

## 1Z0-809 Dumps

### Java SE 8 Programmer II

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



**NEW QUESTION 1**

Given:

```
class Book { int id;
String name;
public Book (int id, String name) { this.id = id;
this.name = name;
}
public boolean equals (Object obj) { //line n1 boolean output = false;
Book b = (Book) obj;
if (this.name.equals(b.name)) output = true;
}
return output;
}
}
```

and the code fragment:

Book b1 = new Book (101, "Java Programing"); Book b2 = new Book (102, "Java Programing"); System.out.println (b1.equals(b2)); //line n2 Which statement is true?

- A. The program prints true.
- B. The program prints false.
- C. A compilation error occur
- D. To ensure successful compilation, replace line n1 with: boolean equals (Book obj) {
- E. A compilation error occur
- F. To ensure successful compilation, replace line n2 with: System.out.println (b1.equals((Object) b2));

**Answer: A**

**NEW QUESTION 2**

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 3**

Given the content of the employee.txt file: Every worker is a master.

Given that the employee.txt file is accessible and the file allemp.txt does NOT exist, and the code fragment:

```
try {
    List<String> content = Files.readAllLines (Paths.get ("employee.txt"));
    content.stream().forEach(line -> {
        try {
            Files.write(
                Paths.get("allemp.txt"),
                line.getBytes(),
                StandardOpenOption.APPEND
            );
        } catch (IOException e) { System.out.println("Exception 1"); }
    });
} catch (IOException e) { System.out.println("Exception 2"); }
```

What is the result?

- A. Exception 1
- B. Exception 2
- C. The program executes, does NOT affect the system, and produces NO output.
- D. allemp.txt is created and the content of employee.txt is copied to it.

**Answer: A**

**NEW QUESTION 4**

Given the code fragments:

```
public class Test {  
    List<String> list = null;  
    public void printValues() {  
        System.out.print(getList());  
    }  
    public List<String> getList(){ return list; }  
    public void setList(List<String> newList){ list = newList; }  
}
```

and

```
List<String> li = Arrays.asList("Dog", "Cat", "Mouse");  
Test t = new Test();  
t.setList(li.stream().collect(Collectors.toList()));  
t.getList().forEach(Test::printValues);
```

What is the result?

- A. null
- B. A compilation error occurs.
- C. DogCatMouse
- D. [Dog, Cat, Mouse]

**Answer: D**

#### NEW QUESTION 5

What is the result?

```
7. BiPredicate<String, String> bp = (String s1, String s2) -> s1.contains("SG") &&  
    s2.contains("Java");  
8. BiFunction<String, String, Integer> bf = (String s1, String s2) -> {  
9.     int fee = 0;  
10.    if (bp.test(s1, s2)) {  
11.        fee = 100;  
12.    }  
13.    return fee;  
14. };  
15. int fee1 = bf.apply("D101SG", "Java Programming");  
16. System.out.println(fee1);
```

- A. A compilation error occurs at line 7.
- B. 100
- C. A compilation error occurs at line 8.
- D. A compilation error occurs at line 15.

**Answer: A**

#### NEW QUESTION 6

Given the code fragment:

```
public class FileThread implements Runnable { String fName;  
public FileThread(String fName) { this.fName = fName; } public void run () System.out.println(fName);}  
public static void main (String[] args) throws IOException, InterruptedException {  
    ExecutorService executor = Executors.newCachedThreadPool(); Stream<Path> listOfFiles = Files.walk(Paths.get("Java Projects")); listOfFiles.forEach(line -> {  
        executor.execute(new FileThread(line.getFileName().toString ())); //  
        line n1  
    });  
    executor.shutdown(); executor.awaitTermination(5, TimeUnit.DAYS); // line n2  
}
```

The Java Projects directory exists and contains a list of files. What is the result?

- A. The program throws a runtime exception at line n2.
- B. The program prints files names concurrently.
- C. The program prints files names sequentially.
- D. A compilation error occurs at line n1.

