**Java MCQs: Boxing & Unboxing**

**Q1. What will be the output?**

java

CopyEdit

Integer a = 1000;

Integer b = 1000;

System.out.println(a == b);

* A. true
* B. false
* C. Compilation error
* D. Runtime exception

**Answer: B. false**  
➡️ Because == compares references. Caching in Java works for values **from -128 to 127** only.

**Q2. What will be the result?**

java

CopyEdit

Integer x = 127;

Integer y = 127;

System.out.println(x == y);

* A. true
* B. false
* C. Compilation error
* D. Runtime exception

**Answer: A. true**  
➡️ Values between -128 to 127 are cached (Integer.valueOf()), so x and y point to the same object.

**Q3. What happens during unboxing?**

java

CopyEdit

Integer i = null;

int j = i;

* A. j becomes 0
* B. j becomes null
* C. NullPointerException
* D. Compilation error

**Answer: C. NullPointerException**  
➡️ Unboxing a null wrapper throws NullPointerException.

**Q4. Which of the following involves autoboxing?**

* A. int i = new Integer(5);
* B. Integer i = 5;
* C. Integer i = Integer.valueOf(5);
* D. int i = 5;

**Answer: B. Integer i = 5;**  
➡️ Assigning a primitive int to Integer triggers **autoboxing**.

**Q5. Which of the following is not a wrapper class?**

* A. Integer
* B. Float
* C. Character
* D. String

**Answer: D. String**  
➡️ String is not a wrapper class. It's an immutable object class.

**✅ Java MCQs: Data Types**

**Q6. What is the size of double and float in Java?**

* A. float - 32 bits, double - 64 bits
* B. float - 64 bits, double - 32 bits
* C. Both 32 bits
* D. float - 16 bits, double - 32 bits

**Answer: A. float - 32 bits, double - 64 bits**

**Q7. Which data type is used to store a single 16-bit Unicode character?**

* A. char
* B. byte
* C. int
* D. string

**Answer: A. char**

**Q8. Which of the following are primitive types in Java?**

*(Select all that apply)*

* A. boolean
* B. Integer
* C. double
* D. Long

**Answer: A and C**  
➡️ Integer and Long are wrapper classes. int, long, boolean, double are primitive.

**✅ Java MCQs: Iteration**

**Q9. What will this print?**

java

CopyEdit

for (int i = 0; i < 5; i++) {

System.out.print(i + " ");

}

* A. 1 2 3 4 5
* B. 0 1 2 3 4
* C. 1 2 3 4
* D. Compilation error

**Answer: B. 0 1 2 3 4**

**Q10. Which is the correct syntax for enhanced for loop?**

* A. for (i : list)
* B. for (int i : list)
* C. foreach (int i : list)
* D. loop (int i : list)

**Answer: B. for (int i : list)**  
➡️ Java enhanced for loop: for (Type var : collection)

**Q11. Which of the following allows index access?**

* A. Iterator
* B. for-each loop
* C. for loop
* D. Stream.forEach()

**Answer: C. for loop**

**Q12. What will be the output?**

java

CopyEdit

List<Integer> list = Arrays.asList(1, 2, 3);

for (int i : list) {

i += 1;

}

System.out.println(list);

* A. [2, 3, 4]
* B. [1, 2, 3]
* C. Compilation error
* D. Runtime exception

**Answer: B. [1, 2, 3]**  
➡️ Primitive int i is a **copy**, not reference to list values.

**Q13. What is a valid reason to use Iterator over for-each?**

* A. Better performance
* B. To access index directly
* C. To remove elements during iteration
* D. To loop faster

**Answer: C. To remove elements during iteration**  
➡️ Use Iterator.remove() safely during iteration.

**✅ Want More?**

Would you like:

✅ PDF version of this set?  
✅ 50 more MCQs across these topics?  
✅ A **Mock Quiz UI** (web-based test format)?  
✅ Add **Java 17/21 features** like var, record, etc.?

Let me know!

**You said:**

50 more MCQs across these topics?

**ChatGPT said:**

Updated saved memory

Here are **50 more Java MCQs** on **Boxing/Unboxing, Data Types, and Iteration** with explanations:

**🔁 Section 1: Boxing & Unboxing (17 MCQs)**

**Q1.**

java

CopyEdit

Integer x = new Integer(10);

Integer y = new Integer(10);

System.out.println(x == y);

* A. true
* B. false  
  **Answer:** B. false  
  ➡️ new Integer() creates **new object**, so references differ.

