

1Z0-809 Dumps

Java SE 8 Programmer II

<https://www.certleader.com/1Z0-809-dumps.html>



NEW QUESTION 1

Given the code fragment:

```
public static void main (String[] args) throws IOException { BufferedReader brCopy = null;
try (BufferedReader br = new BufferedReader (new FileReader("employee.txt")))
{ // line n1
br.lines().forEach(c -> System.out.println(c)); brCopy = br; //line n2
}
brCopy.ready(); //line n3;
}
```

Assume that the ready method of the BufferedReader, when called on a closed BufferedReader, throws an exception, and employee.txt is accessible and contains valid text.

What is the result?

- A. A compilation error occurs at line n3.
- B. A compilation error occurs at line n1.
- C. A compilation error occurs at line n2.
- D. The code prints the content of the employee.txt file and throws an exception at line n3.

Answer: D

NEW QUESTION 2

Which code fragment is required to load a JDBC 3.0 driver?

- A. Connection con = Connection.getDriver ("jdbc:xyzdata://localhost:3306/EmployeeDB");
- B. Class.forName("org.xyzdata.jdbc.NetworkDriver");
- C. Connection con = DriverManager.getConnection ("jdbc:xyzdata://localhost:3306/EmployeeDB");
- D. DriverManager.loadDriver ("org.xyzdata.jdbc.NetworkDriver");

Answer: B

NEW QUESTION 3

Given the code fragment:

```
for (Course a : Course.values()) {
    System.out.print(a + " Fees " + a.getCost()+" " );
}
```

Which is the valid definition of the Course enum?

- A.

```
enum Course { JAVA(100), J2ME(150);
    private int cost;
    public Course(int c) {
        this.cost = c;
    }
    int getCost() {
        return cost;
    }
}
```
- B.

```
enum Course { JAVA(100), J2ME(150);
    private static int cost;
    private Course(int c) {
        this.cost = c;
    }
    static int getCost() {
        return cost;
    }
}
```

```
C. final enum Course { JAVA(100), J2ME(150);
    private int cost;
    public Course(int c) {
        this.cost = c;
    }
    int getCost() {
        return cost;
    }
    void setCost(int c) {
        this.cost = c;
    }
}

D. enum Course { JAVA(100), J2ME(150);
    private int cost;
    Course(int c) {
        this.cost = c;
    }
    int getCost() {
        return cost;
    }
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: A

NEW QUESTION 4

Given:

```
public class Product {
    public double applyDiscount(double price) {
        assert (price > 0); // line n1
        return price * 0.50;
    }
    public static void main(String[] args) {
        Product p = new Product();
        double newPrice =
            p.applyDiscount(Double.parseDouble(args[0]));
        System.out.println("New Price: " + newPrice);
    }
}
```

and the command: `java Product 0` What is the result?

- A. An AssertionError is thrown.
- B. A compilation error occurs at line n1.
- C. New Price: 0.0
- D. A NumberFormatException is thrown at run time.

Answer: D

NEW QUESTION 5

Given the code fragment:

```
List<String> words = Arrays.asList("win", "try", "best", "luck", "do");
Predicate<String> test1 = w -> {
    System.out.println("Checking...");
    return w.equals("do"); // line n1
};
Predicate test2 = (String w) -> w.length() > 3; // line n2
words.stream()
    .filter(test2)
    .filter(test1)
    .count();
```

What is the result?

- A. A compilation error occurs at line n1.
- B. Checking...
- C. Checking... Checking...
- D. A compilation error occurs at line n2.

Answer: A

NEW QUESTION 6

Which class definition compiles?

```
A. class Vehicle {
    int id;
    public void start() {
        public class Engine { int eNo = id; }
    }
}

B. class Computer {
    private Card sCard = new SoundCard();
    private abstract class Card { }
    private class SoundCard extends Card { }
}

C. class Block {
    int bno;
    static class Counter {
        int locator;
        Counter() { locator = bno; }
    }
}

D. class Product {
    interface Moveable { void move(); }
    Moveable mProduct = new Moveable() {
        void move() { }
    };
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: A

NEW QUESTION 7

Given:


```
public class Foo {
    public void methodB(String s) { System.out.println("Foo " + s ); }
}

public class Bar extends Foo {
    public void methodB(String s) { System.out.println("Bar " + s); }
}

public class Baz extends Bar {
    public void methodB(String s) { System.out.println("Baz " + s); }
}

public class Daze extends Baz{
    private Bar bb = new Bar();
    public void methodB(String s) {
        bb.methodB(s);
        super.methodB(s);
    }
}

public class TestClass {
    public static void main(String[] args) {
        Baz d = new Daze();
        d.methodB("Hello");
    }
}
```

What is the result?

