

Oracle.1Z0-819.v2022-02-08.q102

□□□□:	1Z0-819
□□□□:	Java SE 11 Developer
□□□:	Oracle
□□ □□ □□□:	102
□□:	v2022-02-08
# □□ □:	1523
# □□ □□□:	1020
https://www.krdump.com/Oracle.1Z0-819.v2022-02-08.q102.html	

NEW QUESTION: 1

□□□ □□ □□:

11□□ □□□ □ □□ □□?

- A. UnaryOperator<Interger> uo = (var x) -> (x * 3);
- B. UnaryOperator<Interger >uo = var x -> { return x 3 ; };
- C. UnaryOperator<Interger >uo = x -> { return x * 3; };
- D. UnaryOperator<Interger>uo = (int x) -> x * 3;

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 2

□□□:

□□ □ □□□ □□□□□ 1 □□□□ □□□ □ □□□□?

- A. □□ aba = v, w -> x * y;
- B. □□ aba = (a, b) -> a * b;
- C. □□ aba = (int i, j) -> (return i * j; };
- D. □□ aba = (int e, int f) -> { return e * f; };
- E. □□ aba = (int m, int n) -> { m * n };

Answer: B,C ([LEAVE A REPLY](#))

NEW QUESTION: 3

□□□:

1□□ □□□ □ 1.17□ □□□ □□□□ □□□□□?

- A. float z = (float)(Math.round((float)x/y*100)/100);
- B. float z = Math.round((int)(x/y),2);
- C. float z = Math.round((float)x/y,2);
- D. float z = Math.round((float)x/y*100)/(float)100;

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 4

□□□:

□ □□□□ □□:

□□ □□ □□ □ □

□ □□□□ □□□ □□□□□?

A. 1) □□

B. □□□□

C. □□□□ □□□□□.

D. java.lang.ArrayIndexOutOfBoundsException □ □□□□□□.

E. 1) □□2) □3) □

Answer: E ([LEAVE A REPLY](#))

NEW QUESTION: 5

□□□:

□□□ □□□□□?

A. □□□□ ClassCastException □ □□□□□.

B. AnotherClass#methodA()AnotherClass#methodA()

C. □□□□ □□□□□□.

D. SomeClass#methodA()AnotherClass#methodA()

E. AnotherClass#methodA()SomeClass#methodA()

F. SomeClass#methodA()SomeClass#methodA()

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 6

□□□:

□□□.□□

□□□.□□

□□□ 4□ □□□□□ □□□ □□ □□□?

A. 3□□ □□ □□□□□ □□□□□ □□□□□.

B. 2□□ @Override □□□ □□□□□.

C. 2□□ □□□ Car ob = new Car()□ □□□□□.

D. 1□□□ □□ □□□□ □□□□□.

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 7

Java SE □□□ API□ □□□ □ □□□ □□□□□? (2□□ □□□□□.)

A. java.logging

B. jdk.jartool

C. jdk.httpserver

D. java.desktop

E. javafx

Answer: A,C ([LEAVE A REPLY](#))

NEW QUESTION: 8

□□□:

□□<□□□> □□1 = □□□ ArrayList<>();

list1.add("□");

list1.add("B");

□□ list2 = List.copyOf(list1);

list2.add("C");

□□<□□<□□□>> □□3 = □□.of(□□1, □□2);

System.out.println(list3);

□□□ □□□□□?

A. [[A, B],[A, B]]

B. □□□□ □□□ □□□□□.

C. [[A, B], [A, B, C]]

D. [[A, B, C], [A, B, C]]

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 9

□□□ □□ □□:

□□□ □□□□□?

A. 13□□□ □□□□ □□□□□.

B. 10

C. 1

D. 16□□□ □□□□ □□□□□.

E. □□□□ □□□ □□

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 10

□□□ □□ □□:

□□□ □□□□□?

A. abc def

B. □□□ cf

C. □□□□ □□□□□□.

D. □□□□ ArrayIndexOutOfBoundsException□ □□□□□.

E. ab cd ef

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 11

□□□:

Fox □□□□ □□ □□□□ □□ □□?

- A. Forest□ Town□ □□ □□□ □□□ □□□□ □□ Fox □□□□ inhabit □□□□ □□ □□ □□□ □□□□.
- B. Fox□ □□□□ □ □□ □□□□□□ inhabit □□□ □□□ □□□□□.
- C. Fox □□□□ □□□ □□□□ □□ □ inhabit □□□□ □□□□ □□□ □□□□.
- D. Fox □□□□ Forest □□ Town □□□□□□ □□□□ □□□ □ □ □□□□□ □ □ □□.
- E. Fox □□□□ inhabit □□□□ □□ □□□ □□□□ □□□.

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 12

□□ □□ = List.of(0,1,2,3,4,5,6,7,8,9);

□□□ □□□ □□□□□ □□□. □□ □ □□ □□□ □□□ □□□ □□□□? (2□□ □ □□□□.)

- A. □□ □□ = number.stream().parallel().averagingDouble(a -> a);
- B. □□ □□ = number.parallelStream().mapToInt(m -> m).average().getAsDouble();
- C. □□ □□ = number.stream().mapToInt (i -> i).average().parallel();
- D. □□ □□ = □□.stream().average().getAsDouble();
- E. □□ □□ = number.stream().collect(Collectors.averagingDouble(n -> n));

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 13

□□□:

String originalPath = "data\\projects\\a-project\\..\\..\\another-project"; □□ □□ = Paths.get(originalPath); System.out.print(□□.□□□()); □□□ □□□□□?

