Static import is resolved at:
A) Runtime
B) Compile time
C) Load time
D) Execution

✅ **Answer:** B

---

44. Which improves code modularity?
A) Static import
B) Packages
C) Default access
D) Anonymous classes

✅ **Answer:** B

---

45. Which access modifier is most restrictive overall?
A) protected
B) default
C) private
D) public

✅ **Answer:** C

🧩 CODE SNIPPET MCQs (15)

Snippet 1 – private access

```
class A {
    private int x = 10;
}
class B {
    void show() {
        // System.out.println(new A().x);
    }
}
```

✅ **Compile-time error**

Snippet 2 – default access

```
class A {
    int x = 10;
}
```

✅ **Accessible only in same package**

Snippet 3 – protected access (same package)

```
class A {
    protected int x = 10;
}
class B extends A {
    void show() {
        System.out.println(x);
    }
}
```

✅ **Valid**

Snippet 4 – protected outside package

```
class B extends A {
    void show() {
        System.out.println(x);
    }
}
```

✅ **Valid via inheritance**

Snippet 5 – invalid protected access

```
A a = new A();
System.out.println(a.x);
```

✅ **Compile-time error (outside package)**

Snippet 6 – package statement

```
package com.cdac.demo;
```

✅ **Must be first line**

Snippet 7 – import statement

```
import java.util.ArrayList;
```

✅ **Single class import**

Snippet 8 – wildcard import

```
import java.util.*;
```

✅ **All classes in package**

Snippet 9 – static import

```
import static java.lang.Math.PI;
```

✅ **Valid**

Snippet 10 – static import usage

```
System.out.println(PI);
```

✅ **Works without Math.**

Snippet 11 – constructor chaining

```
class A {
    A() {
        this(10);
    }
    A(int x) {}
}
```

✅ **Valid**

Snippet 12 – super constructor

```
class A {
    A(int x) {}
}
class B extends A {
    B() {
        super(10);
    }
}
```

✅ **Valid**

Snippet 13 – invalid chaining

```
class A {
    A() {
        super();
    }
}
```

✅ **Compile-time error (Object has no explicit super())**

Snippet 14 – protected constructor

```
class A {
    protected A() {}
}
class B extends A {}
```

✅ **Valid**

Snippet 15 – default constructor absence

```
class A {
    A(int x) {}
}
new A();
```

✅ **Compile-time error**

📘 SESSION 12 – GARBAGE COLLECTION IN JAVA

🟢 EASY LEVEL MCQs (1–10)

1. Garbage collection in Java is used to:
A) Free stack memory
B) Free heap memory
C) Free method area
D) Free registers

✅ **Answer:** B

---

2. Which memory is managed by Garbage Collector?
A) Stack
B) Heap
C) Method area
D) CPU cache

✅ **Answer:** B

---

3. Who performs garbage collection in Java?
A) Programmer
B) Compiler
C) JVM
D) OS

✅ **Answer:** C

---

4. Which keyword makes an object eligible for GC?
A) new
B) this
C) null
D) static

✅ **Answer:** C

---

5. `System.gc()` means:
A) JVM must run GC
B) JVM may run GC
C) JVM stops program
D) JVM clears stack

✅ **Answer:** B

---

6. Which method is called before object destruction?
A) destroy()
B) clean()
C) finalize()
D) gc()

✅ **Answer:** C

---

7. Garbage collection happens:
A) At compile time
B) At load time
C) At runtime
D) At shutdown only

✅ **Answer:** C

---

8. Which object is eligible for GC?
A) Referenced by static variable
B) Referenced by local variable
C) Not referenced by any variable
D) Active object

✅ **Answer:** C

---

9. Can we force JVM to run GC?
A) Yes, always
B) No, never
C) JVM decides
D) Only once

✅ **Answer:** C

10. Default value of reference after nulling is:
A) 0
B) garbage
C) null
D) undefined

✅ **Answer:** C

🟡 MEDIUM LEVEL MCQs (11–20)

11. Which is NOT a way to make object eligible for GC?
A) Nulling reference
B) Reassigning reference
C) Object out of scope
D) Using static variable

✅ **Answer:** D

---

12. Reassigning reference variable means:
A) Object destroyed immediately
B) Reference points to new object
C) Both objects destroyed
D) JVM crash

✅ **Answer:** B

---

13. Island of isolation means:
A) Single unreachable object
B) Group of objects referencing each other only
C) Object with static reference
D) Object in stack

✅ **Answer:** B

---

14. When does `finalize()` execute?
A) Before GC
B) During GC (if at all)
C) After GC always
D) At program end

✅ **Answer:** B

---

15. `finalize()` is called:
A) Exactly once always
B) Multiple times
C) At JVM's discretion
D) Immediately after null

✅ **Answer:** C

---

16. Which memory leak situation is possible in Java?
A) No GC
B) Unused objects with references
C) Stack overflow
D) Final variables

✅ **Answer:** B

---

17. Static references affect GC because:
A) They live in stack
B) They prevent GC
C) They speed GC
D) They trigger GC

✅ **Answer:** B

---

18. Which statement is TRUE?
A) GC deletes object immediately
B) GC runs on demand
C) GC runs when JVM wants
D) GC frees stack memory

✅ **Answer:** C

---

19. Which is deprecated in modern Java?
A) System.gc()
B) finalize()
C) Object class
D) Heap memory

✅ **Answer:** B

---

20. Which objects are GC roots?
A) Heap objects
B) Stack references
C) Local variables
D) Both B and C

✅ **Answer:** D

🔴 HARD LEVEL MCQs (21–30)

21. Which scenario creates island of isolation?
A) Object refers to null
B) Two objects refer to each other only
C) Object with static reference
D) Object with local reference

✅ **Answer:** B

---

22. Which is true about `finalize()`?
A) Guaranteed execution
B) Called by programmer
C) May never be called
D) Always called once

✅ **Answer:** C

---

23. Which GC request method is preferred?
A) System.gc()
B) Runtime.getRuntime().gc()
C) Both are hints only
D) None

✅ **Answer:** C

---

24. Which object is NOT eligible for GC?
A) Object with no reference
B) Object with static reference
C) Object out of method scope
D) Island of isolation

✅ **Answer:** B

---

25. Which best describes Java GC?
A) Deterministic
B) Manual
C) Non-deterministic
D) Compile-time

✅ **Answer:** C

---

26. Why finalize() is discouraged?
A) Performance issues
B) Unpredictable execution
C) Security risk
D) All of the above

✅ **Answer:** D

---

27. Which causes memory leak even with GC?
A) Circular references
B) Static collections holding objects
C) finalize()
D) Null references

✅ **Answer:** B

---

28. Which memory is NOT cleaned by GC?
A) Heap
B) Stack
C) Young generation
D) Old generation

✅ **Answer:** B

---

29. Which inheritance scenario affects GC?
A) Parent-child references alive
B) No references alive
C) Protected access
D) Package inheritance

✅ **Answer:** A

---

30. Best practice instead of finalize():
A) Rely on GC

B) try-catch
C) try-with-resources / explicit close
D) static blocks

✅ **Answer:** C

🧩 CODE SNIPPETS (GC CONCEPTS)

Snippet 1 – Nulling reference

```
A a = new A();
a = null;
```

✅ **Object eligible for GC**

Snippet 2 – Reassigning reference

```
A a = new A();
a = new A();
```

✅ **First object eligible for GC**

Snippet 3 – Island of isolation

```
A a1 = new A();
A a2 = new A();
a1.ref = a2;
a2.ref = a1;
a1 = null;
a2 = null;
```

✅ **Both objects eligible for GC**

Snippet 4 – Requesting GC

```
System.gc();
```

✅ **Only a request, not guarantee**

Snippet 5 – Runtime GC

```
Runtime.getRuntime().gc();
```

✅ **Same as System.gc()**

Snippet 6 – finalize() override

```
class Test {
    @Override
    protected void finalize() throws Throwable {
        System.out.println("Finalize called");
    }
}
```

✅ **May or may not execute**

Snippet 7 – finalize() misconception

```
Test t = new Test();
t = null;
System.gc();
```

✅ **No guarantee finalize() runs**

Snippet 8 – Static reference prevents GC

```
class A {
    static Test t = new Test();
}
```

✅ **Object not eligible for GC**

Snippet 9 – Local scope GC

```
void m() {
    Test t = new Test();
}
```

✅ **Eligible after method ends**

Snippet 10 – Better cleanup

```
try (Resource r = new Resource()) {
    // use resource
}
```

✅ **Preferred over finalize()**

📘 SESSIONS 13 & 14 – WRAPPER CLASSES & STRING CLASSES (JAVA)

🟢 EASY LEVEL MCQs (1–15)

1. Wrapper class is used to:
A) Create objects
B) Convert primitive to object
C) Improve performance
D) Replace primitives

✅ **Answer:** B

---

2. Which is a wrapper class for `int`?
A) Int
B) Integer
C) intWrapper
D) Number

✅ **Answer:** B

---

3. Which wrapper class is immutable?
A) Integer
B) StringBuffer
C) StringBuilder
D) All

✅ **Answer:** A

---

4. Autoboxing means:
A) Object → primitive
B) Primitive → object
C) Object → object
D) Primitive → primitive

✅ **Answer:** B

---

5. Unboxing means:
A) Primitive → object
B) Object → primitive
C) Object → object
D) None

✅ **Answer:** B

---

6. Which class is immutable?
A) String
B) StringBuffer
C) StringBuilder
D) All

✅ **Answer:** A

---

7. Which class is thread-safe?
A) String
B) StringBuilder
C) StringBuffer
D) Wrapper classes

✅ **Answer:** C

---

8. Which class is fastest for string manipulation?
A) String
B) StringBuffer
C) StringBuilder
D) All same

✅ **Answer:** C

---

9. String literals are stored in:
A) Heap only
B) Stack
C) String pool
D) Method area

✅ **Answer:** C

---

10. Wrapper objects are stored in:
A) Stack
B) Heap
C) Registers
D) Cache only

✅ **Answer:** B

---

11. Which wrapper belongs to `char`?
A) Character
B) Char
C) String
D) Alphabet

✅ **Answer:** A

---

12. `StringBuffer` is:
A) Immutable
B) Mutable
C) Final
D) Static

✅ **Answer:** B

---

13. Which keyword creates String literal?
A) new
B) String
C) No keyword
D) static

✅ **Answer:** C

---

14. Which method compares content of strings?
A) ==
B) equals()
C) compare()
D) hash()

✅ **Answer:** B

---

15. Wrapper classes belong to package:
A) java.util
B) java.lang
C) java.io
D) java.math

✅ **Answer:** B

🟡 MEDIUM LEVEL MCQs (16–30)