- A. Bar Hello Foo Hello
- B. Bar Hello Baz Hello
- C. Baz Hello
- D. A compilation error occurs in the Daze class.

Answer: C

NEW QUESTION 8

Given the code fragment:

```
List<String> listVal = Arrays.asList("Joe", "Paul", "Alice", "Tom"); System.out.println (
// line n1
);
```

Which code fragment, when inserted at line n1, enables the code to print the count of string elements whose length is greater than three?

- A. listVal.stream().filter(x -> x.length()>3).count()
- B. listVal.stream().map(x -> x.length()>3).count()
- C. listVal.stream().peek(x -> x.length()>3).count().get()
- D. listVal.stream().filter(x -> x.length()>3).mapToInt(x -> x).count()

Answer: A

NEW QUESTION 9

Given the code fragment:

```
public static void main(String[] args) {
    Stream.of("Java", "Unix", "Linux")
        .filter(s -> s.contains("n"))
        .peek(s -> System.out.println("PEEK: " + s))
        // line n1
}
```

Which two code fragments, when inserted at line n1 independently, result in the output PEEK: Unix?

- A. .anyMatch ();
- B. .allMatch ();
- C. .findAny ();
- D. .noneMatch ();
- E. .findFirst ();

Answer: E

NEW QUESTION 10

Given:

```
IntStream stream = IntStream.of (1,2,3); IntFunction<Integer> inFu= x -> y -> x*y; //line n1
IntStream newStream = stream.map(inFu.apply(10)); //line n2 newStream.forEach(System.out::print);
Which modification enables the code fragment to compile?
```

- A. Replace line n1 with: IntFunction<UnaryOperator> inFu = x -> y -> x*y;
- B. Replace line n1 with: IntFunction<IntUnaryOperator> inFu = x -> y -> x*y;
- C. Replace line n1 with: BiFunction<IntUnaryOperator> inFu = x -> y -> x*y;
- D. Replace line n2 with: IntStream newStream = stream.map(inFu.applyAsInt (10));

Answer: B

NEW QUESTION 10

Given:

```
public class Customer { private String fName; private String lName; private static int count;
public customer (String first, String last) {fName = first, lName = last;
++count;}
static { count = 0; }
public static int getCount() {return count; }
}
public class App {
public static void main (String [] args) { Customer c1 = new Customer("Larry", "Smith");
Customer c2 = new Customer("Pedro", "Gonzales"); Customer c3 = new Customer("Penny", "Jones"); Customer c4 = new Customer("Lars", "Svenson"); c4 =
null;
c3 = c2;
System.out.println (Customer.getCount());
}
}
```

What is the result?

- A. 2
- B. 3
- C. 4
- D. 5

Answer: D

NEW QUESTION 14

Given:

```
class Bird {
public void fly () { System.out.print("Can fly"); }
}
class Penguin extends Bird {
public void fly () { System.out.print("Cannot fly"); }
}
and the code fragment: class Birdie {
public static void main (String [ ] args) { fly( ( ) -> new Bird ( ));
fly (Penguin : : new);
}
/* line n1 */
}
```

Which code fragment, when inserted at line n1, enables the Birdie class to compile?

- A. static void fly (Consumer<Bird> bird) { bird :: fly ();}
- B. static void fly (Consumer<? extends Bird> bird) {bird.accept() fly ();}
- C. static void fly (Supplier<Bird> bird) { bird.get() fly ();}
- D. static void fly (Supplier<? extends Bird> bird) { LOST

Answer: C

NEW QUESTION 17

Given the code fragment:

```
Connection con = null;
try {
    // line n1
    if(con != null){
        System.out.print("Connection Established.");
    }

} catch (Exception e) {
    System.out.print(e);
}
```

Assume that dbURL, userName, and password are valid.

Which code fragment can be inserted at line n1 to enable the code to print Connection Established?