- A. data\\another-project
- B. data\\projects\\a-project\\another-project
- C. data\\projects\\a-project\\..\\..\\another-project
- D. data\\projects\\a-project\\..\\..\\another-project

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 14

□□□ TripleThis.java:

TripleThis.java□ □□□□□ □□□ □□ □□□□ □□□ □□□□□.

□□: TripleThis.java□ □□□□ □□□ □□□□ □□ □□□ □□□□□.

□ □□□□ □□□ □□□□ □ □□ □□ □□□ □□□□□?

- A. 12□□ public static void printValue function<Integer> f, int num) {□ □□□□□.
- B. 12□□ public static int printValue function<Integer, Integer>, f, T num {□□ □□□□□.

- C. 9□□ function<Integer>□ □□, Integer> = X -> { return (integer) x * 3; }.
- D. 9□□ function<Integer>□ □□ tripler = x-> - { return (Integer) X * 3 ; }.
- E. 12□□ public static <T> void printValue (Function<T, T> f, T num) {,

Answer: B,D ([LEAVE A REPLY](#))

NEW QUESTION: 15

□□□:

□□ □□ □□□□?

- A. □□□□ IllegalStateException□ □□□□□.
- B. 3□□ □□□□ □□□□□.
- C. □□□□ □□□□□□.
- D. 4□□ □□□□ □□□□□.

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 16

□□□:

□ □□□ □□□ □□□ □□□□□□.

□□□□ □□□□ □□ 1□□ □□ □□□ □□□□ □□□?

- A. □□□ □□□ = msg -> { return System.out.print(msg); };
- B. □□□ □□□ = var arg > {System.out.print(arg);};
- C. □□□ □□□ = (□□□ □□) > System.out.print(args);
- D. □□□ □□□ = System.out::print;

Answer: D ([LEAVE A REPLY](#))

1Z0-819 □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ 1Z0-819 □□!
 DumpTop □ □□ **1Z0-819** □□ □□□ □□□□□□, DumpTop 1Z0-819 □□ □□□
 □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop
 1Z0-819 □□□ □□□□□. <https://www.dumptop.com/Oracle/1Z0-819-dump.html> (297
 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 17

□□□:

□□□ □□□□□?

- A. 2,34,34,5
- B. 2,34,54,5
- C. 2,54,54,5
- D. 2,34,54,3

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 18

□□□ □□ □□:

□□ "for" □□□ □□□ □□□ □□□□□?

A. □□ A

B. □□ D

C. □□ B

D. □□ C

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 19

□□□:

1□□ □□ □□□□ □□ □□?

A. □□ □□□ □□□□ NoSuchElementException □ □□□□□.

B. □□ System.out::print □□ □□□□□.

C. □□ □□□ □□□□ NullPointerException □ □□□□□.

D. □□ □□□ □□ □□□ □□□□ □□□□.

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 20

□□□:

□□□ □□ □ Stream □□□ □□□□ □□□. 1□□ □□□ □□ □ □ □□□ □□□□
□□□ □□□□□?

A. var lines = Files.lines(Paths.get(INPUT_FILE_NAME));

B. Stream<String> □ = Files.lines(INPUT_FILE_NAME);

C. var lines = Files.readAllLines(Paths.get(INPUT_FILE_NAME));

D. □□□ □□ = Files.readAllLines(Paths.get(INPUT_FILE_NAME));

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 21

□□□□ □□□□ Java□ □□□ □□□ □□?

A. Java□ □□ □□□□ □□□ □□□□□□ □□□□□□.

B. Java□□ □□□□□ □□□ □□□ □□□□□ □ □□□ □ □□ □□ □□ □□□□
□□ □□□□.

C. Java□ □□□ □□□ □□□□□□.

D. Java□ □□ □□□ □□□□ □□□□□□.

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 22

□□□:

Main □□□□ □□□□□ □□□ □ □□ □□□□ □□ □□□ □□□□□? (2□□ □□ □□□.)

- A. □□ Student □□□ □□□ □□□ Java □□□ Student.java□ □□□□□.
- B. 2□□ public Student(String classname)□ □□□□□.
- C. 1□□ public class Student {□ □□□□□.
- D. 3□□ Student student = new Student("Biology");□ □□□□□.
- E. 1□□ □□ □□□ Student {□ □□□□□.

Answer: B,D ([LEAVE A REPLY](#))

NEW QUESTION: 23

□□□:
□□□ □□□□□?

- A. 1□□□?
- B. □□□□ IndexOutOfBoundsException□ □□□□□.
- C. 0□□□?
- D. 2□□□
- E. 3□□□

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 24

□□□:
Foo □□□ □□□□ □□□□ □□□ □□ □□□ □□ □□□□□?

- A. □□ □□□ □□□ □□□ ReentrantLock(true)□□ □□□□□.
- B. □□ □□□ □□□□ □□□□.
- C. □□□ □□□ □□□□.
- D. foo □□□ □□□ □□ □□□ □□□□□.

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 25

□□□:
p instanceof Pair□□ p.isValid()□ true□ □□□□ □□□.
□ □□ □□□ □□□□□ □□ □□ □□ □□□ □□ □□□ □□□□□?