16. Integer constant pool range is:
A) 0 to 255
B) −128 to 127
C) −256 to 255
D) Unlimited

✅ **Answer:** B

---

17. Which creates new object always?
A) Integer.valueOf(10)
B) new Integer(10)
C) Autoboxing
D) Constant pool

✅ **Answer:** B

---

18. Which comparison checks reference?
A) equals()
B) compareTo()
C) ==
D) contentEquals()

✅ **Answer:** C

---

19. String concatenation using + creates:
A) Same object
B) New object
C) Thread-safe object
D) No object

✅ **Answer:** B

---

20. Which method converts primitive to String?
A) toString()
B) valueOf()
C) parseInt()
D) charAt()

✅ **Answer:** B

---

21. Which wrapper parsing method converts String to int?
A) Integer.toString()
B) Integer.parseInt()
C) Integer.valueOf()
D) intValue()

✅ **Answer:** B

---

22. Which is mutable but not thread-safe?
A) String
B) StringBuffer
C) StringBuilder
D) Character

✅ **Answer:** C

---

23. Which method returns string length?
A) size()
B) length
C) length()
D) count()

✅ **Answer:** C

---

24. String pool exists in:
A) Stack

B) Heap
C) Method area
D) CPU cache

✅ **Answer:** B

---

25. Which method checks prefix of string?
A) endsWith()
B) startsWith()
C) contains()
D) indexOf()

✅ **Answer:** B

---

26. Wrapper classes are:
A) Abstract
B) Final
C) Static
D) Mutable

✅ **Answer:** B

---

27. Which method converts wrapper to primitive?
A) parse()
B) valueOf()
C) xxxValue()
D) toPrimitive()

✅ **Answer:** C

---

28. Which string method splits string?
A) cut()
B) divide()
C) split()
D) tokenize()

✅ **Answer:** C

---

29. Which string comparison ignores case?
A) equals()
B) compareTo()

C) equalsIgnoreCase()
D) ==

✅ **Answer:** C

---

30. Which creates string outside pool?
A) "Java"
B) String s = "Java"
C) new String("Java")
D) Interned string

✅ **Answer:** C

🔴 HARD LEVEL MCQs (31–45)

31. Which avoids creating new String object?
A) new String("A")
B) String literal
C) Concatenation
D) replace()

✅ **Answer:** B

---

32. Autoboxing happens at:
A) Runtime
B) Compile time
C) Load time
D) Execution

✅ **Answer:** B

---

33. Which is true about StringBuffer?
A) Faster than StringBuilder
B) Thread-safe
C) Immutable
D) Uses string pool

✅ **Answer:** B

---

34. Which wrapper comparison may fail due to cache?
A) equals()

B) compareTo()
C) ==
D) parseInt()

✅ **Answer:** C

---

35. Which operation is costly for String?
A) Reading
B) Comparison
C) Modification
D) Printing

✅ **Answer:** C

---

36. Which method puts String into pool manually?
A) pool()
B) intern()
C) cache()
D) store()

✅ **Answer:** B

---

37. Which is NOT allowed in wrapper class?
A) Static methods
B) Constructors
C) Mutability
D) Final keyword

✅ **Answer:** C

---

38. Which is true?
A) StringBuilder is synchronized
B) StringBuffer is synchronized
C) String is mutable
D) Wrapper is mutable

✅ **Answer:** B

---

39. Which improves performance in loops?
A) String
B) StringBuffer

C) StringBuilder
D) Wrapper

✅ **Answer:** C

---

40. Which method compares lexicographically?
A) equals()
B) ==
C) compareTo()
D) contentEquals()

✅ **Answer:** C

---

41. Wrapper classes are useful in:
A) Generics
B) Collections
C) Streams
D) All

✅ **Answer:** D

---

42. Which conversion causes NumberFormatException?
A) Integer.parseInt("10")
B) Integer.parseInt("10A")
C) valueOf("10")
D) toString(10)

✅ **Answer:** B

---

43. Which string method replaces characters?
A) replace()
B) change()
C) modify()
D) swap()

✅ **Answer:** A

---

44. StringBuilder is preferred when:
A) Thread safety needed
B) Frequent modifications
C) Constants only
D) Shared strings

✅ **Answer:** B

---

45. Which wrapper class supports radix parsing?
A) Double
B) Boolean
C) Integer
D) Character

✅ **Answer:** C

🧩 CODE SNIPPETS (20)

Snippet 1 – Autoboxing

```
Integer i = 10;
```

✅ **Primitive → Object**

Snippet 2 – Unboxing

```
int x = i;
```

✅ **Object → Primitive**

Snippet 3 – Integer cache

```
Integer a = 100;
Integer b = 100;
System.out.println(a == b);
```

✅ **Output:** `true`

Snippet 4 – Outside cache

```
Integer a = 200;
Integer b = 200;
System.out.println(a == b);
```

✅ **Output:** `false`

Snippet 5 – equals()

```
System.out.println(a.equals(b));
```

✅ **Output:** `true`

Snippet 6 – String pool

```
String s1 = "Java";
String s2 = "Java";
System.out.println(s1 == s2);
```

✅ **true**

Snippet 7 – new String

```
String s1 = new String("Java");
String s2 = new String("Java");
System.out.println(s1 == s2);
```

✅ **false**

Snippet 8 – intern()

```
String s = new String("Java").intern();
```

✅ **Moves to pool**

Snippet 9 – String immutability

```
String s = "Hi";
s.concat("Bye");
System.out.println(s);
```

✅ **Output:** `Hi`

Snippet 10 – StringBuilder

```
StringBuilder sb = new StringBuilder("Hi");
sb.append("Bye");
System.out.println(sb);
```

✅ **Output:** `HiBye`

Snippet 11 – StringBuffer

```
StringBuffer sb = new StringBuffer("A");
sb.append("B");
```

✅ **Thread-safe**

Snippet 12 – parseInt

```
int x = Integer.parseInt("123");
```

✅ **Valid**

Snippet 13 – valueOf

```
String s = String.valueOf(10);
```

✅ **"10"**

Snippet 14 – split

```
String[] arr = "A,B,C".split(",");
```

✅ **3 elements**

Snippet 15 – compareTo

```
"abc".compareTo("abd");
```

✅ **Negative value**

Snippet 16 – Wrapper method

```
Integer i = Integer.valueOf("100");
```

✅ **Valid**

Snippet 17 – length

```
System.out.println("Java".length());
```

✅ **4**

Snippet 18 – replace

```
System.out.println("Java".replace('a','o'));
```

✅ **Jovo**

Snippet 19 – StringBuilder in loop

```
for(int i=0;i<3;i++) sb.append(i);
```

✅ **Efficient**

Snippet 20 – Array of wrappers

```
Integer[] arr = {1,2,3};
```

✅ **Autoboxing**

📘 SESSIONS 15 & 16 – EXCEPTION HANDLING (JAVA)

🟢 EASY LEVEL MCQs (1–15)

1. All exceptions in Java are subclasses of:
A) Object
B) Error
C) Throwable
D) RuntimeException

✅ **Answer:** C

2. Which block is always executed?
A) try
B) catch
C) throw
D) finally

✅ **Answer:** D

3. Which package contains exception classes?
A) java.util
B) java.io
C) java.lang
D) java.exception

✅ **Answer:** C

4. Checked exceptions are checked at:
A) Runtime
B) Compile time
C) Load time
D) JVM shutdown

✅ **Answer:** B

5. Unchecked exceptions are subclasses of:
A) Exception
B) Throwable
C) RuntimeException
D) Error

✅ **Answer:** C

---

6. Which is a checked exception?
A) NullPointerException
B) ArithmeticException
C) IOException
D) ArrayIndexOutOfBoundsException

✅ **Answer:** C

---

7. Which keyword is used to explicitly throw exception?
A) throws
B) throw
C) catch
D) finally

✅ **Answer:** B

---

8. Which keyword declares exception to be handled by caller?
A) throw
B) throws
C) try
D) catch

✅ **Answer:** B

---

9. Which exception occurs when dividing by zero?
A) NullPointerException
B) ArithmeticException
C) NumberFormatException
D) IOException

✅ **Answer:** B

---

10. Which block handles exception?
A) try

B) finally
C) catch
D) throw

✅ **Answer:** C

---

11. Errors are:
A) Recoverable
B) Checked
C) Unchecked & serious
D) User-defined

✅ **Answer:** C

---

12. Which is NOT an error?
A) StackOverflowError
B) OutOfMemoryError
C) IOException
D) VirtualMachineError

✅ **Answer:** C

---

13. Can try block exist without catch?
A) Yes
B) No
C) Only with finally
D) Only once

✅ **Answer:** C

---

14. Default exception handler is provided by:
A) Programmer
B) Compiler
C) JVM
D) OS

✅ **Answer:** C

---

15. Which statement is correct?
A) finally always executes
B) catch always executes

C) try always executes
D) throw handles exception

✅ **Answer:** A

🟡 MEDIUM LEVEL MCQs (16–30)

16. Exception propagation means:
A) Exception handled immediately
B) Exception passed to JVM
C) Exception passed to caller method
D) Exception ignored