**Q2.**

java

CopyEdit

Integer x = 10;

Integer y = 10;

System.out.println(x == y);

**Answer:** A. true  
➡️ Within **Integer cache range** [-128, 127], so same reference.

**Q3.**

java

CopyEdit

Integer a = 128;

Integer b = 128;

System.out.println(a == b);

**Answer:** B. false  
➡️ Outside cache range, new instances created.

**Q4.**

java

CopyEdit

int a = 5;

Integer b = a;

This is:

* A. Manual boxing
* B. Autoboxing
* C. Unboxing
* D. Invalid  
  **Answer:** B. Autoboxing

**Q5.**

java

CopyEdit

Integer a = null;

int b = a;

**Answer:** C. NullPointerException  
➡️ **Unboxing null** throws NPE.

**Q6.**

java

CopyEdit

Integer x = 5;

Long y = 5L;

System.out.println(x.equals(y));

**Answer:** B. false  
➡️ .equals() checks **type** too.

**Q7.**

Which is true about unboxing?

* A. Converts object to primitive
* B. Only for Strings
* C. Always safe
* D. Prevents NullPointerException  
  **Answer:** A. Converts object to primitive

**Q8.**

Which method is used for **manual boxing**?

* A. Integer.valueOf()
* B. new int()
* C. parseInt()
* D. String.valueOf()  
  **Answer:** A. Integer.valueOf()

**Q9.**

What does this print?

java

CopyEdit

System.out.println(Integer.valueOf("10") + 5);

**Answer:** 15  
➡️ Converts "10" to Integer.

**Q10.**

What happens if you pass a primitive to a method expecting an Object?

* A. Compile error
* B. Autoboxing occurs
* C. Runtime exception
* D. Unboxing occurs  
  **Answer:** B. Autoboxing occurs

**Q11.**

Boolean b = true; is:

* A. Invalid
* B. Autoboxing
* C. Unboxing
* D. Type mismatch  
  **Answer:** B. Autoboxing

**Q12.**

Which of the following is **not** a wrapper class?

* A. Double
* B. Boolean
* C. String
* D. Character  
  **Answer:** C. String

**Q13.**

java

CopyEdit

Long l = 10L;

System.out.println(l == 10);

**Answer:** A. true  
➡️ 10 is **int**, **unboxed** and compared.

**Q14.**

Wrapper classes are located in:

* A. java.util
* B. java.lang
* C. java.box
* D. java.object  
  **Answer:** B. java.lang

**Q15.**

What is the result of:

java

CopyEdit

Integer i = 5;

System.out.println(i + 1);

**Answer:** 6  
➡️ i unboxed to int.

**Q16.**

java

CopyEdit

Double d = Double.valueOf("10.5");

What is d?

* A. 10
* B. 10.5
* C. "10.5"
* D. Error  
  **Answer:** B. 10.5

**Q17.**

Which of these is **not** needed for boxing?

* A. Primitive
* B. Wrapper
* C. Collection
* D. Object reference  
  **Answer:** C. Collection

**🧠 Section 2: Data Types (17 MCQs)**

**Q18.**

Which of the following is **not** a primitive?

* A. byte
* B. short
* C. float
* D. String  
  **Answer:** D. String

**Q19.**

Default value of boolean in a class field?

* A. false
* B. true
* C. null
* D. 0  
  **Answer:** A. false

**Q20.**

Which keyword is used to define constant?

* A. const
* B. final
* C. static
* D. abstract  
  **Answer:** B. final

**Q21.**

Range of byte in Java?

* A. -255 to 255
* B. -128 to 127
* C. 0 to 255
* D. -256 to 255  
  **Answer:** B. -128 to 127

**Q22.**

What is the output?

java

CopyEdit

int x = (int) 3.9;

System.out.println(x);

**Answer:** 3  
➡️ Cast truncates the decimal.

**Q23.**

Which type occupies **least** memory?

* A. byte
* B. short
* C. int
* D. long  
  **Answer:** A. byte

**Q24.**

Which types can hold **decimal values**?