- A. isValid□ □□□□□ □□□.
- B. setLeft □ setRight□ □□□□□ □□□.
- C. □□, □□□, setLeft □ setRight□ □□□□□ □□□.
- D. □□□ □□□□ □□□□□ □□□.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 26

□□□ □ □□ □□□□ □□□□ □ □□□□ □ □□ □□□ □□□□□? (2□□ □□□ □□.)

- A. jmod □□
- B. □□ --show-module-resolution
- C. jar --show-module-resolution
- D. □□ Hello.java
- E. jdeps --list-deps

Answer: B,E ([LEAVE A REPLY](#))

NEW QUESTION: 27

□□ □□□ □□ □□□ □□□□□□?

@□□□□□ □□□ {

- A. □□□ □□() □□□ "";
- □□();

}

@interface □□□□ □□□ □□ {□ □□□□□.

- B. □□□ □□() □□ null;
- □□();

}

@□□□□□ □□□ {

- C. □□□ □□();
- □□ □□□ "";

}

- D. □□□ □□();
- □□;

}

@□□□□□ □□□ {

- E. □□□ □□() □□□ "";
- □□();

}

@□□□□□ □□□ {

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 28

□□□:

□□□ □□□□□?

- A. Hello World! Hello World!
- B. Hello world!□□□□□!
- C. Hello world! Hello world!
- D. Hello world! Hello world!

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 29

□□□:

□□□ □□□□□?

A. 0

B. □□□□

C. 0 4 9

D. 10

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 30

□□□:

□□□ □□□□□?

A. □□□ □□

B. □□□□ □□□□□□.

C. □□□ □□ □□ □□ □□ □□□□ □□□□ □□□

D. □□□□□ □□ □□ □□□□ □□□□.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 31

□□□:

□ □□ □□:

□□□ □□□□□?

A. 9001: java.io.FileNotFoundException: MyFile.txt-MyFile.txt

B. 9001: java.io.FileNotFoundException: MyFile.txt-MyFile.txt

9001: □□ □□□□ □□-9001-MyFile. txt

C. 9001: □□ □□□□ □□-9001-MyFile. txt

D. 1□□□ □□□□ □□□□□.

Answer: D ([LEAVE A REPLY](#))

1Z0-819 □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ 1Z0-819 □□!
DumpTop □ □□ **1Z0-819** □□ □□□ □□□□□□, DumpTop 1Z0-819 □□ □□□
□□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop
1Z0-819 □□□ □□□□□. <https://www.dumptop.com/Oracle/1Z0-819-dump.html> (297
Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 32

□□□ □□ □□:

Computatorclass□ □□□□ □ □□ □□□ □□□□□?

- A. 3□□ Double sum = 0.0□□ □□□□□.
- B. 5□□ List<Double> □□□ □□ = List.of(5, 4, 6, 3, 7, 2, 8, 1, 9);
- C. 2□□ public Double sum(C □□□)□□ □□ {
- D. 1□□ □□□□ NumberFormatException throw □□
- E. 4□□ (Double n: □□□) {□ □□□□□.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 33

□□□:

□ □□□□ □□:

□□ □□ □□ □ □

□□□ □□□□□?

- A. □□ □□ □□:0)□ □□□□□. □□1). □1). □...
- B. java.lang.NullPointerException□ □□□□□□.
- C. 0). □□1). □2). □
- D. □□□□ □□□□□.
- E. 0). □□

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 34

□□□:

□□□ □□□□□?

- A. 3
- B. 5
- C. 23
- D. 11
- E. 25

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 35

□□□:

□□ □□ x□ □□□ □□□□□?

- A. □□□[]
- B. □□
- C. □□□
- D. □□□

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 36

getter □ setter, □□ □□ □ clubMembersmembers □□□ □□□□ name □
yearsMembership □□□ □□ Memberclass□ □□□□□.

□□□ □□□ □□□□ □□ □□□ □ □□ □ □□ Stream □□□□ □□□□□? (2□□
□□□□□.)

- A. □□(Member::getYearsMembership() >= testMembershipLength)
- B. □□□(□□::□□□)
- C. map(testName::compareToIgnoreCase)
- D. □□(□□::□□(0))

Answer: A,C ([LEAVE A REPLY](#))

NEW QUESTION: 37

□□□:

□□ □□ □□□□?

A. java.time □□□□ LocalDate □□□□ □□□□□.

B. □□□ □□□□ java.time.LocalDate□ □□□ □□□ □□□□. □□ □□□ □□□□
□□□□ □ □ □□ □□□□□.

C. □□□□ □□□□ □□ java.time.LocalDate□ □□□□ □□□.

D. java.time □□□□ □□ □□□. Diary □□□□ □□ □□□□□.

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 38

□□□:

/code/a/Test.java

□□:

□□□

/code/b/Best.java

□□:

□□□ b;

□□ □□□ □□□ { }

□□ □□□□ □□ □□□ □□□ □□□□ □□□ □□□□□?

- A. javac -d /code /code/a/Test
- B. □□ /code/a/Test.java
- C. □□ /code/a/Test.java /code/b/Best.java
- D. javac -d /code /code/a/Test.java
- E. java -cp /code a.Test
- F. javac -d /code /code/a/Test.java /code/b/Best.java

Answer: F ([LEAVE A REPLY](#))

NEW QUESTION: 39

□□□:

□□□ □□□□□?