✅ **Answer:** C

---

17. Which block can have multiple catch?
A) try
B) finally
C) catch
D) throw

✅ **Answer:** A

---

18. Multi-catch block was introduced in:
A) Java 5
B) Java 6
C) Java 7
D) Java 8

✅ **Answer:** C

---

19. Correct multi-catch syntax:
A) catch(A, B e)
B) catch(A | B e)
C) catch(A || B e)
D) catch(A & B e)

✅ **Answer:** B

---

20. Which is correct order?
A) catch → try → finally

B) try → finally → catch
C) try → catch → finally
D) finally → try → catch

✅ **Answer:** C

---

21. Which exception is unchecked?
A) SQLException
B) IOException
C) ClassNotFoundException
D) NullPointerException

✅ **Answer:** D

---

22. Which class is root of checked exceptions?
A) RuntimeException
B) Error
C) Exception
D) Throwable

✅ **Answer:** C

---

23. Can finally block be skipped?
A) Never
B) If exception occurs
C) If System.exit() is called
D) Always

✅ **Answer:** C

---

24. Which is correct about throw?
A) Throws multiple exceptions
B) Used inside method body
C) Used in method signature
D) Handles exception

✅ **Answer:** B

---

25. Which is correct about throws?
A) Used inside catch
B) Used to create exception

C) Used in method declaration
D) Used in finally

✅ **Answer:** C

---

26. Which exception must be handled or declared?
A) RuntimeException
B) Error
C) Checked exception
D) NullPointerException

✅ **Answer:** C

---

27. Which is true about user-defined exception?
A) Must extend Throwable
B) Must extend Exception or RuntimeException
C) Must extend Error
D) Must extend Object

✅ **Answer:** B

---

28. Which block executes even if exception is not thrown?
A) catch
B) throw
C) finally
D) throws

✅ **Answer:** C

---

29. Which keyword is used to rethrow exception?
A) throws
B) throw
C) catch
D) finally

✅ **Answer:** B

---

30. Which is better for handling multiple exceptions with same logic?
A) Multiple catch blocks
B) Nested try
C) Multi-catch
D) finally

✅ **Answer:** C

🔴 HARD LEVEL MCQs (31–45)

31. Which exception is thrown manually by programmer?
A) ArithmeticException
B) IOException
C) User-defined exception
D) JVM exception

✅ **Answer:** C

---

32. Which breaks exception propagation?
A) throws
B) catch
C) throw
D) finally

✅ **Answer:** B

---

33. Which is true about checked exceptions?
A) JVM forces handling
B) Compiler forces handling
C) OS forces handling
D) Runtime forces handling

✅ **Answer:** B

---

34. Which is NOT allowed in multi-catch?
A) Related exceptions
B) Unrelated exceptions
C) Parent & child together
D) Same handling logic

✅ **Answer:** C

---

35. Which exception cannot be caught?
A) Error
B) Exception
C) Throwable
D) All can be caught

✅ **Answer:** D

---

36. Which exception indicates serious JVM problem?
A) Exception
B) RuntimeException
C) Error
D) IOException

✅ **Answer:** C

---

37. Which is correct hierarchy?
A) Object → Throwable → Exception
B) Throwable → Object → Exception
C) Exception → Throwable → Object
D) Object → Exception → Throwable

✅ **Answer:** A

---

38. Which keyword allows passing exception responsibility upward?
A) throw
B) throws
C) try
D) finally

✅ **Answer:** B

---

39. Which is true about custom unchecked exception?
A) Must be handled
B) Must extend Exception
C) Must extend RuntimeException
D) Must extend Error

✅ **Answer:** C

---

40. Which is preferred for business logic violation?
A) Error
B) Checked exception
C) Unchecked exception
D) JVM exception

✅ **Answer:** B

---

41. Which method is called when exception is unhandled?
A) finalize()
B) main()
C) defaultExceptionHandler
D) JVM handler

✅ **Answer:** D

---

42. Which is wrong?
A) finally may not execute
B) throw creates exception object
C) throws declares exception
D) catch handles exception

✅ **Answer:** A

---

43. Which exception can occur without programmer error?
A) ArithmeticException
B) IOException
C) NullPointerException
D) ArrayIndexOutOfBoundsException

✅ **Answer:** B

---

44. Which improves code readability?
A) Nested try-catch
B) Multi-catch
C) Empty catch
D) Catching Throwable

✅ **Answer:** B

---

45. Best practice is to:
A) Catch Exception
B) Catch Throwable
C) Catch specific exception
D) Ignore exception

✅ **Answer:** C

🧩 EXCEPTION HANDLING – CODE SNIPPET MCQs (WITH OPTIONS)

Snippet 1

```
public class Test {
    public static void main(String[] args) {
        System.out.println(10 / 0);
    }
}
```

What will be the output?

A) 0
B) Infinity
C) Compile-time error
D) Runtime exception

✅ **Answer:** D) Runtime exception (`ArithmeticException`)

Snippet 2

```
public class Test {
    public static void main(String[] args) {
        try {
            int x = 10 / 0;
        } catch (ArithmeticException e) {
            System.out.print("A");
        }
        System.out.print("B");
    }
}
```

A) A
B) B
C) AB
D) Compile-time error

✅ **Answer:** C) AB

Snippet 3

```
public class Test {
    public static void main(String[] args) {
        try {
            System.out.print("T");
        } finally {
            System.out.print("F");
        }
    }
}
```

A) T
B) F
C) TF
D) Compile-time error

✅ **Answer:** C) TF

Snippet 4

```
public class Test {
    public static void main(String[] args) {
        try {
            return;
        } finally {
            System.out.print("Finally");
        }
    }
}
```

A) No output
B) Finally
C) Compile-time error
D) Runtime exception

✅ **Answer:** B) Finally

Snippet 5

```
public class Test {
    static void m() {
        int x = 10 / 0;
    }

    public static void main(String[] args) {
        m();
    }
}
```

A) Compile-time error
B) ArithmeticException in m()
C) Exception handled automatically
D) No exception

✅ **Answer:** B) ArithmeticException (exception propagation)

Snippet 6

```
public class Test {
    public static void main(String[] args) {
        try {
            String s = null;
            System.out.println(s.length());
        } catch (NullPointerException e) {
            System.out.print("NPE");
        }
    }
}
```

A) null
B) Compile-time error
C) NPE
D) Runtime crash

✅ **Answer:** C) NPE

Snippet 7

```
public class Test {
    public static void main(String[] args) {
        try {
            int[] a = new int[2];
            System.out.println(a[3]);
        } catch (Exception e) {
            System.out.print("Exception");
        }
    }
}
```

A) 0
B) Compile-time error
C) ArrayIndexOutOfBoundsException
D) Exception

✅ **Answer:** D) Exception

Snippet 8

```
public class Test {
    public static void main(String[] args) {
        try {
            System.out.print("A");
            throw new RuntimeException();
        } catch (Exception e) {
            System.out.print("B");
        } finally {
            System.out.print("C");
        }
    }
}
```

A) A
B) AB
C) ABC
D) BC

✅ **Answer:** C) ABC

Snippet 9

```
public class Test {
    public static void main(String[] args) {
        try {
            System.exit(0);
        } finally {
            System.out.print("Finally");
        }
    }
}
```

A) Finally
B) No output
C) Runtime exception
D) Compile-time error

✅ **Answer:** B) No output

Snippet 10

```
public class Test {
    public static void main(String[] args) {
        try {
            throw new Exception();
        } catch (ArithmeticException e) {
            System.out.print("A");
        } catch (Exception e) {
            System.out.print("B");
        }
    }
}
```

A) A
B) B
C) AB
D) Compile-time error

✅ **Answer:** B) B

Snippet 11

```
public class Test {
    public static void main(String[] args) {
        try {
            int x = 10 / 0;
        } catch (ArithmeticException | NullPointerException e) {
            System.out.print("Handled");
        }
    }
}
```

A) Compile-time error
B) Runtime exception
C) Handled
D) No output

✅ **Answer:** C) Handled

Snippet 12

```
public class Test {
    static void m() throws Exception {
        throw new Exception();
    }

    public static void main(String[] args) {
        m();
    }
}
```

A) Compile-time error
B) Runtime exception
C) Program runs normally
D) Exception handled

✅ **Answer:** A) Compile-time error (checked exception not handled)

Snippet 13

```
class MyException extends RuntimeException {}

public class Test {
    public static void main(String[] args) {
        throw new MyException();
    }
}
```

A) Compile-time error
B) Checked exception
C) Runtime exception
D) No error

✅ **Answer:** C) Runtime exception

Snippet 14

```
public class Test {
    public static void main(String[] args) {
        try {
            return;
        } catch (Exception e) {
            System.out.print("Catch");
        } finally {
            System.out.print("Finally");
        }
    }
}
```

A) Catch
B) Finally
C) CatchFinally
D) No output

✅ **Answer:** B) Finally

Snippet 15

```
public class Test {
    public static void main(String[] args) {
        try {
            throw new Exception();
        } finally {
            System.out.print("F");
        }
    }
}
```

A) F
B) Compile-time error
C) Runtime exception without output
D) Program terminates normally

✅ **Answer:** A) F (then exception propagates)

✅ IMPORTANT NOTE (FOR EXAM)

- **finally runs even with return**
- **finally does NOT run with `System.exit()`**
- **Checked exceptions → must handle or declare**
- **Unchecked exceptions → compiler doesn't force handling**
- **Multi-catch cannot have parent + child together**

📘 SESSIONS 17 & 18 – JAVA I/O, NIO, OBJECT CLASS & UTIL PACKAGE

🟢 EASY LEVEL MCQs (1–15)