* A. float, double
* B. byte, short
* C. char, boolean
* D. int, long  
  **Answer:** A. float, double

**Q25.**

Which is not valid?

* A. int x = 5L;
* B. float y = 5.0f;
* C. double z = 5;
* D. long l = 10;  
  **Answer:** A. int x = 5L;

**Q26.**

Which of the following has **64 bits**?

* A. long, double
* B. int, float
* C. byte, short
* D. char, boolean  
  **Answer:** A. long, double

**Q27.**

Which keyword is used to indicate **immutable**?

* A. abstract
* B. static
* C. final
* D. volatile  
  **Answer:** C. final

**Q28.**

Which can store **Unicode** character?

* A. int
* B. byte
* C. char
* D. boolean  
  **Answer:** C. char

**Q29.**

Size of int in Java?

**Answer:** 32 bits

**Q30.**

Which is a **reference type**?

* A. int
* B. double
* C. Integer
* D. boolean  
  **Answer:** C. Integer

**Q31.**

Which conversion requires casting?

* A. int → long
* B. long → int
* C. byte → int
* D. float → double  
  **Answer:** B. long → int

**Q32.**

Default value of Object type field?

* A. ""
* B. null
* C. 0
* D. undefined  
  **Answer:** B. null

**Q33.**

What will this print?

java

CopyEdit

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

**Answer:** 66  
➡️ 'A' is 65 (ASCII), so 65+1 = 66.

**Q34.**

Which is **not** part of type promotion rules?

* A. byte → int
* B. short → int
* C. int → float
* D. double → int  
  **Answer:** D. double → int

**🔁 Section 3: Iteration (16 MCQs)**

**Q35.**

Which loop ensures the body runs **at least once**?

* A. for
* B. while
* C. do-while
* D. enhanced for  
  **Answer:** C. do-while

**Q36.**

Which loop uses : syntax?

* A. for
* B. while
* C. enhanced for
* D. do-while  
  **Answer:** C. enhanced for

**Q37.**

What does this print?

java

CopyEdit

int i = 0;

do {

i++;

} while (i < 3);

System.out.println(i);

**Answer:** 3

**Q38.**

Which of the following is not valid?

* A. for (;;)
* B. while(true)
* C. do {} while(true);
* D. for(;;true)  
  **Answer:** D. for(;;true)

**Q39.**

Which loop is best for modifying values **by index**?

* A. while
* B. enhanced for
* C. for loop with index
* D. do-while  
  **Answer:** C. for loop with index

**Q40.**

Which loop structure is safest when modifying a collection during iteration?

* A. for-each
* B. for
* C. Iterator
* D. Stream.forEach  
  **Answer:** C. Iterator

**Q41.**

What is the result?

java

CopyEdit

int[] arr = {1, 2, 3};

for (int i : arr) i++;

System.out.println(Arrays.toString(arr));

**Answer:** [1, 2, 3]  
➡️ i++ does **not** change original array.

**Q42.**

Which of these supports **external iteration**?

* A. forEach
* B. Stream
* C. Iterator
* D. map()  
  **Answer:** C. Iterator

**Q43.**

Which interface is needed for iteration?

* A. Iterable
* B. Runnable
* C. Comparable
* D. Cloneable  
  **Answer:** A. Iterable

**Q44.**

java

CopyEdit

for (int i = 0; i < 0; i++) System.out.println(i);

How many times will it run?

**Answer:** 0

**Q45.**

Can we use break inside a for loop?

**Answer:** Yes

**Q46.**

Can a continue skip an iteration?

**Answer:** Yes

**Q47.**

Which is true?

* A. for-each gives index
* B. for-each works on arrays
* C. for-each allows element removal
* D. for-each doesn't work on collections  
  **Answer:** B. for-each works on arrays

**Q48.**

java

CopyEdit

List<String> list = List.of("A", "B");

for (String s : list) list.remove(s);

**Answer:** ConcurrentModificationException

**Q49.**

Which is better for performance?

* A. for loop
* B. Iterator
* C. forEach
* D. Depends on use case  
  **Answer:** D. Depends on use case

**Q50.**

Which loop supports **multi-variable** initializations?