- A. Properties prop = new Properties(); prop.put ("user", userName); prop.put ("password", password); con = DriverManager.getConnection (dbURL, prop);
- B. con = DriverManager.getConnection (userName, password, dbURL);
- C. Properties prop = new Properties(); prop.put ("userid", userName); prop.put ("password", password); prop.put("url", dbURL); con = DriverManager.getConnection (prop);
- D. con = DriverManager.getConnection (dbURL); con.setClientInfo ("user", userName); con.setClientInfo ("password", password);

Answer: A

NEW QUESTION 21

Given the structure of the Student table: Student (id INTEGER, name VARCHAR) Given the records from the STUDENT table:

ID	NAME
102	Edwin
103	Edward
103	Edwin

Given the code fragment:

```
Connection conn = DriverManager.getConnection(dbURL, userName, passWord);
Statement st = conn.createStatement();
String query = "DELETE FROM Student WHERE id = 103";
System.out.println("Status: " + st.execute(query));
```

Assume that:

The required database driver is configured in the classpath.

The appropriate database is accessible with the dbURL, userName, and passWord exists. What is the result?

- A. The program prints Status: true and two records are deleted from the Student table.
- B. The program prints Status: false and two records are deleted from the Student table.
- C. A SQLException is thrown at runtime.
- D. The program prints Status: false but the records from the Student table are not deleted.

Answer: B

NEW QUESTION 24

Given the code fragment:

```
public class Foo {
    public static void main (String [ ] args) {
        Map<Integer, String> unsortMap = new HashMap< > ( ); unsortMap.put (10, "z");
        unsortMap.put (5, "b");
        unsortMap.put (1, "d");
        unsortMap.put (7, "e");
        unsortMap.put (50, "j");
        Map<Integer, String> treeMap = new TreeMap <Integer, String> (new Comparator<Integer> ( ) {
            @Override public int compare (Integer o1, Integer o2) {return o2.compareTo
            (o1); } } );
        treeMap.putAll (unsortMap);
        for (Map.Entry<Integer, String> entry : treeMap.entrySet ( ) ) { System.out.print (entry.getValue ( ) + " ");
        }
    }
}
```

What is the result?

- A. A compilation error occurs.
- B. d b e z j
- C. j z e b d
- D. z b d e j

Answer: C

NEW QUESTION 27

Locale	Currency Symbol	Currency Code
US	\$	USD

and the code fragment?

```
double d = 15;  
Locale l = new Locale("en", "US");  
NumberFormat formatter = NumberFormat.getCurrencyInstance(l);  
System.out.println(formatter.format(d));
```

What is the result?

- A. \$15.00
- B. 15 \$
- C. USD 15.00
- D. USD \$15

Answer: A

NEW QUESTION 31

Given the code fragment:

```
BiFunction<Integer, Double, Integer> val = (t1, t2) -> t1 + t2; //line n1  
System.out.println(val.apply(10, 10.5));
```

What is the result?

- A. 20
- B. 20.5
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

Answer: C

NEW QUESTION 32

Given the code fragments:

```
public class Book implements Comparator<Book> { String name;  
double price; public Book () {}  
public Book(String name, double price) { this.name = name;  
this.price = price;  
}  
public int compare(Book b1, Book b2) { return b1.name.compareTo(b2.name);  
}  
public String toString() { return name + ":" + price;  
}  
}
```

and

```
List<Book>books = Arrays.asList (new Book ("Beginning with Java", 2), new book ("A  
Guide to Java Tour", 3));  
Collections.sort(books, new Book()); System.out.print(books);
```

What is the result?

- A. [A Guide to Java Tour:3.0, Beginning with Java:2.0]
- B. [Beginning with Java:2, A Guide to Java Tour:3]
- C. A compilation error occurs because the Book class does not override the abstract method compareTo().
- D. An Exception is thrown at run time.

Answer: A

NEW QUESTION 36

Which statement is true about the DriverManager class?

- A. It returns an instance of Connection.
- B. it executes SQL statements against the database.
- C. It only queries metadata of the database.
- D. it is written by different vendors for their specific database.

Answer: A

Explanation:

The DriverManager returns an instance of Doctrine\DBAL\Connection which is a wrapper around the underlying driver connection (which is often a PDO instance).

NEW QUESTION 37

Given:

```
class CheckClass {  
public static int checkValue (String s1, String s2) { return s1.length() – s2.length();
```



```
}  
}  
and the code fragment:  
String[] strArray = new String [] {"Tiger", "Rat", "Cat", "Lion"}  
//line n1  
for (String s : strArray) { System.out.print (s + " ");  
}
```

Which code fragment should be inserted at line n1 to enable the code to print Rat Cat Lion Tiger?