1. Which package provides basic I/O operations?
A) java.util
B) java.lang
C) java.io
D) java.nio

✅ **Answer:** C

---

2. InputStream is used for:
A) Character input
B) Byte input
C) Object input
D) File metadata

✅ **Answer:** B

---

3. Reader is used for:
A) Byte data

B) Binary data
C) Character data
D) Object data

✅ **Answer:** C

---

4. Which class is used to write bytes?
A) Reader
B) Writer
C) OutputStream
D) Scanner

✅ **Answer:** C

---

5. Which interface is marker interface for serialization?
A) Cloneable
B) Serializable
C) Runnable
D) Closeable

✅ **Answer:** B

---

6. Serialization means:
A) Object → byte stream
B) Byte stream → object
C) Object → string
D) File → object

✅ **Answer:** A

---

7. Deserialization means:
A) Object → byte stream
B) Byte stream → object
C) File → string
D) Stream → file

✅ **Answer:** B

---

8. Which package is faster and non-blocking?
A) java.io
B) java.util

C) java.nio
D) java.lang

✅ **Answer:** C

---

9. Shallow copy copies:
A) Only primitive fields
B) Only reference addresses
C) Complete object graph
D) Deep nested objects

✅ **Answer:** B

---

10. Deep copy copies:
A) References only
B) Primitive only
C) Complete object graph
D) Static fields only

✅ **Answer:** C

---

11. Date class belongs to:
A) java.sql
B) java.time
C) java.util
D) java.lang

✅ **Answer:** C

---

12. Object class is parent of:
A) Exception
B) String
C) All Java classes
D) Interfaces only

✅ **Answer:** C

---

13. Which method converts object to String?
A) equals()
B) hashCode()
C) toString()
D) valueOf()

✅ **Answer:** C

---

14. equals() method is used to compare:
A) Memory address
B) Hash value
C) Object content
D) Class type

✅ **Answer:** C

---

15. hashCode() is mainly used in:
A) File handling
B) Collections
C) Threads
D) Streams

✅ **Answer:** B

🟡 MEDIUM LEVEL MCQs (16–30)

16. Which is byte-oriented stream?
A) FileReader
B) BufferedReader
C) FileInputStream
D) PrintWriter

✅ **Answer:** C

---

17. Which is character-oriented stream?
A) FileOutputStream
B) FileWriter
C) ByteArrayInputStream
D) ObjectInputStream

✅ **Answer:** B

---

18. Which NIO class represents buffer?
A) Channel
B) Buffer
C) Stream
D) Reader

✅ **Answer:** B

---

19. Which is true about java.nio?
A) Blocking only
B) Uses channels & buffers
C) Slower than IO
D) Only for files

✅ **Answer:** B

---

20. Which keyword prevents serialization?
A) static
B) final
C) transient
D) volatile

✅ **Answer:** C

---

21. Static variables in serialization are:
A) Serialized
B) Not serialized
C) Deep copied
D) Shallow copied

✅ **Answer:** B

---

22. Which method must be overridden for cloning?
A) clone()
B) copy()
C) equals()
D) finalize()

✅ **Answer:** A

---

23. Shallow copy problem occurs due to:
A) Primitive fields
B) Reference fields
C) Static fields
D) Final fields

✅ **Answer:** B

---

24. Which class formats date?
A) Calendar
B) Date
C) SimpleDateFormat
D) Formatter

✅ **Answer:** C

---

25. String → Date conversion uses:
A) parse()
B) format()
C) toString()
D) valueOf()

✅ **Answer:** A

---

26. Date → String conversion uses:
A) parse()
B) format()
C) equals()
D) compareTo()

✅ **Answer:** B

---

27. Calendar class is:
A) Immutable
B) Abstract
C) Final
D) Static

✅ **Answer:** B

---

28. equals() contract requires:
A) Same hashcode always
B) Reflexive, symmetric, transitive
C) Same reference
D) Same memory

✅ **Answer:** B

---

29. hashCode() must be same if:
A) Objects are different
B) equals() is false
C) equals() is true
D) Classes differ

✅ **Answer:** C

---

30. Which collection uses hashCode heavily?
A) ArrayList
B) LinkedList
C) HashMap
D) TreeSet

✅ **Answer:** C

🔴 HARD LEVEL MCQs (31–45)

31. Which stream is used for object serialization?
A) FileWriter
B) ObjectOutputStream
C) DataOutputStream
D) BufferedWriter

✅ **Answer:** B

---

32. Which exception occurs if class not found during deserialization?
A) IOException
B) ClassCastException
C) ClassNotFoundException
D) FileNotFoundException

✅ **Answer:** C

---

33. Deep copy can be achieved using:
A) Assignment
B) clone() shallow
C) Serialization
D) equals()

✅ **Answer:** C

---

34. Which NIO component represents connection?
A) Buffer
B) Channel
C) Stream
D) Reader

✅ **Answer:** B

---

35. java.nio improves performance due to:
A) Garbage collection
B) Blocking calls
C) Non-blocking IO
D) Synchronization

✅ **Answer:** C

---

36. ObjectOutputStream writes:
A) Characters
B) Bytes
C) Objects
D) Files

✅ **Answer:** C

---

37. If equals() is overridden, what else must be overridden?
A) toString()
B) clone()
C) hashCode()
D) finalize()

✅ **Answer:** C

---

38. hashCode() collision means:
A) Objects are equal
B) Objects are same reference
C) Objects may be unequal
D) JVM error

✅ **Answer:** C

---

39. Date class is:
A) Thread-safe

B) Deprecated partially
C) Immutable
D) Abstract

✅ **Answer:** B

---

40. java.time package was introduced in:
A) Java 6
B) Java 7
C) Java 8
D) Java 11

✅ **Answer:** C

---

41. Which is better replacement for Date/Calendar?
A) Timestamp
B) LocalDate
C) SimpleDateFormat
D) Formatter

✅ **Answer:** B

---

42. Object.clone() provides:
A) Deep copy
B) Shallow copy
C) Serialization copy
D) Immutable copy

✅ **Answer:** B

---

43. Which method is inherited from Object?
A) compareTo()
B) format()
C) equals()
D) parse()

✅ **Answer:** C

---

44. Which breaks equals-hashCode contract?
A) Same equals, different hashcode
B) Same hashcode, different equals

C) Same reference
D) Same content

✅ **Answer:** A

---

45. Which method prints object meaningfully?
A) hashCode()
B) equals()
C) toString()
D) clone()

✅ **Answer:** C

🧩 CODE SNIPPET MCQs (WITH OPTIONS)

Snippet 1

```
FileInputStream fis = new FileInputStream("a.txt");
```

A) Character stream
B) Byte stream
C) Object stream
D) NIO stream

✅ **Answer:** B

Snippet 2

```
FileReader fr = new FileReader("a.txt");
```

A) Byte stream
B) Character stream
C) Object stream
D) Buffered stream

✅ **Answer:** B

Snippet 3

```
class A implements Serializable {}
```

A) Compile-time error
B) Object can be serialized
C) Object cannot be serialized
D) Shallow copy

✅ **Answer:** B

Snippet 4

```
class A implements Serializable {
    transient int x = 10;
}
```

A) x serialized
B) x not serialized
C) Compile error
D) Runtime error

✅ **Answer:** B

Snippet 5

```
ObjectOutputStream oos =
 new ObjectOutputStream(new FileOutputStream("a.ser"));
```

A) File writing
B) Object serialization
C) Character writing
D) NIO operation

✅ **Answer:** B

Snippet 6

```
A a1 = new A();
A a2 = a1;
```

A) Deep copy
B) Shallow copy

C) Clone
D) Serialization

✅ **Answer:** B

Snippet 7

```
ByteBuffer buffer = ByteBuffer.allocate(10);
```

A) IO stream
B) NIO buffer
C) File object
D) Channel

✅ **Answer:** B

Snippet 8

```
Date d = new Date();
```

A) java.time
B) java.sql
C) java.util
D) java.lang

✅ **Answer:** C

Snippet 9

```
SimpleDateFormat sdf =
 new SimpleDateFormat("dd-MM-yyyy");
```

A) Date parsing
B) Date formatting
C) Both
D) Invalid

✅ **Answer:** C

Snippet 10

```
Date d = sdf.parse("10-05-2024");
```

A) String → Date
B) Date → String
C) Invalid
D) Compile error

✅ **Answer:** A

Snippet 11

```
String s = sdf.format(d);
```

A) Date → String
B) String → Date
C) Clone
D) Hashing

✅ **Answer:** A

Snippet 12

```
System.out.println(obj.toString());
```

A) Prints hash only
B) Prints meaningful state
C) Compile error
D) Runtime error

✅ **Answer:** B

Snippet 13

```
System.out.println(obj1.equals(obj2));
```

A) Reference comparison
B) Content comparison
C) Hash comparison
D) Class comparison

✅ **Answer:** B

Snippet 14

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

A) equals not needed
B) hashCode not needed
C) equals & hashCode needed
D) Serialization required

✅ **Answer:** C

Snippet 15

```
obj1.hashCode() == obj2.hashCode()
```

A) Objects equal
B) Objects same reference
C) Objects may or may not be equal
D) Compile error

✅ **Answer:** C

📘 JAVA COLLECTIONS FRAMEWORK & STREAM API

🟢 EASY LEVEL MCQs (1–20)