A. 2,34,34,5

B. 2,34,54,5

C. 2,54,54,5

D. 2,34,54,3

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 40

□□□:

□ □□ □ □□ □□? (3□□ □□□□□.)

A. b1.foo(li)□ Hello World!□ □□□□□!

B. b1.foo(li)□ Hello world!□ □□□□□!

C. f1.foo(li)□ Hello World!□ □□□□□!

D. f2.foo(li)□ Hello World!□ □□□□□!

E. f1.foo(li)□ Hello world!□ □□□□□!

F. f2.foo(li)□ Hello world!□ □□□□□!

G. f1.foo(li)□ Hello world!□ □□□□□!

H. b1.foo(li)□ Hello world!□ □□□□□!

I. f2.foo(li)□ Hello world!□ □□□□□!

Answer: A,E,H ([LEAVE A REPLY](#))

NEW QUESTION: 41

□□□:

□□□ □□□□□?

A. 4□□□ NullPointerException□ □□□□□.

B. 10□□□ NullPointerException□ □□□□□□.

C. □□□ □□□ □□□□□□.

D. □□□□□

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 42

□□□ □□ □□:

□□□ □□□□□?

A. □□ □ □

B. □ □□ □□

C. □□ □□ □

D. □□ □ □□

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 43

□□ □□□ □□□□ □□ □□□□ □ □□ □□□ □□□□□? (3□□ □□□□□.)

- A. □□□ □□□□ □□ □□□ □□□□ □□□□□.
- B. □□ □□□ □□□□ □□ □□□ □□□□□.
- C. □□ □□□ □□ □□□ □□ □□□□ □□□□□.
- D. □□□ □□□ □□ □□□ □□□□□.
- E. □□ □ □□ □□□ □□□□ □□□□□.
- F. □□ □□□ □□□□□□.
- G. □□ □□□ □□□□ □□□□ □□□□ □□□□□.

Answer: A,B,C ([LEAVE A REPLY](#))

NEW QUESTION: 44

□□□:

□□□ □□□□□?

- A. 2134
- B. 2143
- C. 214
- D. 234

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 45

□□□:

□□□ □□□□□?

- A. □□ □□□□□, □□
- B. □ □, □□
- C. □□□, □□
- D. □ □, □□

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 46

□□□ □□□ □□□ □□□ □□□ □□□ □□ □□ □□□□ □□□□□.

□□ □□□ □□ {

□□ □□□ □□;

□□ □□<□> □□;

// □□□, □□□ □ □□□ □□□□ □□

}

□□ □□ □□ {

□□ □□□ □□;

□□ □□ □□;

// □□□, □□□ □ □□□ □□□□ □□

}

□□ □□ □□□ □□□□ tList□□ □ □□□ □□ □ □□ □□□ □□□□□□ □□□□
□□ □□□ □□□□□?

- A. □□ D
- B. □□ C
- C. □□ A
- D. □□ B

Answer: B ([LEAVE A REPLY](#))

1Z0-819 □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ 1Z0-819 □□!
DumpTop □ □□ **1Z0-819** □□ □□□ □□□□□□, DumpTop 1Z0-819 □□ □□□
□□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop
1Z0-819 □□□ □□□□□. <https://www.dumptop.com/Oracle/1Z0-819-dump.html> (297
Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 47

□□ □□ □□□ □□□□□.

□ □□ □□ □ □□ □□? (2□□ □□□□□.)

- A. ServiceProvider □□□ com.myimpl □□□□ □□□□ □□□.
- B. □□□ □□□□ ServiceProvider □□□ □□□□□.
- C. ServiceProvider □□□ com.example.api API□ □□□□ □□(□: □□□)□ ID□ □□
□□.
- D. ServiceProvider □□□ □□□□ com.example.api API□ □□□ □ □□ □□□ □□□
□□.
- E. com.example.api API□ □□□ □□□ ServiceAPI□ □□□□ □□ □□□ □□□ □□
□□□ □ □□□□.

Answer: B,D ([LEAVE A REPLY](#))

NEW QUESTION: 48

□□□:

javac□ □□ □□□ □□□□□?

- A. javac□ □□□□ □□□□□ □□□ □□ □□□□ □□□□□. C:
\workspace4\Mycar.java:1:error: □□ import java.lang
- B. javac□ □□□□ □□□□□ □□□ □□ □□□□ □□□□□. C:
\workspace4\Mycar.java:1:error: □□□ java□ □□□□ □□□□.
- C. javac□ □□□ □□ □□ Mycar.java□ □□□□□□.
- D. javac□ □□□□ □□□□□ □□□ □□ □□□□ □□□□□. □□: □□ □□□
Mycar.class□ □□□ □□□ □ □□□□.

Answer: C (LEAVE A REPLY)

NEW QUESTION: 49

□ □ □ □ □ □ □ □ □ □ :

□ □ □ □ □ □ □ □ □ .

```
System.out.println(□□□.getFirstLetter());
```

□ □ □ □ A □ □ □ □ □ 3 □ □ □ □ □ □ □ □ □ □ □ □ □ ?

- A.** ☐ ☐ ☐ ☐ `getFirstLetter() { return Alphabet.values()[1].toString(); }`
- B.** ☐ ☐ ☐ ☐ `getFirstLetter() { return A.toString(); }`
- C.** ☐ ☐ ☐ ☐ `getFirstLetter() { return A.toString(); }`
- D.** ☐ ☐ ☐ ☐ `String getFirstLetter() { return A.toString(); }`

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 50

□ □ □ :

□ □ □ □ □ □ □ □ ?