* A. for
* B. while
* C. do-while
* D. enhanced for  
  **Answer:** A. for

Would you like these 50 MCQs in:

✅ **PDF format**  
✅ **Mock Quiz App UI (web-based)**  
✅ **Printable version**  
✅ Or converted into a **CSV/Excel** file?

Let me know!

**You said:**

about memory , heap ,stack , string pool , PC , metaspace etc

**ChatGPT said:**

Great! Here’s a curated set of **Java MCQs (with answers and explanations)** on memory-related concepts: **Heap**, **Stack**, **String Pool**, **Metaspace**, **Program Counter (PC)**, and **Class Loader Memory**.

**✅ Java Memory Model MCQs (Heap, Stack, String Pool, PC, Metaspace)**

**🔹 Heap Memory MCQs**

**Q1.** What type of data is stored in the Java heap memory?

* A. Primitive variables
* B. Local method variables
* C. Objects and arrays
* D. Class bytecode

**Answer:** C. Objects and arrays  
📝 The **heap** stores all class instances and arrays.

**Q2.** Which memory area is shared among all threads?

* A. Stack
* B. Program Counter
* C. Heap
* D. JVM Registers

**Answer:** C. Heap  
📝 The heap is global and shared across threads.

**Q3.** What happens when the heap is full?

* A. Thread is blocked
* B. JVM expands heap
* C. Garbage Collector is triggered
* D. JVM restarts

**Answer:** C. Garbage Collector is triggered

**Q4.** You can increase heap size with which flag?

* A. -Xms and -Xmx
* B. -Xss
* C. -XX:MetaspaceSize
* D. -XX:NewSize

**Answer:** A. -Xms and -Xmx  
📝 -Xms = initial heap size, -Xmx = max heap size.

**Q5.** The heap is logically divided into:

* A. Stack and heap
* B. Eden, Survivor, Old
* C. Static and dynamic
* D. Object and method

**Answer:** B. Eden, Survivor, Old  
📝 This is part of the **Young Generation (Eden + Survivor)** and **Old Generation**.

**🔹 Stack Memory MCQs**

**Q6.** Stack memory stores:

* A. Objects
* B. Static variables
* C. Method calls and local variables
* D. Heap pointers

**Answer:** C. Method calls and local variables

**Q7.** Stack memory is:

* A. Shared between threads
* B. Thread-local
* C. Used only by JVM
* D. For storing object references

**Answer:** B. Thread-local

**Q8.** Which error is thrown when the stack overflows?

* A. OutOfMemoryError
* B. StackOverflowError
* C. RuntimeException
* D. NoClassDefFoundError

**Answer:** B. StackOverflowError

**Q9.** Which flag is used to increase stack size?

* A. -Xmx
* B. -Xss
* C. -XX:NewSize
* D. -Xms

**Answer:** B. -Xss

**Q10.** Which memory is **faster** in access?

* A. Stack
* B. Heap
* C. Metaspace
* D. String pool

**Answer:** A. Stack  
📝 Stack access is faster due to LIFO structure.

**🔹 String Pool MCQs**

**Q11.** The String pool is stored in:

* A. Heap
* B. Stack
* C. Metaspace
* D. Special interned heap

**Answer:** A. Heap  
📝 But it's a **dedicated area in heap**.

**Q12.** What does String.intern() do?

* A. Converts to uppercase
* B. Returns new String
* C. Places the string into the String pool
* D. Allocates in metaspace

**Answer:** C. Places the string into the String pool

**Q13.** What is the result?

java

CopyEdit

String a = "Java";

String b = new String("Java");

System.out.println(a == b);

* A. true
* B. false

**Answer:** B. false  
📝 One is literal from pool, other is new object.

**Q14.** What is the result?

java

CopyEdit

String a = "Java";

String b = "Java";

System.out.println(a == b);

**Answer:** A. true  
📝 Both point to same interned String from the pool.

**Q15.** What is the output?

java

CopyEdit

String a = new String("Hello").intern();

String b = "Hello";

System.out.println(a == b);