1. Root interface of Collection hierarchy is:
A) List
B) Set
C) Collection
D) Iterable

✅ **Answer:** C

---

2. Which collection allows duplicates?
A) Set
B) Map
C) List
D) Queue

✅ **Answer:** C

---

3. Which List implementation is resizable array?
A) LinkedList
B) Vector
C) ArrayList
D) Stack

✅ **Answer:** C

---

4. Which List preserves insertion order?
A) HashSet
B) TreeSet
C) ArrayList
D) HashMap

✅ **Answer:** C

---

5. Which collection does NOT allow duplicate elements?
A) ArrayList
B) LinkedList
C) HashSet
D) Vector

✅ **Answer:** C

---

6. Which Map allows one null key?
A) Hashtable
B) TreeMap
C) HashMap
D) ConcurrentHashMap

✅ **Answer:** C

---

7. Which Map does NOT allow null key/value?
A) HashMap

B) LinkedHashMap
C) Hashtable
D) TreeMap

✅ **Answer:** C

---

8. Queue follows which principle?
A) LIFO
B) FIFO
C) FILO
D) Random

✅ **Answer:** B

---

9. Which class is synchronized?
A) ArrayList
B) Vector
C) HashSet
D) HashMap

✅ **Answer:** B

---

10. Which interface is used for sorting objects naturally?
A) Comparator
B) Comparable
C) Collections
D) Stream

✅ **Answer:** B

---

11. Which method sorts a list?
A) sort()
B) Collections.sort()
C) List.sort()
D) Arrays.sort()

✅ **Answer:** B

---

12. Which Set maintains insertion order?
A) HashSet
B) TreeSet

C) LinkedHashSet
D) ConcurrentSet

✅ **Answer:** C

---

13. TreeSet stores elements in:
A) Insertion order
B) Random order
C) Sorted order
D) Reverse order only

✅ **Answer:** C

---

14. Which Map maintains insertion order?
A) HashMap
B) TreeMap
C) Hashtable
D) LinkedHashMap

✅ **Answer:** D

---

15. Which collection allows key-value pairs?
A) List
B) Set
C) Map
D) Queue

✅ **Answer:** C

---

16. Which interface represents sequence of elements?
A) Set
B) Map
C) List
D) Queue

✅ **Answer:** C

---

17. Which is legacy class?
A) ArrayList
B) Vector
C) LinkedList
D) HashSet

✅ **Answer:** B

---

18. Which Queue implementation is thread-safe?
A) ArrayDeque
B) PriorityQueue
C) ConcurrentLinkedQueue
D) LinkedList

✅ **Answer:** C

---

19. Which method removes element from queue head?
A) add()
B) remove()
C) push()
D) put()

✅ **Answer:** B

---

20. Which Java version introduced Stream API?
A) Java 6
B) Java 7
C) Java 8
D) Java 11

✅ **Answer:** C

🟡 MEDIUM LEVEL MCQs (21–40)

21. ArrayList internal structure is:
A) Doubly linked list
B) Singly linked list
C) Dynamic array
D) Hash table

✅ **Answer:** C

---

22. LinkedList is better than ArrayList for:
A) Random access
B) Searching
C) Insert/delete in middle
D) Sorting

✅ **Answer:** C

---

23. HashSet internally uses:
A) Array
B) Tree
C) HashMap
D) LinkedList

✅ **Answer:** C

---

24. TreeMap is based on:
A) Hashing
B) Red-Black Tree
C) Array
D) Linked list

✅ **Answer:** B

---

25. Which interface allows custom sorting?
A) Comparable
B) Comparator
C) Collections
D) Stream

✅ **Answer:** B

---

26. Comparator.compare() returns:
A) boolean
B) int
C) long
D) double

✅ **Answer:** B

---

27. Which allows duplicate keys?
A) HashMap
B) TreeMap
C) None
D) LinkedHashMap

✅ **Answer:** C

28. Backed collection means:
A) Independent copy
B) View backed by original collection
C) Immutable collection
D) Concurrent collection

✅ **Answer:** B

29. Which is fail-fast?
A) ConcurrentHashMap
B) HashMap iterator
C) CopyOnWriteArrayList
D) ConcurrentLinkedQueue

✅ **Answer:** B

30. Which is fail-safe?
A) HashMap
B) ArrayList
C) ConcurrentHashMap
D) Vector

✅ **Answer:** C

31. Which method converts collection to stream?
A) map()
B) filter()
C) stream()
D) collect()

✅ **Answer:** C

32. Which stream operation is intermediate?
A) forEach
B) collect
C) map
D) reduce

✅ **Answer:** C

33. Which stream operation is terminal?
A) filter
B) map
C) peek
D) forEach

✅ **Answer:** D

---

34. TreeSet requires elements to be:
A) Serializable
B) Comparable or Comparator
C) Cloneable
D) Immutable

✅ **Answer:** B

---

35. Which collection is best for priority based processing?
A) Stack
B) LinkedList
C) PriorityQueue
D) Vector

✅ **Answer:** C

---

36. ConcurrentHashMap allows:
A) One thread only
B) Multiple threads without locking whole map
C) No iteration
D) Null keys

✅ **Answer:** B

---

37. Which method replaces element in list?
A) add()
B) set()
C) put()
D) replace()

✅ **Answer:** B

---

38. Which collection does NOT extend Collection interface?
A) List

B) Set
C) Queue
D) Map

✅ **Answer:** D

---

39. Which Map sorts keys naturally?
A) HashMap
B) LinkedHashMap
C) TreeMap
D) Hashtable

✅ **Answer:** C

---

40. Stream API supports:
A) External iteration
B) Internal iteration
C) No iteration
D) Manual iteration

✅ **Answer:** B

🔴 HARD LEVEL MCQs (41–60)

41. Which collection allows concurrent modification safely?
A) HashMap
B) ArrayList
C) ConcurrentHashMap
D) Vector

✅ **Answer:** C

---

42. Fail-fast iterator throws:
A) IOException
B) RuntimeException
C) ConcurrentModificationException
D) IllegalStateException

✅ **Answer:** C

---

43. Which stream method reduces elements to single value?
A) map
B) filter
C) reduce
D) collect

✅ **Answer:** C

---

44. Which collection allows null elements multiple times?
A) HashSet
B) TreeSet
C) ArrayList
D) Hashtable

✅ **Answer:** C

---

45. TreeSet insertion complexity is:
A) O(1)
B) O(log n)
C) O(n)
D) $O(n^2)$

✅ **Answer:** B

---

46. Which breaks Comparable contract?
A) compareTo inconsistent with equals
B) equals overridden
C) hashCode overridden
D) Using Comparator

✅ **Answer:** A

---

47. Which stream is infinite?
A) list.stream()
B) Stream.iterate()
C) map.stream()
D) set.stream()

✅ **Answer:** B

---

48. Which collection preserves insertion + allows duplicates?
A) Set

B) Map
C) List
D) TreeSet

✅ **Answer:** C

---

49. Which Map iteration is fastest generally?
A) Hashtable
B) TreeMap
C) HashMap
D) LinkedHashMap

✅ **Answer:** C

---

50. Best choice for thread-safe List?
A) ArrayList
B) Vector
C) CopyOnWriteArrayList
D) LinkedList

✅ **Answer:** C

---

51. Which operation is lazy in streams?
A) filter
B) forEach
C) collect
D) reduce

✅ **Answer:** A

---

52. Which Comparator allows reverse order?
A) Comparator.naturalOrder()
B) Comparator.reverseOrder()
C) Collections.sort()
D) Comparable

✅ **Answer:** B

---

53. Which collection supports blocking operations?
A) Queue
B) Deque

C) BlockingQueue
D) List

✅ **Answer:** C

---

54. Which stream collects result into List?
A) collect(Collectors.toList())
B) map()
C) forEach()
D) reduce()

✅ **Answer:** A

---

55. Which Map allows concurrent read/write without lock?
A) HashMap
B) Hashtable
C) ConcurrentHashMap
D) TreeMap

✅ **Answer:** C

---

56. Which is NOT backed collection?
A) subList()
B) keySet()
C) values()
D) new ArrayList()

✅ **Answer:** D

---

57. Which operation is stateful?
A) map
B) filter
C) distinct
D) forEach

✅ **Answer:** C

---

58. Which collection is best for LIFO?
A) Queue
B) Stack
C) Deque
D) Set

✅ **Answer:** C

---

59. Which stream method short-circuits?
A) anyMatch
B) map
C) filter
D) peek

✅ **Answer:** A

---

60. Which is best design practice?
A) Program to implementation
B) Program to interface
C) Avoid collections
D) Use Vector

✅ **Answer:** B

🧩 CODE SNIPPET MCQs (40)

(Each with 4 options + answer)

Snippet 1

```
List<Integer> list = new ArrayList<>();
list.add(10);
list.add(10);
System.out.println(list.size());
```

A) 0
B) 1
C) 2
D) Compile error

✅ **Answer:** C

Snippet 2

```
Set<Integer> set = new HashSet<>();
```

```
set.add(10);
set.add(10);
System.out.println(set.size());
```

A) 0
B) 1
C) 2
D) Error

✅ **Answer:** B

Snippet 3

```
Map<Integer,String> map = new HashMap<>();
map.put(1,"A");
map.put(1,"B");
System.out.println(map.size());
```

A) 1
B) 2
C) 0
D) Error

✅ **Answer:** A

Snippet 4

```
List<Integer> l = new LinkedList<>();
l.add(1);
l.add(0, 2);
System.out.println(l);
```

A) [1,2]
B) [2,1]
C) [1,0]
D) Error

✅ **Answer:** B

Snippet 5

```
Queue<Integer> q = new PriorityQueue<>();
```

```
q.add(3);
q.add(1);
q.add(2);
System.out.println(q.poll());
```

A) 3
B) 2
C) 1
D) Random

✅ **Answer:** C

Snippet 6

```
List<Integer> list = Arrays.asList(1,2,3);
list.add(4);
```

A) Adds element
B) Runtime exception
C) Compile error
D) No change

✅ **Answer:** B

Snippet 7

```
Set<Integer> set = new TreeSet<>();
set.add(3);
set.add(1);
set.add(2);
System.out.println(set);
```

A) [3,1,2]
B) [1,2,3]
C) Random
D) Error

✅ **Answer:** B

Snippet 8

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

```
Iterator<Integer> it = list.iterator();
list.add(1);
it.next();
```

