Question	Option1	Option2	Option3	Option4	Right Answer
Which is faster for iteration?	Iterator	ListIterator	Enumeration	Stream	Option3
Which method checks if a key exists in a Map?	containsKey()	findKey()	hasKey()	searchKey()	Option1
Which traversal is allowed by ListIterator but not Iterator?	Forward only	Backward	Skipping	Circular	Option2
A Map cannot contain:	Duplicate keys	Duplicate values	Null values	String keys	Option1
Vector increases size by default:	+1 element	Doubles	+10 elements	+2 elements	Option2
Which supports random access?	LinkedList	Queue	ArrayList	HashSet	Option3
Java 8 added Streams mainly for:	Parallel execution	Serialization	Collections sorting	File handling	Option1
Which method returns sorted stream?	order()	arrange()	sorted()	sort()	Option3
Comparator can be created using:_	Anonymous class	Lambda expressions	Method reference	All of these	Option4
Which cannot be caught?	Exception	RuntimeException	Error	IOException	Option3
Which keyword is used to explicitly throw exception?	throws	throw	final	error	Option2
Which exception is checked?	NullPointerException	ArithmeticException	IOException	ArrayIndexOutOfBoundsException	Option3
Throwable has methods:	printStackTrace()	getMessage()	getCause()	All	Option4
What happens if exception not caught?	Program continues	JVM terminates program	OS handles	Exception ignored	Option2
throws keyword is used in:	Method declaration	Method call	Class declaration	Package	Option1
Finally block executes even when:	return in try	break	continue	All	Option4
Which of the following is true about static nested classes in Java?	Can access instance variables of outer class directly	Cannot access instance variables of outer class directly	Cannot have static members	Must extend the outer class	Option2
How do you instantiate a static nested class?	OuterClass.Inner obj = new Inner();	OuterClass.Inner obj = new OuterClass.Inner();	Inner obj = new OuterClass.Inner();	Inner obj = new Inner();	Option2
True or False: Static nested classes can have static methods.	TRUE	FALSE	Only if outer class is static		Option2
Difference between static nested class and non-static inner class:	Static nested class cannot access outer instance variables directly	Inner class cannot have constructors	Static nested class cannot implement interfaces	Inner class cannot be private	Option1

Question	Option1	Option2	Option3	Option4	Right Answer
Can a static nested class throw exceptions from its constructor?	Yes	Only unchecked exceptions	Only checked exceptions		Option1
How do you call a method of a static nested class without creating an instance of the outer class?	Using outer object	Using nested object	Using OuterClass.NestedClass.methodName()	Not possible	Option3
Can a static nested class have a constructor?	Yes	No	Only default constructor	Only if outer class is final	Option1
Which of these is not allowed in a static nested class?	Static methods	Instance methods	Direct access to outer instance variables ☑	Implementing interfaces	Option3
Which of the following can be used as nested classes?	Only static classes	Only non-static classes	Both static and non-static ☑	Only abstract classes	Option3
Which is correct syntax for an anonymous class implementing Runnable?	Runnable r = new Runnable() { public void run() {} };	Runnable r = new Runnable();	Runnable r = Runnable() { public void run() {} };	Runnable r = new Runnable.run() {};	Option1
Can an anonymous class access local variables of the enclosing method?	Yes, all local variables	No local variables	Only final or effectively final variables	Only static variables	Option1
True or False: Anonymous classes can implement multiple interfaces.	TRUE	FALSE	Only if outer class is abstract	Only if inner class is static	Option2
Which of the following is a limitation of anonymous classes?	Cannot have static members	Cannot access final variables	Cannot implement interfaces	Cannot override methods	Option1
How can you instantiate a local inner class?	new LocalClass(); inside method	OuterClass.LocalClass obj = new OuterClass.LocalClass();	new OuterClass().LocalClass();	Outside the method directly	Option1
Lambda expression in Java is used to:	Create objects	Implement functional interfaces	Replace static classes	Replace main method	Option2
Syntax of lambda with one parameter x:	(x) -> x*x ☑	x -> (x*x)	x => x*x	(x) => x*x	Option1
True or False: Lambda expressions can only be used with functional interfaces.	TRUE	FALSE	Only with Runnable	Only with Comparator	Option1
Lambda expressions with multiple statements must use:	Braces {}	Semicolon only	Parentheses ()	Nothing	Option1

Question	Option1	Option2	Option3	Option4	Right Answer
True or False: Lambda expressions can capture this reference.	TRUE	FALSE	Only in static context	Only in nested class	Option1
Lambda expression for calculating square of number x:	x -> x*x	(x) -> return x*x;	(int x) { return x*x; }	x => x*x	Option1
Type inference in lambda expressions allows:	Omitting parameter types	Omitting return type	Omitting method body	Omitting parentheses	Option1
Lambda expression is equivalent to:	Static nested class	Anonymous class implementing single abstract method	Local inner class	Top-level class	Option2
Lambda expressions can be assigned to:	Variable of functional interface type	Any variable	Only Runnable	Only static classes	Option1
Which of these is correct lambda syntax for comparator?	(a, b) -> a.length() - b.length() ☑	(a, b) => a.length() - b.length()	(a, b) -> { return a.length() - b.length() }	Both 1 and 3	Option4
A functional interface is defined as:	Interface with multiple abstract methods	Interface with exactly one abstract method	Interface with no abstract methods	Interface with static methods only	Option2
Which annotation ensures an interface is a functional interface?	@Functional	@Interface	@FunctionalInterface	@SingleMethod	Option3
True or False: A functional interface can have default methods.	TRUE	FALSE	Only static methods allowed	Only abstract methods allowed	Option1
Can a functional interface have static methods?	Yes	No	Only default methods	Only if outer class is static	Option1
Which of these is a built-in functional interface in Java?	Runnable	Comparator	Serializable	Both 1 and 2	Option4
True or False: Lambda expressions can implement functional interfaces without explicitly writing a class.	Ture	FALSE	Only Static Classes	Only anonymous classes	Option1
True or False: Streams are lazy, i.e., they don’t execute until terminal operation.	TRUE	FALSE			Option1
Which method converts a Collection to a Stream?	collection.stream()	Stream.of(collection)	Stream.create(collection)	collection.toStream()	Option1
Terminal operation to convert Stream to List:	toList()	collect(Collectors.toList())	collectList()	Both 1 and 2	Option1
Which of these is not an intermediate operation?	distinct()	limit()	sorted()	forEach()	Option4