**Answer:** A. true  
📝 Interning makes a point to the pooled "Hello".

**🔹 Metaspace MCQs (Java 8+)**

**Q16.** Where is class metadata stored in Java 8+?

* A. PermGen
* B. Heap
* C. Stack
* D. Metaspace

**Answer:** D. Metaspace

**Q17.** What replaces PermGen in Java 8?

* A. Native Memory
* B. Metaspace
* C. Direct Buffer
* D. Eden Space

**Answer:** B. Metaspace

**Q18.** Metaspace is part of:

* A. Heap
* B. JVM Stack
* C. Native memory
* D. String Pool

**Answer:** C. Native memory  
📝 Metaspace grows in **OS memory**, not in heap.

**Q19.** You can limit metaspace with:

* A. -XX:MaxMetaspaceSize
* B. -Xmx
* C. -Xss
* D. -Xms

**Answer:** A. -XX:MaxMetaspaceSize

**Q20.** If metaspace exceeds limit:

* A. GC clears it
* B. JVM crashes
* C. StackOverFlowError
* D. ClassCastException

**Answer:** B. JVM crashes with OutOfMemoryError: Metaspace

**🔹 Program Counter (PC) Register**

**Q21.** The PC register holds:

* A. Object reference
* B. Address of next JVM instruction
* C. Reference to heap
* D. Local variables

**Answer:** B. Address of next JVM instruction

**Q22.** PC register is:

* A. Shared by all threads
* B. Not used in JVM
* C. Thread-specific
* D. Only for heap allocation

**Answer:** C. Thread-specific

**🔹 Misc JVM Memory Areas**

**Q23.** ClassLoader stores class info in:

* A. Metaspace
* B. Heap
* C. Stack
* D. Code cache

**Answer:** A. Metaspace

**Q24.** Which component handles garbage collection?

* A. Metaspace
* B. GC thread
* C. ClassLoader
* D. Program Counter

**Answer:** B. GC thread

**Q25.** Where are static variables stored?

* A. Heap
* B. Metaspace
* C. Stack
* D. String pool

**Answer:** A. Heap  
📝 Class-level fields are in heap with the class.

Would you like:

✅ 25 more questions (Advanced GC tuning, Native Memory, Memory Leaks)?  
✅ All 50 as **PDF / CSV / printable format**?  
✅ A web-based **mock quiz app**?

Let me know how you'd like to continue!

**You said:**

5 more questions (Advanced GC tuning, Native Memory, Memory Leaks)?

**ChatGPT said:**

Here are **5 advanced MCQs** focused on **GC Tuning**, **Native Memory**, and **Memory Leaks**, suitable for experienced Java developers and system tuners:

**🔧 Q1. Which of the following GC algorithms is designed for low-latency applications in Java 11+?**

* A. Serial GC
* B. Parallel GC
* C. CMS (Concurrent Mark Sweep)
* D. ZGC (Z Garbage Collector)

**Answer:** D. ZGC (Z Garbage Collector)  
📝 ZGC is a scalable, low-latency garbage collector introduced in Java 11.

**🧠 Q2. What tool helps detect native memory leaks in JVM applications?**

* A. JConsole
* B. VisualVM
* C. Native Memory Tracking (NMT)
* D. JFR (Java Flight Recorder)

**Answer:** C. Native Memory Tracking (NMT)  
📝 NMT is a JVM feature to monitor native memory (non-heap) usage.

**🧼 Q3. Which JVM flag logs detailed GC activity for performance tuning?**

* A. -verbose:class
* B. -Xss1024k
* C. -Xlog:gc\* or -XX:+PrintGCDetails
* D. -XX:+UseG1GC

**Answer:** C. -Xlog:gc\* or -XX:+PrintGCDetails  
📝 These flags provide detailed GC logs helpful for tuning.

**🕳️ Q4. What is a typical cause of a Java memory leak despite GC being available?**

* A. Using final variables
* B. Unclosed JDBC connections
* C. Use of G1 GC
* D. Frequent String allocations

**Answer:** B. Unclosed JDBC connections  
📝 Resources like connections, streams, and listeners not being released cause memory leaks.

**📦 Q5. Which Java option helps in analyzing heap dumps after an OutOfMemoryError?**

* A. -XX:+HeapDumpOnOutOfMemoryError
* B. -Xloggc
* C. -XX:+UseStringDeduplication
* D. -XX:MaxPermSize

**Answer:** A. -XX:+HeapDumpOnOutOfMemoryError  
📝 This generates a .hprof file for deep analysis using tools like Eclipse MAT.