A) 1
B) null
C) ConcurrentModificationException
D) No output

✅ **Answer:** C

Snippet 9

```
Map<String,Integer> map = new Hashtable<>();
map.put(null,1);
```

A) Allowed
B) NullPointerException
C) Compile error
D) Ignored

✅ **Answer:** B

Snippet 10

```
List<Integer> list = new ArrayList<>();
list.add(3);
list.add(1);
Collections.sort(list);
System.out.println(list);
```

A) [3,1]
B) [1,3]
C) Error
D) Random

✅ **Answer:** B

Snippet 11

```
List<Integer> list = new Vector<>();
list.add(1);
list.add(2);
```

A) Not thread-safe
B) Thread-safe
C) Compile error
D) Deprecated

✅ **Answer:** B

Snippet 12

```
List<Integer> sub = list.subList(0,1);
sub.clear();
```

A) Only sub cleared
B) Parent unchanged
C) Parent modified
D) Error

✅ **Answer:** C

Snippet 13

```
Map<Integer,String> map = new TreeMap<>();
map.put(2,"B");
map.put(1,"A");
System.out.println(map);
```

A) {2=B,1=A}
B) {1=A,2=B}
C) Random
D) Error

✅ **Answer:** B

Snippet 14

```
List<Integer> list = new CopyOnWriteArrayList<>();
for(Integer i : list) {
    list.add(1);
}
```

A) Exception
B) Infinite loop
C) Safe execution
D) Compile error

✅ **Answer:** C

Snippet 15

```
Stream.of(1,2,3).map(x -> x*x).forEach(System.out::print);
```

A) 123
B) 149
C) 149
D) 149

✅ **Answer:** C (prints 149)

Snippet 16

```
List<Integer> list = new ArrayList<>();
list.add(null);
list.add(null);
System.out.println(list.size());
```

A) 0
B) 1
C) 2
D) NullPointerException

✅ **Answer:** C) 2

Snippet 17

```
Set<Integer> set = new HashSet<>();
set.add(null);
set.add(null);
System.out.println(set.size());
```

A) 0
B) 1

C) 2
D) NullPointerException

✅ **Answer:** B) 1

Snippet 18

```
Set<Integer> set = new TreeSet<>();
set.add(null);
```

A) Allowed
B) Ignored
C) Compile-time error
D) NullPointerException

✅ **Answer:** D) NullPointerException

Snippet 19

```
Map<Integer,String> map = new HashMap<>();
map.put(null, "A");
map.put(null, "B");
System.out.println(map);
```

A) {null=A}
B) {null=B}
C) Compile-time error
D) Runtime exception

✅ **Answer:** B) {null=B}

Snippet 20

```
Map<Integer,String> map = new TreeMap<>();
map.put(null, "A");
```

A) Allowed
B) Ignored
C) Compile-time error
D) NullPointerException

✅ **Answer:** D) NullPointerException

Snippet 21

```
List<Integer> list = new ArrayList<>();
list.add(1);
list.add(2);
list.remove(1);
System.out.println(list);
```

A) [1,2]
B) [1]
C) [2]
D) Compile-time error

✅ **Answer:** B) [1]

Snippet 22

```
List<Integer> list = new ArrayList<>();
list.add(1);
list.add(2);
list.remove(Integer.valueOf(1));
System.out.println(list);
```

A) [1]
B) [2]
C) []
D) Error

✅ **Answer:** B) [2]

Snippet 23

```
Queue<Integer> q = new LinkedList<>();
q.add(10);
q.add(20);
System.out.println(q.poll());
```

A) 10
B) 20
C) null
D) Exception

✅ **Answer:** A) 10

Snippet 24

```java
Queue<Integer> q = new LinkedList<>();
System.out.println(q.poll());
```

A) 0
B) null
C) Exception
D) Compile error

✅ **Answer:** B) null

Snippet 25

```java
Queue<Integer> q = new LinkedList<>();
System.out.println(q.remove());
```

A) null
B) 0
C) NoSuchElementException
D) Compile-time error

✅ **Answer:** C) NoSuchElementException

Snippet 26

```java
List<Integer> list = Arrays.asList(1,2,3);
list.set(1, 10);
System.out.println(list);
```

A) [1,2,3]
B) [1,10,3]
C) Runtime exception
D) Compile error

✅ **Answer:** B) [1,10,3]

Snippet 27

```
List<Integer> list = Arrays.asList(1,2,3);
list.remove(1);
```

A) Removes element
B) No change
C) UnsupportedOperationException
D) Compile error

✅ **Answer:** C) UnsupportedOperationException

Snippet 28

```
Map<String,String> map = new ConcurrentHashMap<>();
map.put("A", null);
```

A) Allowed
B) Ignored
C) Compile-time error
D) NullPointerException

✅ **Answer:** D) NullPointerException

Snippet 29

```
Map<String,String> map = new ConcurrentHashMap<>();
map.put(null, "A");
```

A) Allowed
B) Ignored
C) Compile-time error
D) NullPointerException

✅ **Answer:** D) NullPointerException

Snippet 30

```
List<Integer> list = new CopyOnWriteArrayList<>();
for(Integer i : list) {
    list.remove(i);
}
```

A) ConcurrentModificationException
B) Infinite loop
C) Safe execution
D) Compile-time error

✅ **Answer:** C) Safe execution

Snippet 31

```
List<Integer> list = new ArrayList<>();
list.add(1);
list.add(2);
list.forEach(System.out::print);
```

A) 12
B) 21
C) [1,2]
D) Compile error

✅ **Answer:** A) 12

Snippet 32

```
Stream<Integer> s = Stream.of(1,2,3);
System.out.println(s.count());
System.out.println(s.count());
```

A) 3 3
B) 3 Exception
C) 3 0
D) Compile-time error

✅ **Answer:** B) 3 then IllegalStateException

Snippet 33

```
Stream<Integer> s = Stream.of(1,2,3);
long c = s.filter(x -> x > 1).count();
System.out.println(c);
```

A) 1
B) 2

C) 3
D) 0

✅ **Answer:** B) 2

Snippet 34

```
List<Integer> list = Arrays.asList(1,2,3);
list.stream().map(x -> x * 2).forEach(System.out::print);
```

A) 123
B) 246
C) 369
D) Error

✅ **Answer:** B) 246

Snippet 35

```
List<Integer> list = Arrays.asList(1,2,3);
list.stream().filter(x -> x > 5).forEach(System.out::print);
```

A) 1
B) 3
C) 0
D) No output

✅ **Answer:** D) No output

Snippet 36

```
int sum = Stream.of(1,2,3).reduce(0, Integer::sum);
System.out.println(sum);
```

A) 0
B) 3
C) 6
D) Error

✅ **Answer:** C) 6

Snippet 37

```
List<Integer> list = Arrays.asList(1,2,3);
Optional<Integer> max = list.stream().max(Integer::compareTo);
System.out.println(max.get());
```

A) 1
B) 2
C) 3
D) Optional[3]

✅ **Answer:** C) 3

Snippet 38

```
List<Integer> list = new ArrayList<>();
System.out.println(list.stream().findFirst().orElse(10));
```

A) 0
B) null
C) 10
D) Exception

✅ **Answer:** C) 10

Snippet 39

```
List<String> list = Arrays.asList("a","bb","ccc");
list.stream().map(String::length).forEach(System.out::print);
```

A) 123
B) 321
C) abc
D) Error

✅ **Answer:** A) 123

Snippet 40

```
List<Integer> list = Arrays.asList(1,2,3,4);
boolean r = list.stream().anyMatch(x -> x > 3);
System.out.println(r);
```

A) true
B) false
C) Compile-time error
D) Runtime exception

✅ **Answer:** A) true

📘 SESSIONS 27–29 – MULTITHREADING & CONCURRENCY

🟢 EASY LEVEL MCQs (1–20)

1. Multithreading means:
A) Multiple programs
B) Multiple processes
C) Multiple threads in a program
D) Multiple CPUs

✅ **Answer:** C

---

2. Thread in Java can be created by:
A) Extending Thread class
B) Implementing Runnable
C) Using Executor
D) All of the above

✅ **Answer:** D

---

3. Which method starts a thread?
A) run()
B) start()
C) execute()
D) begin()

✅ **Answer:** B

---

4. run() method contains:
A) Thread creation logic
B) Thread scheduling
C) Thread execution code
D) Thread destruction

✅ **Answer:** C

---

5. sleep() method belongs to:
A) Object
B) Runnable
C) Thread
D) ThreadGroup

✅ **Answer:** C

---

6. sleep() method causes thread to:
A) Stop permanently
B) Yield CPU temporarily
C) Die
D) Restart

✅ **Answer:** B

---

7. join() method is used to:
A) Stop thread
B) Pause current thread until another finishes
C) Resume thread
D) Kill thread

✅ **Answer:** B

---

8. yield() method:
A) Stops thread
B) Makes thread sleep
C) Gives chance to other threads
D) Terminates JVM

✅ **Answer:** C

---

9. Default priority of a thread is:
A) 1
B) 5
C) 10
D) 0

✅ **Answer:** B

---

10. Thread priority range is:
A) 0–10
B) 1–10
C) 1–5
D) 0–5

✅ **Answer:** B

---

11. getPriority() returns:
A) Thread name
B) Thread state
C) Thread priority
D) Thread group

✅ **Answer:** C

---

12. setPriority() affects:
A) Scheduling hint
B) Thread speed guarantee
C) Execution order always
D) Deadlock

✅ **Answer:** A

---

13. ThreadGroup is used to:
A) Synchronize threads
B) Group multiple threads
C) Kill JVM
D) Share memory

✅ **Answer:** B

---

14. Which is thread-safe?
A) ArrayList

B) HashMap
C) Vector
D) HashSet

✅ **Answer:** C

---

15. Synchronization is used to:
A) Improve speed
B) Prevent data inconsistency
C) Kill threads
D) Avoid GC

✅ **Answer:** B

---

16. synchronized keyword ensures:
A) Multiple threads execute together
B) Mutual exclusion
C) Faster execution
D) Deadlock