- A.** $\square: 0 \square: 0 \square: 0$
B. $\square\square\square\square \square\square\square\square\square\square\square.$
C. $\square: 4 \square: 4 \square: 4$
D. $\square: 4 \square: 0 \square: 0$
E. $\square: 0 \square: 4 \square: 4$

Answer: D (LEAVE A REPLY)

NEW QUESTION: 51

□ □ □ :

□ □ □ □ □ □ □ ?

- A.** □□ □ □ □□
B. □□
C. □□□□ □ □□
D. □□□□ □□□□□.
E. □□□
F. □□ □

Answer: B (LEAVE A REPLY)

NEW QUESTION: 52

□ □ □ :

? (2 .)

- A.** □□□ □□□ 2 1 3 4 5□ □□□.
- B.** □□□□□ 1 4 2 3□ □□□□□ □□□ □□□ □ □□□□□.

NEW QUESTION: 55

□□□□□□ □□ □□□□ □□□ □□□□ □□□□ □ □□□ □□□□□? (2□□ □□ □□□.)

- A. □□□□□□ □□□□ □□□ □□ □ □□□ □□ □□□□ □□□□□.
- B. □□□□□□ □□□ □□□□ □□ □ □□□ □□ □□□□ □□□□□.
- C. □□□□□□ □□ □□□ □□ □□ □□□□ □□ □ □□□□.
- D. □□□□□□ □□ □□□□ □□ □ □□□ □□ □□□□ □□□□□.
- E. □□□□□□ □□□ □□ □□□□ □□ □ □□□ □□ □□□□ □□□□□.

Answer: (SHOW ANSWER)

NEW QUESTION: 56

□□□:

□□□ □□□□ □□:

java Hello "Hello World" Hello World

□□□ □□□□□?

- A. Hello WorldHelloWorld
- B. Hello WorldHello World
- C. Hello World Hello World
- D. HelloHello WorldHelloWorld
- E. □□□□ □□□ □□□□□.

Answer: C (LEAVE A REPLY)

NEW QUESTION: 57

□□□:

a.Main□ □ Person□ □□□□□ □□□□ □ □□□ □□□□□? (2□□ □□□□□.)

- A. 1□□□ □□ □□□□ publicpublic Person() {□□ □□□□□.
- B. 2□□□ □□ □□□□ protectedprotected class Main {□ □□□□□.
- C. 1□□□ □□ □□□□ privateprivate Person() {□□ □□□□□.
- D. 2□□□ extends Person□ Main □□□□ □□□□□.
- E. 1□□□ □□□ modifierPerson() {□ □□□□□.

Answer: A,D (LEAVE A REPLY)

NEW QUESTION: 58

□□□:

□□□ □□□□□?

- A. b1
- B. 2
- C. b2
- D. 1
- E. b3

F. 3

G. 1□□ □□□ □□ □□□□ □□□□□.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 59

□□□:

□□□□ □□□ □□ □□□ □□□ □□□□ □□ □ □□ □□□ □□□□□? (2□□ □ □□□□.)

A. getCount

B. getACount

C. getTCount

D. getCCount

E. getTotalCount

Answer: D,E ([LEAVE A REPLY](#))

NEW QUESTION: 60

□□□ □□□□ □□□ Java 8 jar □□□ □□□□ □□ □□ □□□□□□□ □□□□.

com.company.customer;

com.company.customer.orders;

com.company.customer.info;

com.company.sales;

com.company.sales.leads;

com.company.sales.closed;

com.company.orders;

com.company.orders.pending;

com.company.orders.shipped.

□ jar □□□ □□, □□ □ □□□ □ □□ □□□ □□□□□□ □□ module-info.java□

□□□□□?

A)

B)

C)

D)

A. □□ C

B. □□ A

C. □□ D

D. □□ B

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 61

□□□ □□□□□ □□□□□□ □□ □□ □□□ □□□□□?

- A. 1□□ PriceChecker <Plushy extends Products> □□□ □□□□□.
- B. 1□□ PriceChecker(?) prod□ □□□□□.
- C. 1□□ PriceChecker <> prod□ □□□□□□.
- D. 1□□ PriceChecker <Electronics> prod□ □□□□□.

Answer: B ([LEAVE A REPLY](#))

1Z0-819 □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ 1Z0-819 □□!
 DumpTop □ □□ **1Z0-819** □□ □□□ □□□□□□, DumpTop 1Z0-819 □□ □□□
 □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop
 1Z0-819 □□□ □□□□□. <https://www.dumptop.com/Oracle/1Z0-819-dump.html> (297
 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 62

□□□:

□□ □ □□ assert i < 0 ||□ □□□ □ □□□□. □[*i*] <= □[*i* + 1]; values □□□ □□□□
 □ □□□□□□ □□□□□?

- A. 8□ □□
- B. 6□ □□
- C. 5□ □□
- D. 10□ □□

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 63

org.copyservice□ □□ copyServiceAPI□ □□□□. □□□. □□□□□ □□ □□□□ □
 □□□□ □□□□□ □□ module-info.java□ □□□□□?

- A)
- B)
- C)
- D)
- A. □□ B
- B. □□ A
- C. □□ D
- D. □□ C

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 64

□□□:

s□ AQCD□ □□ □□□ □□□□□?