- A. Arrays.sort(strArray, CheckClass : : checkValue);
- B. Arrays.sort(strArray, (CheckClass : : new) : : checkValue);
- C. Arrays.sort(strArray, (CheckClass : : new).checkValue);
- D. Arrays.sort(strArray, CheckClass : : new : : checkValue);

Answer: A

NEW QUESTION 38

Given:

```
interface P { public void method1(); }  
  
interface Q extends P { public void method1(); }  
  
interface R extends P { public void method2(); }  
  
interface S { public default void method() { } }  
  
interface T { public void method1(); public void method2(); }  
  
interface U { public void method1(); public abstract void method2(); }
```

Which two interfaces can you use to create lambda expressions? (Choose two.)

- A. T
- B. R
- C. P
- D. S
- E. Q
- F. U

Answer: AF

NEW QUESTION 39

Given the code fragment:

```
final String str1 = "Java";  
StringBuffer strBuf = new StringBuffer("Course");  
UnaryOperator<String> u = (str2) -> str1.concat(str2); // line n1  
UnaryOperator<String> c = (str3) -> str3.toLowerCase();  
System.out.println(u.apply(c.apply(strBuf))); // line n2
```

What is the result?

- A. A compilation error occurs at line n1.
- B. courseJava
- C. Javacourse
- D. A compilation error occurs at line n2.

Answer: A

NEW QUESTION 44

In 2015, daylight saving time in New York, USA, begins on March 8th at 2:00 AM. As a result, 2:00 AM becomes 3:00 AM.

Given the code fragment:

```
ZoneId zone = ZoneId.of("America/New_York");  
ZonedDateTime dt = ZonedDateTime.of(LocalDate.of(2015, 3, 8), LocalTime.of(1, 0),  
zone);  
ZonedDateTime dt2 = dt.plusHours(2);  
System.out.print(DateTimeFormatter.ofPattern("H:mm - ").format(dt2));  
System.out.println("difference: " + ChronoUnit.HOURS.between(dt, dt2));
```

Which is the result?

- A. 3:00 – difference: 2

- B. 2:00 – difference: 1
- C. 4:00 – difference: 3
- D. 4:00 – difference: 2

Answer: B

NEW QUESTION 46

Given that data.txt and alldata.txt are accessible, and the code fragment:

```
public void writeFiles() throws IOException {  
    BufferedReader br = new BufferedReader(new FileReader("data.txt"));  
    BufferedWriter bw = new BufferedWriter(new FileWriter("alldata.txt"));  
    String line = null;  
    while ((line = br.readLine()) != null) {  
        bw.append(line + "\n");  
    }  
    // line n1  
}
```

What is required at line n1 to enable the code to overwrite alldata.txt with data.txt?

- A. br.close();
- B. bw.writeLn();
- C. br.flush();
- D. bw.flush();

Answer: D

NEW QUESTION 50

Given the code fragment:

```
String str = "Java is a programming language";  
ToIntFunction<String> indexVal = str::indexOf; //line n1  
int x = indexVal.applyAsInt("Java"); //line n2  
System.out.println(x);
```

What is the result?

- A. 1
- B. A compilation error occurs at line n1.
- C. A compilation error occurs at line n2.

Answer: A

NEW QUESTION 54

Given the code fragment:

```
Path source = Paths.get("/data/december/log.txt");  
Path destination = Paths.get("/data");
```

```
Files.copy(source, destination);
```

and assuming that the file /data/december/log.txt is accessible and contains: 10-Dec-2014 – Executed successfully

What is the result?

- A. A file with the name log.txt is created in the /data directory and the content of the /data/december/ log.txt file is copied to it.
- B. The program executes successfully and does NOT change the file system.
- C. A FileNotFoundException is thrown at run time.
- D. A FileAlreadyExistsException is thrown at run time.

Answer: D

NEW QUESTION 56

Given the code fragment:

```
Stream<List<String>> strs = Stream.of(  
    Arrays.asList("text1", "text2"),  
    Arrays.asList("text2", "text3"));  
Stream<String> bs2 = strs  
    .filter(b -> b.contains("text1"))  
    .flatMap(rs -> rs.stream());  
bs2.forEach(b -> System.out.print(b));
```

What is the result?