✅ **Answer:** B

---

17. Deadlock occurs when:
A) Thread sleeps
B) Thread waits forever for resources
C) Thread yields
D) Thread terminates

✅ **Answer:** B

---

18. wait() method belongs to:
A) Thread
B) Runnable
C) Object
D) ThreadGroup

✅ **Answer:** C

---

19. notify() wakes up:
A) All waiting threads
B) One waiting thread

C) Current thread
D) JVM

✅ **Answer:** B

---

20. Producer–Consumer problem is based on:
A) Inheritance
B) Polymorphism
C) Inter-thread communication
D) Deadlock

✅ **Answer:** C

🟡 MEDIUM LEVEL MCQs (21–40)

21. start() internally calls:
A) run()
B) sleep()
C) join()
D) yield()

✅ **Answer:** A

---

22. Calling run() directly means:
A) New thread created
B) JVM error
C) Normal method call
D) Thread dies

✅ **Answer:** C

---

23. sleep() throws:
A) RuntimeException
B) IOException
C) InterruptedException
D) ExecutionException

✅ **Answer:** C

---

24. join() throws:
A) IOException

B) InterruptedException
C) RuntimeException
D) Error

✅ **Answer:** B

---

25. Which method releases lock?
A) sleep()
B) yield()
C) wait()
D) join()

✅ **Answer:** C

---

26. Which does NOT release lock?
A) wait()
B) sleep()
C) notify()
D) notifyAll()

✅ **Answer:** B

---

27. synchronized method locks:
A) Method
B) Class
C) Object
D) Thread

✅ **Answer:** C

---

28. synchronized static method locks:
A) Object
B) Thread
C) Class
D) JVM

✅ **Answer:** C

---

29. Deadlock requires:
A) Mutual exclusion
B) Hold and wait

C) Circular wait
D) All of the above

✅ **Answer:** D

---

30. Which avoids deadlock?
A) Nested locks
B) Lock ordering
C) Infinite waiting
D) Yield usage

✅ **Answer:** B

---

31. wait() must be called inside:
A) try block
B) synchronized block
C) static block
D) main method

✅ **Answer:** B

---

32. notifyAll() wakes:
A) One thread
B) JVM
C) All waiting threads
D) Dead threads

✅ **Answer:** C

---

33. Producer–Consumer uses:
A) sleep()
B) yield()
C) wait/notify
D) join()

✅ **Answer:** C

---

34. ThreadGroup can contain:
A) Only threads
B) Only thread groups
C) Threads and thread groups
D) Processes

✅ **Answer:** C

---

35. Thread priority guarantees execution order?
A) Yes
B) No
C) Sometimes
D) JVM error

✅ **Answer:** B

---

36. Which state thread enters after wait()?
A) Running
B) Runnable
C) Waiting
D) Dead

✅ **Answer:** C

---

37. notify() selects thread:
A) Based on priority
B) Randomly
C) FIFO
D) LIFO

✅ **Answer:** B

---

38. Which class supports thread pooling?
A) Thread
B) Runnable
C) ExecutorService
D) ThreadGroup

✅ **Answer:** C

---

39. Which causes race condition?
A) Synchronization
B) Multiple threads on shared data
C) Deadlock
D) ThreadGroup

✅ **Answer:** B

40. Which is better for shared resource?
A) Unsynchronized method
B) synchronized block
C) Multiple objects
D) yield()

✅ **Answer:** B

🔴 HARD LEVEL MCQs (41–60)

41. Which method never releases lock?
A) wait()
B) notify()
C) notifyAll()
D) sleep()

✅ **Answer:** D

42. Calling notify() without wait() causes:
A) Deadlock
B) Lost notification
C) Exception
D) JVM crash

✅ **Answer:** B

43. Which is true about deadlock?
A) JVM detects automatically
B) Recoverable easily
C) Must be avoided by design
D) Occurs at compile time

✅ **Answer:** C

44. Which thread method is static?
A) start()
B) run()
C) sleep()
D) join()

✅ **Answer:** C

45. ThreadGroup priority affects:
A) Only group
B) Child threads max priority
C) JVM
D) Garbage collection

✅ **Answer:** B

46. wait() places thread in:
A) Runnable pool
B) Waiting queue of object
C) ThreadGroup
D) Dead state

✅ **Answer:** B

47. notifyAll() is preferred when:
A) Single thread waiting
B) Multiple conditions exist
C) Deadlock required
D) Performance critical

✅ **Answer:** B

48. Producer–Consumer buffer must be:
A) Static
B) Synchronized
C) Final
D) Immutable

✅ **Answer:** B

49. Which can cause starvation?
A) Fair locking
B) Unfair scheduling
C) notifyAll()
D) yield()

✅ **Answer:** B

50. Which is true?
A) sleep() releases lock
B) wait() releases lock
C) join() releases lock
D) yield() releases lock

✅ **Answer:** B

---

51. Deadlock is possible with:
A) Single lock
B) Multiple locks
C) No locks
D) Atomic variables

✅ **Answer:** B

---

52. Best practice for synchronization:
A) Synchronize whole method
B) Synchronize minimal block
C) Avoid synchronization
D) Use sleep()

✅ **Answer:** B

---

53. Which collection avoids synchronization need?
A) Vector
B) HashMap
C) ConcurrentHashMap
D) ArrayList

✅ **Answer:** C

---

54. Thread safety means:
A) Thread executes faster
B) Correct behavior with multiple threads
C) Deadlock free
D) Single thread only

✅ **Answer:** B

---

55. Which IPC method must be inside synchronized block?
A) sleep()

B) join()
C) wait()
D) yield()

✅ **Answer:** C

---

56. notify() wakes thread from:
A) Any object
B) Same object monitor
C) ThreadGroup
D) JVM

✅ **Answer:** B

---

57. Which thread state after notify()?
A) Running immediately
B) Runnable (waiting for lock)
C) Dead
D) New

✅ **Answer:** B

---

58. Which avoids race condition?
A) Multiple threads
B) sleep()
C) synchronized
D) yield()

✅ **Answer:** C

---

59. Which is NOT a thread lifecycle state?
A) New
B) Runnable
C) Running
D) Terminated

✅ **Answer:** C *(Running is part of Runnable)*

---

60. Best solution for Producer–Consumer:
A) sleep/yield
B) busy waiting

C) wait/notify
D) join

✅ **Answer:** C

🧩 CODE SNIPPET MCQs (20)

Snippet 1

```
Thread t = new Thread();
t.run();
```

A) New thread created
B) Normal method call
C) Compile error
D) Runtime exception

✅ **Answer:** B

Snippet 2

```
Thread t = new Thread();
t.start();
t.start();
```

A) Two threads start
B) No error
C) IllegalThreadStateException
D) Deadlock

✅ **Answer:** C

Snippet 3

```
class A extends Thread {
    public void run() {
        System.out.print("A");
    }
}
new A().start();
```

A) Compile error
B) Prints A
C) No output
D) Runtime error

✅ **Answer:** B

Snippet 4

```
class A implements Runnable {
    public void run() {
        System.out.print("R");
    }
}
new Thread(new A()).start();
```

A) Compile error
B) Prints R
C) No output
D) Runtime error

✅ **Answer:** B

Snippet 5

```
Thread.sleep(1000);
```

A) Compile error
B) Runtime error
C) InterruptedException
D) Works without handling

✅ **Answer:** C

Snippet 6

```
Thread.currentThread().setPriority(11);
```

A) Priority set
B) Compile error

C) IllegalArgumentException
D) No effect

✅ **Answer:** C

Snippet 7

```
Thread t1 = new Thread(() -> {});
Thread t2 = new Thread(() -> {});
t1.join();
```

A) Compile error
B) Runtime error
C) Main waits for t1
D) Deadlock

✅ **Answer:** C

Snippet 8

```
public synchronized void m() {}
```

A) Locks method
B) Locks object
C) Locks class
D) Locks thread

✅ **Answer:** B

Snippet 9

```
public static synchronized void m() {}
```

A) Locks object
B) Locks class
C) Locks JVM
D) Locks thread

✅ **Answer:** B

Snippet 10

```
synchronized(this) {
    count++;
}
```

A) Class level lock
B) Object level lock
C) Thread lock
D) No lock

✅ **Answer:** B

Snippet 11

```
wait();
```

A) Compile error
B) Runtime error
C) IllegalMonitorStateException
D) Works always

✅ **Answer:** C

Snippet 12

```
synchronized(obj) {
    obj.wait();
}
```

A) Compile error
B) Runtime error
C) Valid
D) Deadlock

✅ **Answer:** C

Snippet 13

```
obj.notify();
```

A) Wakes all threads
B) Wakes one waiting thread

C) Releases lock
D) Ends program

✅ **Answer:** B

Snippet 14

```
obj.notifyAll();
```

A) Wakes one thread
B) Wakes all waiting threads
C) Releases lock
D) Causes deadlock

✅ **Answer:** B

Snippet 15

```
class A {
    synchronized void m1() {
        m2();
    }
    synchronized void m2() {}
}
```

A) Deadlock
B) Valid (reentrant lock)
C) Compile error
D) Runtime error

✅ **Answer:** B

Snippet 16

```
ThreadGroup tg = new ThreadGroup("Group1");
```

A) Creates thread
B) Creates thread group
C) Compile error
D) Runtime error

✅ **Answer:** B

Snippet 17

```
System.out.println(Thread.currentThread().getName());
```

A) Prints thread id
B) Prints thread name
C) Prints priority
D) Prints state

✅ **Answer:** B

Snippet 18

```
Thread.yield();
```

A) Stops thread
B) Releases lock
C) Hint to scheduler
D) Kills thread

✅ **Answer:** C

Snippet 19

```
class Test {
    int x = 0;
    synchronized void inc() {
        x++;
    }
}
```

