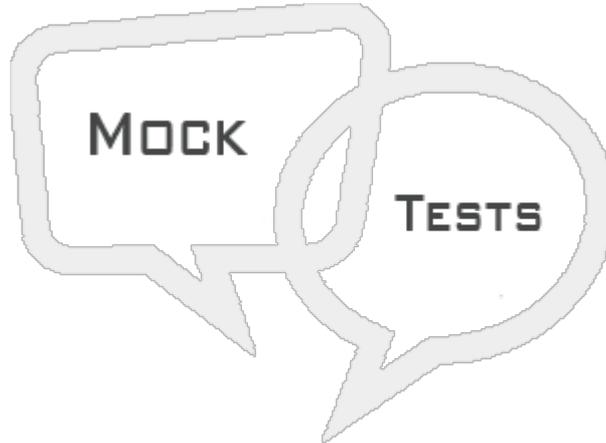


# JAVA 8 MOCK TEST

This section presents you various set of Mock Tests related to **Java 8**. You can download these sample mock tests at your local machine and solve offline at your convenience. Every mock test is supplied with a mock test key to let you verify the final score and grade yourself.



## JAVA 8 MOCK TEST I

**Q 1 - Which of the following is correct about Java8?**

- A - Lambda expression - Adds functional processing capability to Java.
- B - Method references - Referencing functions by their names instead of invoking them directly. Using functions as parameter.
- C - Default method - Interface to have default method implementation.
- D - All of the above.

**Q 2 - Which of the following is correct about Java8?**

- A - New compiler tools and utilities are added like 'jdeps' to figure out dependencies.
- B - New stream API to facilitate pipeline processing is added.
- C - New Improved date time API added.
- D - All of the above.

**Q 3 - Which of the following is correct about Java 8 lambda expression?**

- A - Optional type declaration - No need to declare the type of a parameter. The compiler can inference the same from the value of the parameter.
- B - Optional parenthesis around parameter - No need to declare a single parameter in parenthesis. For multiple parameters, parentheses are required.
- C - Both of the above.
- D - None of the above.

**Q 4 - Which of the following is correct about Java 8 lambda expression?**

- A - Optional curly braces - No need to use curly braces in expression body if the body contains a single statement.
- B - Optional return keyword - The compiler automatically returns the value if the body has a single expression to return the value.
- C - Both of the above.
- D - None of the above.

**Q 5 - Which of the following is correct about Java 8 lambda expression?**

- A - Lambda expressions are used primarily to define inline implementation of a functional interface.
- B - Lambda expression eliminates the need of anonymous class and gives a very simple yet powerful functional programming capability to Java.
- C - Both of the above.
- D - None of the above.

**Q 6 - Which of the following is correct about Java 8 lambda expression?**

- A - Using lambda expression, you can refer to final variable or effectively final variable (which is assigned only once).
- B - Lambda expression throws a compilation error, if a variable is assigned a value the second time.
- C - Both of the above.
- D - None of the above.

**Q 7 - Which of the following is correct about Java 8 method references?**

- A - Method references help to point to methods by their names.
- B - A method reference is described using :: (double colon) symbol.
- C - Both of the above.
- D - None of the above.

**Q 8 - Which of the following is correct about System.out::println expression?**

- A - System.out::println method is a static method reference to println method of out object of System class.
- B - System.out::println method is a instance method reference to println method of out object of System class.
- C - Both of the above.

D - None of the above.

**Q 9 - Which of the following functional interface represents an operation that accepts two input arguments, and returns no result?**

A - BiConsumer<T,U>

B - BiFunction<T,U,R>

C - BinaryOperator<T>

D - BiPredicate<T,U>

**Q 10 - Which of the following functional interface represents an operation that accepts two input arguments, and returns a result?**

A - BiConsumer<T,U>

B - BiFunction<T,U,R>

C - BinaryOperator<T>

D - BiPredicate<T,U>

**Q 11 - Which of the following functional interface represents an operation upon two operands of the same type, producing a result of the same type as the operands?**

A - BiConsumer<T,U>

B - BiFunction<T,U,R>

C - BinaryOperator<T>

D - BiPredicate<T,U>

**Q 12 - Which of the following functional interface represents a predicate (Boolean-valued function) of two arguments?**

A - BiConsumer<T,U>

B - BiFunction<T,U,R>

C - BinaryOperator<T>

D - BiPredicate<T,U>

## ANSWER SHEET

Question Number	Answer Key
-----------------	------------

1	D
2	D
3	C
4	C
5	C
6	C
7	C
8	A
9	A
10	B
11	C
12	D