- A. s.replace(s.indexOf("B"), s.indexOf("B"), "Q");
- B. s.replace(s.indexOf("A"), s.indexOf("C"), "Q");
- C. s.replace(s.indexOf("B"), s.indexOf("C"), "Q");
- D. s.replace(s.indexOf("A"), s.indexOf("B"), "Q");

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 65

□□□:

1□□ □□□□□ □□□ □ □ □□ □□□ □□□ □□□ □□? (2□□ □□□□□.)

- A. □□ D
- B. □□ A
- C. □□ B
- D. □□ E
- E. □□ C

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 66

□□□:

□□□ □□□□□?

- A. □□□□
- B. □□□□ □□□□□.
- C. 0
- D. java.lang.IllegalArgumentException□ □□□□□□.
- E. 10

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 67

□□□:

□□□ □□□□□?

- A. □
- B. □□□□ NoSuchElementException□ □□□□□.
- C. □□
- D. □□□□ NullPointerException□ □□□□□.

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 68

□□□:

□□□ Hello world□ □□□ □ □□□ □□□ 1□□ □□ □□□ □□□□ □□□?

- A. Hello myH = new Hello(); □□□ myG = new Greeting(); myG.sayHi();
- B. Hello myH = new Hello();

Hello.Greeting myG = myH.new ☐☐☐();

myG.sayHi();

C. Hello.Greeting myG = new Hello.Greeting() myG.sayHi();

D. Hello myH = new Hello();

Hello.Greeting myG = myH.new Hello.Greeting();

myG.sayHi();

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 69

☐☐☐ ☐☐ ☐☐:

☐☐☐ <☐☐☐ > ☐☐ = List.of(11,12,13,12,13);

☐☐ ☐ ☐☐☐ ☐☐☐ ☐☐☐ ☐☐☐☐☐☐?

A. ☐☐ a = Integer.valueOf(list.get(0));

B. Double e = Double.valueOf(list.get(0));

C. ☐☐ d = list.get(0);

D. int c =list.get(0);

E. ☐☐ f = list.get(0);

F. ☐☐ b = list.get(0);

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 70

☐☐☐:

☐ ☐☐☐ ☐☐☐ ☐☐☐☐☐.

☐☐☐☐ ☐☐☐ ☐ 1☐☐ ☐☐ ☐☐☐☐ ☐☐☐?

A. ☐☐☐ ☐☐☐ = msg -> { return System.out.print(msg); };

B. ☐☐☐ ☐☐☐ = var arg -> {System.out.print(arg);};

C. ☐☐☐ ☐☐☐ = (☐☐☐ ☐☐) -> System.out.print(args);

D. ☐☐☐ ☐☐☐ = System.out::print;

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 71

☐☐☐:

☐☐☐ ☐☐☐☐☐?

A. 300

B. ☐☐

C. 200

D. 100

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 72

Which of the following is a valid Java identifier? (2 correct answers.)

- A. B
- B. D
- C. A
- D. E
- E. C

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 73

Which of the following is a valid Java identifier? (2 correct answers.)

Which of the following is a valid Java identifier? (2 correct answers.)

- A. country.CountryDetails is a valid Java identifier.
- B. country.CountryDetails is a valid Java identifier.
- C. country.countryDetails is a valid Java identifier.
- D. country.CountryDetails is a valid Java identifier.
- E. country.CountryDetails is a valid Java identifier.

Answer: B,D ([LEAVE A REPLY](#))

Which of the following is a valid Java identifier?

/java-9-error-not-in-a-module-on-the-module-source-path

NEW QUESTION: 74

Which of the following is a valid Java identifier?

Which of the following is a valid Java identifier?

- A. country.CountryDetails is a valid Java identifier.
- B. country.CountryDetails is a valid Java identifier.
- C. country 25 is a valid Java identifier.
- D. country.CountryDetails is a valid Java identifier.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 75

Which of the following is a valid Java identifier?

Which of the following is a valid Java identifier? (3 correct answers.)

- A. B A is a valid Java identifier.
- B. A B is a valid Java identifier.
- C. A is a valid Java identifier.

D. B□ □□□ □ □□□□.

E. B□ □□□□ □ □□.

F. A□ □□□ □ □□□□.

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 76

□□□:

Sportscar □□□□ □□ □□□ □□ □□□□□?

A. Sportscar □□□□ Automobile□ □□ □□□□□ □□ □□□□ □□□□□.

B. Sportscar □□□□ □□□□□ Automobile□□ setTurbo □□□□ □□□□□.

C. Sportscar □□ □□□□ □□ □□□ Automobile□ setTurbo □□□□ □□□□ □ □□□.

D. Sportscar □□□□ Automobile □□□□□ □ □□ □□□ □□ □□ □□□□□□.

Answer: A ([LEAVE A REPLY](#))

1Z0-819 □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ 1Z0-819 □□! DumpTop □ □□ **1Z0-819** □□ □□□ □□□□□□, DumpTop 1Z0-819 □□ □□□ □□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop 1Z0-819 □□□ □□□□□. <https://www.dumptop.com/Oracle/1Z0-819-dump.html> (297 Q&As Dumps, **30%OFF** Special Discount: **KrDump**)

NEW QUESTION: 77

□□□ □□ □□:

□□□ □□□□□?

A. 3 : 0

B. 2:3

C. 2: -1

D. -1 : 2

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 78

□□□:

1□□□ Filechannel □□□ □□□□ □□□.