- A. text1text2
- B. text1text2text2text3
- C. text1
- D. [text1, text2]

Answer: A

NEW QUESTION 60

Given:

```
public interface LengthValidator {  
    public boolean checkLength(String str);  
}
```

and

```
public class Txt {  
    public static void main(String[] args) {  
        boolean res = new LengthValidator() {  
            public boolean checkLength(String str) {  
                return str.length() > 5 && str.length() < 10;  
            }  
        }.checkLength("Hello");  
    }  
}
```

Which interface from the java.util.function package should you use to refactor the class Txt?

- A. Consumer
- B. Predicate
- C. Supplier
- D. Function

Answer: C**NEW QUESTION 62**

Given the code fragment:

```
//line n1  
Double d = str.average().getAsDouble();  
System.out.println("Average = " + d);
```

Which should be inserted into line n1 to print Average = 2.5?

- A. IntStream str = Stream.of (1, 2, 3, 4);
- B. IntStream str = IntStream.of (1, 2, 3, 4);
- C. DoubleStream str = Stream.of (1.0, 2.0, 3.0, 4.0);
- D. Stream str = Stream.of (1, 2, 3, 4);

Answer: C**NEW QUESTION 65**

Given the code fragment:

```
List<String> nL = Arrays.asList("Jim", "John", "Jeff");  
Function<String, String> funVal = s -> "Hello : ".contact(s);  
nL.Stream()  
.map(funVal)  
.peek(System.out::print);  
What is the result?
```

- A. Hello : Jim Hello : John Hello : Jeff
- B. Jim John Jeff
- C. The program prints nothing.
- D. A compilation error occurs.

Answer: C**NEW QUESTION 70**

Given:

```
class RateOfInterest {  
    public static void main (String[] args) { int rateOfInterest = 0;  
    String accountType = "LOAN";  
    switch (accountType) {  
        case "RD": rateOfInterest = 5; break;  
        case "FD": rateOfInterest = 10; break;  
        default:  
            assert false: "No interest for this account"; //line n1  
    }  
    System.out.println ("Rate of interest:" + rateOfInterest);  
}
```

and the command:

java -ea RateOfInterest
What is the result?

- A. Rate of interest: 0
- B. An AssertionError is thrown.
- C. No interest for this account
- D. A compilation error occurs at line n1.

Answer: B

NEW QUESTION 72

Given:

Item table

- ID, INTEGER: PK
- DESCRIP, VARCHAR(100)
- PRICE, REAL
- QUANTITY< INTEGER

And given the code fragment:

```
9. try {
10. Connection conn = DriverManager.getConnection(dbURL, username, password);
11. String query = "Select * FROM Item WHERE ID = 110";
12. Statement stmt = conn.createStatement();
13. ResultSet rs = stmt.executeQuery(query);
14. while(rs.next()) {
15. System.out.println("ID: " + rs.getInt("Id"));
16. System.out.println("Description: " + rs.getString("Descrip"));
17. System.out.println("Price: " + rs.getDouble("Price"));
18. System.out.println(Quantity: " + rs.getInt("Quantity"));
19. }
20. } catch (SQLException se) {
21. System.out.println("Error");
22. }
```

Assume that:

The required database driver is configured in the classpath.

The appropriate database is accessible with the dbURL, userName, and passWord exists. The SQL query is valid.

What is the result?

- A. An exception is thrown at runtime.
- B. Compilation fails.
- C. The code prints Error.
- D. The code prints information about Item 110.

Answer: D

NEW QUESTION 76

Given the code fragment:

```
Deque<String> queue = new ArrayDeque<>();
queue.add("Susan");
queue.add("Allen");
queue.add("David");
System.out.println(queue.pop());
System.out.println(queue.remove());
System.out.println(queue);
```

What is the result?

- A. DavidDavid[Susan, Allen]
- B. SusanSusan[Susan, Allen]
- C. SusanAllen [David]
- D. DavidAllen [Susan]
- E. SusanAllen[Susan, David]

Answer: C

NEW QUESTION 79

Given the code fragment:


```
// Login time:2015-01-12T21:58:18.817Z
Instant loginTime = Instant.now();
Thread.sleep(1000);

// Logout time:2015-01-12T21:58:19.880Z
Instant logoutTime = Instant.now();

loginTime = loginTime.truncatedTo(ChronoUnit.MINUTES);    // line n1
logoutTime = logoutTime.truncatedTo(ChronoUnit.MINUTES);

if (logoutTime.isAfter(loginTime))
    System.out.println("Logged out at:"+logoutTime);
else
    System.out.println("Can't logout");
```

What is the result?