A) Not thread-safe
B) Thread-safe increment
C) Deadlock
D) Compile error

✅ **Answer:** B

Snippet 20

```
Thread t = new Thread(() -> {
    synchronized(this) {}
});
```

A) Compile error
B) Locks Thread object
C) Locks lambda
D) Runtime error

✅ **Answer:** B

📘 SESSION 30 – GENERICS & REFLECTION API (JAVA)

🟢 EASY LEVEL MCQs (1–15)

1. Generics were introduced in Java to:
A) Improve performance
B) Achieve thread safety
C) Provide type safety
D) Reduce memory usage

✅ **Answer:** C

---

2. Generics work at:
A) Runtime
B) Compile time
C) Load time
D) JVM shutdown

✅ **Answer:** B

---

3. Which symbol is used for generics?
A) ()
B) {}
C) <>
D) []

✅ **Answer:** C

---

4. Which is a valid generic type parameter name?
A) int
B) T
C) Integer
D) Object

✅ **Answer:** B

---

5. Generic class example:
A) class Test {}
B) class Test<T> {}
C) class <T> Test {}
D) generic class Test {}

✅ **Answer:** B

---

6. Which collection uses generics?
A) Array
B) Collection
C) Vector
D) All collections

✅ **Answer:** D

---

7. Generic methods are declared using:
A) Class name
B) Method name
C) Type parameter before return type
D) After method name

✅ **Answer:** C

---

8. Which wildcard means “any subtype of Number”?
A) <?>
B) <? super Number>
C) <? extends Number>
D) <Number>

✅ **Answer:** C

9. Which wildcard means “Number or its superclasses”?
A) <? extends Number> B) <? super Number> C) <?>
D) <Number>

✅ **Answer:** B

---

10. Reflection API is used to:
A) Modify source code
B) Inspect classes at runtime
C) Improve performance
D) Create threads

✅ **Answer:** B

---

11. Reflection classes belong to package:
A) java.util
B) java.lang
C) java.lang.reflect
D) java.io

✅ **Answer:** C

---

12. Which class holds metadata of a class?
A) Object
B) Method
C) Field
D) Class

✅ **Answer:** D

---

13. Which method loads class dynamically?
A) newInstance()
B) forName()
C) loadClass()
D) getClass()

✅ **Answer:** B

---

14. Which annotation provides metadata?
A) @Override

B) @FunctionalInterface
C) @Deprecated
D) All of the above

✅ **Answer:** D

---

15. Which operation breaks type safety?
A) Generics
B) Casting
C) Reflection
D) Wildcards

✅ **Answer:** C

🟡 MEDIUM LEVEL MCQs (16–30)

16. Type erasure means:
A) Generics removed at runtime
B) Generics stored in JVM
C) Faster execution
D) Generic object creation

✅ **Answer:** A

---

17. Which generic declaration is invalid?
A) List<Integer>
B) List<int>
C) Map<String,Integer>
D) Set<Double>

✅ **Answer:** B

---

18. Generic type parameters cannot be:
A) Class types
B) Interface types
C) Primitive types
D) Wrapper classes

✅ **Answer:** C

---

19. Which is true about wildcard `<?>`?
A) Only Number allowed
B) Any type allowed
C) Only Object allowed
D) Only primitives allowed

✅ **Answer:** B

---

20. Upper bounded wildcard is used for:
A) Adding elements
B) Reading elements
C) Writing elements
D) Modifying list

✅ **Answer:** B

---

21. Lower bounded wildcard is used for:
A) Reading
B) Writing
C) Sorting
D) Filtering

✅ **Answer:** B

---

22. Which is true about `List<? extends Number>`?
A) Can add Integer
B) Can add Double
C) Cannot add any number
D) Can add null only

✅ **Answer:** D

---

23. Which is true about `List<? super Integer>`?
A) Read Integer safely
B) Add Integer safely
C) Read Number safely
D) Add Double safely

✅ **Answer:** B

---

24. Which reflection class represents method metadata?
A) Field
B) Constructor
C) Method
D) Class

✅ **Answer:** C

---

25. Which reflection API method invokes method dynamically?
A) call()
B) execute()
C) invoke()
D) run()

✅ **Answer:** C

---

26. Which exception may occur in reflection?
A) ClassNotFoundException
B) NoSuchMethodException
C) IllegalAccessException
D) All of the above

✅ **Answer:** D

---

27. Which is true about reflection?
A) Checked at compile time
B) Faster than normal calls
C) Breaks encapsulation
D) Type safe

✅ **Answer:** C

---

28. Which generic constraint restricts type?
A) <?>
B) <T>
C) <T extends Number>
D) <T super Number>

✅ **Answer:** C

---

29. Generic methods are useful when:
A) Class is generic

B) Method needs flexibility
C) Performance required
D) Reflection used

✅ **Answer:** B

---

30. Which is safer than reflection?
A) Casting
B) Generics
C) instanceof
D) newInstance()

✅ **Answer:** B

🔴 HARD LEVEL MCQs (31–45)

31. Why primitives are not allowed in generics?
A) JVM limitation
B) Type erasure
C) Performance
D) Syntax restriction

✅ **Answer:** B

---

32. Which causes heap pollution?
A) Proper generics
B) Raw types
C) Wildcards
D) Type bounds

✅ **Answer:** B

---

33. Which is raw type usage?
A) List<Integer>
B) List<?>
C) List
D) List<Object>

✅ **Answer:** C

---

34. Which is PECS principle?
A) Producer Extends, Consumer Super
B) Process Extends, Call Super
C) Primitive Extends, Class Super
D) Public Extends, Child Super

✅ **Answer:** A

---

35. Which wildcard allows maximum flexibility?
A) <?>
B) <? extends Object>
C) <? super Object>
D) <Object>

✅ **Answer:** A

---

36. Reflection is commonly used in:
A) Games
B) Frameworks (Spring, Hibernate)
C) Simple programs
D) Sorting algorithms

✅ **Answer:** B

---

37. Which reflection method creates object dynamically?
A) new
B) newInstance()
C) getConstructor()
D) invoke()

✅ **Answer:** B

---

38. Which annotation is retained at runtime?
A) @Override
B) @Deprecated
C) @SuppressWarnings
D) Depends on retention

✅ **Answer:** D

---

39. Which reflection API gives all methods?
A) getMethod()

B) getMethods()
C) getDeclaredField()
D) getField()

✅ **Answer:** B

---

40. Which breaks encapsulation most?
A) Inheritance
B) Polymorphism
C) Reflection
D) Overloading

✅ **Answer:** C

---

41. Which generic pattern is unsafe?
A) <T>
B) <T extends Number>
C) Raw types
D) Wildcards

✅ **Answer:** C

---

42. Which reflection call bypasses access control?
A) setAccessible(true)
B) getClass()
C) forName()
D) invoke()

✅ **Answer:** A

---

43. Which combination is invalid?
A) <T extends Number & Runnable>
B) <T extends Runnable & Number>
C) <T extends Number>
D) <T extends Object>

✅ **Answer:** B *(class must come first)*

---

44. Which wildcard is read-only?
A) <? super T> B) <? extends T> C) <T> D) <?>

✅ **Answer:** B

---

45. Best practice for generics:
A) Use raw types
B) Avoid bounds
C) Use bounded types
D) Use reflection

✅ **Answer:** C

🧩 CODE SNIPPET MCQs (15)

Snippet 1

```
List<Integer> list = new ArrayList<>();
list.add(10);
```

A) Compile error
B) Runtime error
C) Valid
D) Warning

✅ **Answer:** C

Snippet 2

```
List list = new ArrayList<Integer>();
list.add("Hello");
```

A) Compile error
B) Runtime error
C) Valid with warning
D) Valid without warning

✅ **Answer:** C

Snippet 3

```
List<int> list = new ArrayList<>();
```

A) Compile error
B) Runtime error
C) Valid
D) Warning

✅ **Answer:** A

Snippet 4

```
static <T> void print(T t) {
    System.out.println(t);
}
```

A) Invalid generic method
B) Valid generic method
C) Runtime error
D) Needs class generic

✅ **Answer:** B

Snippet 5

```
List<? extends Number> list = new ArrayList<Integer>();
list.add(10);
```

A) Adds element
B) Compile error
C) Runtime error
D) Warning

✅ **Answer:** B

Snippet 6

```
List<? super Integer> list = new ArrayList<Number>();
list.add(10);
```

A) Compile error
B) Runtime error
C) Valid
D) Warning

✅ **Answer:** C

Snippet 7

```
Class<?> c = Class.forName("java.lang.String");
```

A) Compile error
B) Runtime error
C) Loads class
D) Creates object

✅ **Answer:** C

Snippet 8

```
Object o = c.newInstance();
```

A) Compile error
B) Runtime error
C) Creates String object
D) Deprecated but works

✅ **Answer:** D

Snippet 9

```
Method m = c.getMethod("length");
```

A) Gets private method
B) Gets public method
C) Compile error
D) Runtime error

✅ **Answer:** B

Snippet 10

```
int len = (int) m.invoke("Java");
```

A) Compile error
B) Runtime error
C) Invokes method
D) Invalid cast

✅ **Answer:** C

Snippet 11

```
Field f = c.getDeclaredField("value");
```

A) Accesses public field
B) Accesses private field metadata
C) Compile error
D) Runtime error

✅ **Answer:** B

Snippet 12

```
f.setAccessible(true);
```

A) Compile error
B) Runtime error
C) Bypasses access control
D) Does nothing

✅ **Answer:** C

Snippet 13

```
class Box<T extends Number> {}
```

A) Invalid bound
B) Valid bounded generic
C) Runtime error
D) Warning

✅ **Answer:** B

Snippet 14

```
List<?> list = new ArrayList<String>();
list.add("Hi");
```

A) Compile error
B) Runtime error
C) Valid
D) Warning

✅ **Answer:** A

Snippet 15

```
@SuppressWarnings("unchecked")
```

A) Runtime annotation
B) Compile-time hint
C) Deprecated
D) Reflection related

✅ **Answer:** B