□□ □□ □□□ □ □□□ □□□□□?

A)

B)

C)

D)

- A.** ☐ ☐ C
- B.** ☐ ☐ B
- C.** ☐ ☐ D
- D.** ☐ ☐ A

Answer: D (LEAVE A REPLY)

NEW QUESTION: 79

java.util.function ☐☐☐☐ ☐☐ ☐☐☐☐☐☐ ☐☐ ☐☐☐☐ ☐☐☐☐?

- A.** ToDoubleFunction
B. BiFunction
C. ☐☐☐☐
D. ☐☐☐☐

Answer: A (LEAVE A REPLY)

NEW QUESTION: 80

□ □ □ :

□ □ □ :

□ □□□ □□□ □□□ □□□□□?

Test.class -> java.sql -> java.base

- A.**
- B. Test.class -> java.sql**

java.sql -> java.base

- C.** ☐☐ ☐ jdeps ☐☐☐ ☐☐☐ ☐☐☐☐.

Test.class -> java.base

- #### D. Test.class -> java.sql

Test.class -> java.base

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 81

□ □ □ :

□ □ □ □ □ □ □ □ ?

- A.** 6910 3
B. 10126 3
C. 3
D. 6104 3

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 82

□□ □ □□□ □□□□□ □□□□□□? (2□□ □□□□□.)

- A.** `□□<? □□ □□> □□ = □□□ ArrayList<□□>();`

- B. `ArrayList<>() = new ArrayList<>();`
- C. `ArrayList<>() = new ArrayList<>();`
- D. `ArrayList<>() = new ArrayList<>();`
- E. `ArrayList<>() = new ArrayList<>();`

Answer: A,C ([LEAVE A REPLY](#))

NEW QUESTION: 83

□□□ □□ □□:

□□ □□ print Mondial: domainmodal□ □□□□□ □□□ □ □□ □□ □□□ □□□□
□?

- A. `□□□ □□() .prefix + □□□ □□().□□`
- B. `□□□ + □□, □□`
- C. `□□□ + □□`
- D. `Main.prefix + Main.name`
- E. `Main.prefix + Main.getName()`
- F. `□□□ + getName`

Answer: A,B ([LEAVE A REPLY](#))

NEW QUESTION: 84

□□□:

□□ □ Bar□ 1□□ □□□□ □□□□□ □□□□ □ □□ □ □□ □□□□ □□□□□?
(2□□ □□□□□.)

- A. `□□ □□<□□> foo(TreeSet<□□□> m) { ... }`
- B. `□□ □□<□□> foo(Set<CharSequence> m) { ... }`
- C. `□□ □□<□□> foo(Set<CharSequence> m) { ... }`
- D. `□□ □□<□□> foo(Set<String> m) { ... }`
- E. `□□ ArrayList<□□> foo(Set<CharSequence> m) { ... }`
- F. `□□ ArrayList<□□> foo(Set<□□□> m) { ... }`

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 85

□□□:

□□□

□□□

□□□ □□□□□?

- A. `□□□□□ 1□□□ Process()□ □□□□□.`
- B. `java.lang.IllegalAccessException□ □□□□□□.`
- C. `java.lang.NoSuchMethodException□ □□□□□□.`
- D. `□□□□□ 2□□ Process()□ □□□□□.`
- E. `□□□□□ □□□□□.`

Answer: E ([LEAVE A REPLY](#))

NEW QUESTION: 86

□□□:

□ □□□ □□ □□□□ □□□□□? (2□□ □□□□□.)

A. □□□□ □□□ □□□□□.

B. serialPersistentFields □□ □□□ □□□□□.

C. writeReplace□ □□□□ □□□□□ readResolve□ □□ □□ □□□□ □□□□□.

D. writeReplace□ □□ □□ □□□□ □□□□□ □□□□□ readResolve□ □□□□□.

E. □□□□ □□□□□□.

Answer: B,D ([LEAVE A REPLY](#))

NEW QUESTION: 87

□□□:

□□□

□□□ □□□□□?

A. □□□□ NullPointerException□ □□□□□.

B. □□□□ □□□□□□.

C. 1

□□

□□

D. 1

1

1

E. □□□□ ClassCastException□ □□□□□.

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 88

□□□:

1□□ □□□□ □□□?

A. □□ totalSalary = list.stream().map(e -> e.getSalary() * ratio).reduce(bo).ifPresent(p -> p.doubleValue());

B. □□ totalSalary = list.stream().mapToDouble(e -> e.getSalary() * ratio).sum;

C. double totalSalary = list.stream().map(Employee::getSalary * □
□).reduce(bo).orElse(0.0);

D. □□ totalSalary = list.stream().mapToDouble(e -> e.getSalary() * ratio).reduce(starts,
bo);

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 89

□□□:

□□ □ □□□□ □□□ □□ □□□□ □□□□□ □□□□□? (2□□ □□□□□.)

A. □□ C

B. □□ E

C. □□ D

D. □□ A

E. □□ B

Answer: A,B ([LEAVE A REPLY](#))

NEW QUESTION: 90

□□ □□ □□□ □□□□□□?

A. □□ A

B. □□ B

C. □□ C

D. □□ D

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 91

□□□ □□ □□:

□□□ □□□□□?