- A. A compilation error occurs at line n1.
- B. Logged out at: 2015-01-12T21:58:19.880Z
- C. Can't logout
- D. Logged out at: 2015-01-12T21:58:00Z

Answer: D

NEW QUESTION 81

Given:

```
class Block {
    String color;
    int size;
    Block(int size, String color) {
        this.size = size;
        this.color = color;
    }
}
```

and the code fragment:

```
List<Block> blocks = new ArrayList<>();
blocks.add(new Block(10, "Green"));
blocks.add(new Block(7, "Red"));
blocks.add(new Block(12, "Blue"));
Collections.sort(blocks, new ColorSorter());
```

Which definition of the ColorSorter class sorts the blocks list?

```
A. class ColorSorter implements Comparable<Block> {
    public boolean compare(Block o1, Block o2) {
        return o1.color.equals(o2.color);
    }
}

B. class ColorSorter implements Comparable<Block> {
    public int compareTo(Block o1, Block o2) {
        return o1.color.compareTo(o2.color);
    }
}

C. class ColorSorter implements Comparator<Block> {
    public int compare(Block o1, Block o2) {
        return o1.color.compareTo(o2.color);
    }
}

D. class ColorSorter implements Comparator<Block> {
    public boolean compare(Block o1, Block o2) {
        return o1.color.compareTo(o2.color);
    }
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: B

NEW QUESTION 86

Given:

```
class Vehicle { int vno;
String name;
public Vehicle (int vno, String name) { this.vno = vno;;
this.name = name;
}
public String toString () { return vno + ":" + name;
}
}
```

and this code fragment:

```
Set<Vehicle> vehicles = new TreeSet <> (); vehicles.add(new Vehicle (10123, "Ford")); vehicles.add(new Vehicle (10124, "BMW")); System.out.println(vehicles);
```

What is the result?

- A. 10123 Ford10124 BMW
- B. 10124 BMW10123 Ford
- C. A compilation error occurs.
- D. A ClassCastException is thrown at run time.

Answer: D

NEW QUESTION 88

Given the code fragment:

```
try {
    Properties prop = new Properties();
    prop.put("user", userName);
    prop.put("password", passWord);
    Connection conn = DriverManager.getConnection(dbURL, prop);
    if(conn != null){
        System.out.print("Connection Established");
    }
} catch (Exception e) {
    System.out.print(e);
}
```

and the information:

- ▶ The required database driver is configured in the classpath.
- ▶ The appropriate database is accessible with the dbURL, username, and passWord exists. What is the result?

- A. A ClassNotFoundException is thrown at runtime.
- B. The program prints nothing.
- C. The program prints Connection Established.
- D. A SQLException is thrown at runtime.

Answer: C

NEW QUESTION 90

Which two are elements of a singleton class? (Choose two.)

- A. a transient reference to point to the single instance
- B. a public method to instantiate the single instance
- C. a public static method to return a copy of the singleton reference
- D. a private constructor to the class
- E. a public reference to point to the single instance

Answer: BD

NEW QUESTION 91

Given:

```
public class product { int id; int price;
public Product (int id, int price) { this.id = id;
this.price = price;
}
public String toString() { return id + ":" + price; }
}
```

and the code fragment:

```
List<Product> products = Arrays.asList(new Product(1, 10), new Product (2, 30),
new Product (2, 30));
Product p = products.stream().reduce(new Product (4, 0), (p1, p2) -> { p1.price+=p2.price;
return new Product (p1.id, p1.price);}); products.add(p); products.stream().parallel()
.reduce((p1, p2) -> p1.price > p2.price ? p1 : p2)
.i fPresent(System.out: :println); What is the result?
```

- A. 2 : 30
- B. 4 : 0
- C. 4 : 60
- D. 4 : 602 : 303 : 201 : 10
- E. The program prints nothing.

Answer: C

NEW QUESTION 92

Given the content:

MessagesBundle.properties file:

```
username = Enter User Name
password = Enter Password
```

MessagesBundle_fr_FR.properties file:

```
username = Entrez le nom d'utilisateur
password = Entrez le mot de passe
```

and the code fragment:

```
Locale currentLocale = new Locale.Builder().setRegion("FR").setLanguage("fr").build();
ResourceBundle messages = ResourceBundle.getBundle("MessagesBundle", currentLocale);
Enumeration<String> names = messages.getKeys();
while (names.hasMoreElements()) {
    String key = names.nextElement();
    String name = messages.getString(key);
    System.out.println(key + " = " + name);
}
```

What is the result?

- A. username = Entrez le nom d'utilisateur password = Entrez le mot de passe
- B. username = Enter User Name password = Enter Password
- C. A compilation error occurs.
- D. The program prints nothing.

Answer: A

NEW QUESTION 95

Given the code fragment:

```
List<String> li = Arrays.asList("Java", "J2EE", "J2ME", "JSTL", "JSP", "Oracle DB");
Predicate<String> val = p -> p.contains("J");
List<String> neLi = li.stream().filter(x -> x.length() > 3)
    .filter(val).collect(Collectors.toList());
System.out.println(neLi);
```

What is the result?

- A. A compilation error occurs.
- B. [Java, J2EE, J2ME, JSTL, JSP]
- C. null
- D. [Java, J2EE, J2ME, JSTL]

Answer: A

NEW QUESTION 97

Given:

```
public interface Moveable<Integer> {
    public default void walk (Integer distance) {System.out.println("Walking");}
    public void run(Integer distance);
}
```

Which statement is true?

- A. Moveable can be used as below: Moveable<Integer> animal = n -> System.out.println("Running" + n); animal.run(100); animal.walk(20);
- B. Moveable can be used as below: Moveable<Integer> animal = n -> n + 10; animal.run(100); animal.walk(20);
- C. Moveable can be used as below: Moveable animal = (Integer n) -> System.out.println(n); animal.run(100); Moveable.walk(20);
- D. Movable cannot be used in a lambda expression.

Answer: A

NEW QUESTION 102

Given the code fragment:


```
LocalTime now = LocalTime.now();
long timeToBreakfast = 0;
LocalTime office_start = LocalTime.of(7, 30);
if (office_start.isAfter(now)) {
    timeToBreakfast = now.until(office_start, MINUTES);
} else {
    timeToBreakfast = now.until(office_start, HOURS);
}
System.out.println(timeToBreakfast);
```

Assume that the value of now is 6:30 in the morning. What is the result?

- A. An exception is thrown at run time.
- B. 60
- C. 1

Answer: D

NEW QUESTION 103

Which two reasons should you use interfaces instead of abstract classes? (Choose two.)

- A. You expect that classes that implement your interfaces have many common methods or fields, or require access modifiers other than public.
- B. You expect that unrelated classes would implement your interfaces.
- C. You want to share code among several closely related classes.
- D. You want to declare non-static or non-final fields.
- E. You want to take advantage of multiple inheritance of type.

Answer: BE

NEW QUESTION 105

Given the definition of the Employee class:

```
class Employee {
    String dept, name;
    public Employee(String d, String n) {
        dept = d;
        name = n;
    }
    public String toString() {
        return getDept() + ":" + getName();
    }
    public String getDept() { return dept; }
    public String getName() { return name; }
}
```

and this code fragment:

```
List<Employee> emps = Arrays.asList(new Employee("sales", "Ada"),
    new Employee("sales", "Bob"),
    new Employee("hr", "Bob"),
    new Employee("hr", "Eva"));
Stream<Employee> s = emps.stream()
    .sorted(Comparator.comparing((Employee e) -> e.getDept())
        .thenComparing((Employee e) -> e.getName()));
List<Employee> eSorted = s.collect(Collectors.toList());
System.out.println(eSorted);
```

What is the result?

- A. [sales:Ada, hr:Bob, sales:Bob, hr:Eva]
- B. [Ada:sales, Bob:sales, Bob:hr, Eva:hr]
- C. [hr:Eva, hr:Bob, sales:Bob, sales:Ada]

D. [hr:Bob, hr:Eva, sales:Ada, sales:Bob]

Answer: A

NEW QUESTION 110

Given:

```
class MyClass implements AutoCloseable {  
    int test;  
    public void close() { }  
    public MyClass copyObject() { return this; }  
}
```

and the code fragment:

```
MyClass obj = null;  
try (MyClass obj1 = new MyClass()) {  
    obj1.test = 100;  
    obj = obj1.copyObject(); // line n1  
}  
System.out.println(obj.test); // line n2
```

What is the result?

- A. An exception is thrown at line n2.
- B. 100
- C. A compilation error occurs because the try block is declared without a catch or finally block.
- D. A compilation error occurs at line n1.

Answer: D

NEW QUESTION 115

Given the code fragment:

```
Path path1 = Paths.get("/software/../../sys/readme.txt");  
Path path2 = path1.normalize();  
Path path3 = path2.relativize(path1);  
System.out.print(path1.getNameCount());  
System.out.print(" : " + path2.getNameCount());  
System.out.print(" : " + path3.getNameCount());
```

What is the result?

- A. 5 : 3 : 6
- B. 6 : 5 : 6
- C. 3 : 3 : 4
- D. 4 : 4 : 4

Answer: D

NEW QUESTION 120

Which two methods from the `java.util.stream.Stream` interface perform a reduction operation? (Choose two.)

- A. `count()`
- B. `collect()`
- C. `distinct()`
- D. `peek()`
- E. `filter()`

Answer: AB

NEW QUESTION 125

Given:

```
public class Test<T> { private T t;  
    public T get () { return t;  
    }  
    public void set (T t) { this.t = t;  
    }  
    public static void main (String args [ ]) { Test<String> type = new Test<>();  
    Test type 1 = new Test (); //line n1 type.set("Java");
```

```
type1.set(100); //line n2 System.out.print(type.get() + " " + type1.get());
}
}
```

What is the result?

- A. Java 100
- B. java.lang.string@<hashcode>java.lang.Integer@<hashcode>
- C. A compilation error occur
- D. To rectify it, replace line n1 with: Test<Integer> type1 = new Test<>();
- E. A compilation error occur
- F. To rectify it, replace line n2 with: type1.set (Integer(100));

Answer: A

NEW QUESTION 130

Given:

```
class Worker extends Thread { CyclicBarrier cb;
public Worker(CyclicBarrier cb) { this.cb = cb; } public void run () {
try { cb.await();
System.out.println("Worker...");
} catch (Exception ex) { }
}
}
class Master implements Runnable { //line n1 public void run () { System.out.println("Master...");
}
}
```

and the code fragment:

```
Master master = new Master();
```

```
//line n2
```

```
Worker worker = new Worker(cb); worker.start();
```

You have been asked to ensure that the run methods of both the Worker and Master classes are executed. Which modification meets the requirement?

- A. At line n2, insert CyclicBarrier cb = new CyclicBarrier(2, master);
- B. Replace line n1 with class Master extends Thread {
- C. At line n2, insert CyclicBarrier cb = new CyclicBarrier(1, master);
- D. At line n2, insert CyclicBarrier cb = new CyclicBarrier(master);

Answer: C

NEW QUESTION 133

Given the Greetings.properties file, containing:

```
HELLO_MSG = Hello, everyone!
GOODBYE_MSG = Goodbye everyone!
```

and given:

```
import java.util.Enumeration;
import java.util.Locale;
import java.util.ResourceBundle;

public class ResourcesApp {
    public void loadResourceBundle() {
        ResourceBundle resource = ResourceBundle.getBundle("Greetings", Locale.US);
        System.out.println(resource.getObject(1));
    }
    public static void main(String[] args) {
        new ResourcesApp().loadResourceBundle();
    }
}
```

What is the result?

- A. Compilation fails.
- B. GOODBYE_MSG
- C. Hello, everyone!
- D. Goodbye everyone!
- E. HELLO_MSG

Answer: A

NEW QUESTION 138

Given the code fragments:


```
public static Optional<String> getCountry(String loc) {  
    Optional<String> couName = Optional.empty();  
    if ("Paris".equals(loc))  
        couName = Optional.of("France");  
    else if ("Mumbai".equals(loc))  
        couName = Optional.of("India");  
    return couName;  
}
```

and

```
Optional<String> city1 = getCountry("Paris");  
Optional<String> city2 = getCountry("Las Vegas");  
System.out.println(city1.orElse("Not Found"));  
if (city2.isPresent())  
    city2.ifPresent(x -> System.out.println(x));  
else  
    System.out.println(city2.orElse("Not Found"));
```

What is the result?

- A. FranceOptional[NotFound]
- B. Optional [France] Optional [NotFound]
- C. Optional[France] Not Found
- D. FranceNot Found

Answer: D

NEW QUESTION 142

.....

Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

100% Pass Your 1Z0-809 Exam with Our Prep Materials Via below:

<https://www.certleader.com/1Z0-809-dumps.html>