EUR -> 0.84

A. GBP -> 0.75

USD -> 1.00

CNY -> 6.42

B. □□□□ □□□□□□.

CNY -> 6.42

C. EUR -> 0.84

□□□ -> 0.75

USD -> 1.00

USD -> 1.00

D. GBP -> 0.75

EUR -> 0.84

CNY -> 6.42

Answer: ([SHOW ANSWER](#))

1Z0-819 □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ 1Z0-819 □□!
DumpTop □ □□ **1Z0-819** □□ □□□ □□□□□□, DumpTop 1Z0-819 □□ □□□
□□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop

NEW QUESTION: 92

□□□ □□ □□:

□□□ □□□□□?

A. -3

B. -1

C. 1

D. 2

Answer: D ([LEAVE A REPLY](#))

NEW QUESTION: 93

□□□ JDBC URL□ □□□□□?

A. http://localhost.mysql.com:3306/database

B. jdbc:mysql://localhost:3306/database

C. http://localhost mysql.jdbc:3306/database

D. jdbe.mysql.com://localhost:3306/database

Answer: B ([LEAVE A REPLY](#))

NEW QUESTION: 94

□□□:

□□□ □□□□□?

A. :□□:□□□:□□□

□□□□□□□□

B. :□□:□□□:□□□

C. APPLE:□□ ORANGE:□□□ BANANA:□□□

D. □□□□□□□□

:□□:□□□:□□□

E. □□:APPLE □□□:ORANGE □□□:BANANA

Answer: E ([LEAVE A REPLY](#))

NEW QUESTION: 95

□□□:

□□ □□□□?

A. System.in□ □□ □□ □□□□□□. □□□□ □□ □□ □□□□.

B. System.in□ □□ □□□□ □□□□ □ □□□□.

C. System.out□ □□ □□ □□□□□□. □□□□ System.out□ □□□ □□ □□□□.

D. System.out□ □□□□□ java.io.OutputStream□ □□□□□□□.

Answer: A ([LEAVE A REPLY](#))

NEW QUESTION: 96

☐ ☐☐☐ ☐☐☐ ☐☐☐☐

□ □ □ □ □ □ □ □ □ □ .

```
System.out.println(Letter.values()[1]);
```

□ □ □ □ 200 □ □ □ □ □ 5 □ □ □ □ □ □ □ □ □ □ □ □ □ ?

```
A. public String toString() { return String.valueOf(ALPHA.v); }
```

```
B. public String toString() { return String.valueOf(Letter.values()[1]); }
```

```
C. public String toString() { return String.valueOf(v); }
```

D. `toString()` { `"200";` }

Answer: C (LEAVE A REPLY)

NEW QUESTION: 97

A)

B)

C)

D)

A. ☐ ☐ B

B. ☐ ☐ A

C. □ □ C

D. □ □ D

Answer: D (LEAVE A REPLY)

NEW QUESTION: 98

□□ □□□ □□□□ □□□□□? (2□□ □□□□□.)

A. `□□<? □□ □□> □□ = □□□ ArrayList<□□>();`

B. `□□<? □□ □□> □□ = □□□ ArrayList<□□>();`

C. `□□<? □□ □□> □□ = □□□ ArrayList<□□□>();`

D. `□□□<? □□ □□> □□ = new ArrayList<Object>();`

E. `□□<? □□ □□□> □□ = □□□ ArrayList<Double>();`

Answer: A,C (LEAVE A REPLY)

NEW QUESTION: 99

Java ☐ ☐☐☐☐ ☐☐☐ ☐☐ ☐☐☐☐

A. ☐☐☐☐ ☐☐☐ ☐☐☐ ☐☐ ☐☐☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐.

B. Java □□□□□ □□□□□ □□□□ □□□ □□□□ □□ □□□□ □□□□□ □□
□□□ □□□□□□□.

C. □□□□ □□□□ □□□□ □□□□ □ □□ □□□ □□□ □□ □□□ □ □□□

□.

D. Java □□□□□ □□ □□ □□□□□ □□□□□ □□□□ □□ □□□□□.

Answer: C ([LEAVE A REPLY](#))

NEW QUESTION: 100

□□□:

□□□ □□□□□?

- A. Hello World! Hello World!
- B. Hello world!□□□□□!
- C. Hello world! Hello world!
- D. Hello world! Hello world!

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 101

□□□:

□□□ □□□□□?

- A. 9
- B. □□□□ □□□ □□□□□.
- C. 3
- D. 6

Answer: ([SHOW ANSWER](#))

NEW QUESTION: 102

□□□:

Y □□□□ □□□□□□ 1□□ □□□ □ □□ □ □□ □□□□□? (2□□ □□□□□.)

- A. map.forEach((k, v) -> set(v));
- B. set(map.values());
- C. super.set(List<String> □)
- D. □□(□□)
- E. super.set(map.values());

Answer: ([SHOW ANSWER](#))

1Z0-819 □□ □□□ □□□□□ □□ DumpTop □□ □□□□ □□□ 1Z0-819 □□!
DumpTop □ □□ **1Z0-819** □□ □□□ □□□□□□, DumpTop 1Z0-819 □□ □□□
□□□□□□□□ □□□ □□□□□□□□. □□□□ □□□ □□□□ □□ DumpTop
1Z0-819 □□□ □□□□□. <https://www.dumptop.com/Oracle/1Z0-819-dump.html> (297
Q&As Dumps, **30%OFF** Special Discount: **KrDump**)