**Answer: B**

#### NEW QUESTION 7

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 8**

Given the code fragment:

```
Stream<Path> files = Files.walk(Paths.get(System.getProperty("user.home"))); files.forEach (fName -> { //line n1
try {
Path aPath = fName.toAbsolutePath(); //line n2 System.out.println(fName + ":"
+ Files.readAttributes(aPath, Basic.File.Attributes.class).creationTime ());
} catch (IOException ex) { ex.printStackTrace();
});
});
```

What is the result?

- A. All files and directories under the home directory are listed along with their attributes.
- B. A compilation error occurs at line n1.
- C. The files in the home directory are listed along with their attributes.
- D. A compilation error occurs at line n2.

**Answer:** A

**NEW QUESTION 9**

Given the code fragment:

```
List<Integer> values = Arrays.asList (1, 2, 3); values.stream ()
.map(n -> n*2) //line n1
.p eek(System.out::print) //line n2
.count();
```

What is the result?

- A. 246
- B. The code produces no output.
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

**Answer:** A

**NEW QUESTION 10**

Given:

```
public class Job {
    String name;
    Integer cost;
    Job(String name, Integer cost) {
        this.name = name;
        this.cost = cost;
    }
    String getName() { return name; }
    int getCost() { return cost; }
    public static void main(String[] args) {
        Job j1 = new Job("IT", null);
        DoubleSupplier js1 = j1::getCost;
        System.out.println(j1.getName() + ":" + js1.getAsDouble());
    }
}
```

What is the result?

- A. IT:null
- B. A NullPointerException is thrown at run time.
- C. A compilation error occurs.
- D. IT:0.0

**Answer:** D

**NEW QUESTION 10**

Given the code fragment:

```
Path p1 = Paths.get("/Pics/MyPic.jpeg"); System.out.println (p1.getNameCount() + ":" + p1.getName(1) +
":" + p1.getFileName());
```

Assume that the Pics directory does NOT exist.

What is the result?

- A. An exception is thrown at run time.
- B. 2:MyPic.jpeg: MyPic.jpeg
- C. 1:Pics:/Pics/ MyPic.jpeg
- D. 2:Pics: MyPic.jpeg

**Answer:** B

#### NEW QUESTION 14

Given the code fragment:

```
Path file = Paths.get ("courses.txt");
```

```
// line n1
```

Assume the courses.txt is accessible.

Which code fragment can be inserted at line n1 to enable the code to print the content of the courses.txt file?

- A. `List<String> fc = Files.list(file); fc.stream().forEach (s -> System.out.println(s));`
- B. `Stream<String> fc = Files.readAllLines (file); fc.forEach (s -> System.out.println(s));`
- C. `List<String> fc = readAllLines(file); fc.stream().forEach (s -> System.out.println(s));`
- D. `Stream<String> fc = Files.lines (file); fc.forEach (s -> System.out.println(s));`

**Answer:** D

#### NEW QUESTION 15

Given:

```
public final class IceCream { public void prepare() {}
```

```
}
```

```
public class Cake {
```

```
public final void bake(int min, int temp) {} public void mix() {}
```

```
}
```

```
public class Shop {
```

```
private Cake c = new Cake (); private final double discount = 0.25;
```

```
public void makeReady () { c.bake(10, 120); }
```

```
}
```

```
public class Bread extends Cake {
```

```
public void bake(int minutes, int temperature) {} public void addToppings() {}
```

```
}
```

Which statement is true?

- A. A compilation error occurs in IceCream.
- B. A compilation error occurs in Cake.
- C. A compilation error occurs in Shop.
- D. A compilation error occurs in Bread
- E. All classes compile successfully.

**Answer:** D

#### NEW QUESTION 17

Given the code fragment:

```
List<String> valList = Arrays.asList("", "George", "", "John", "Jim");  
Long newVal = valList.stream()           // line n1  
    .filter(x -> !x.isEmpty())  
    .count();                             // line n2  
System.out.print(newVal);
```

What is the result?

- A. A compilation error occurs at line n2.
- B. 3
- C. 2
- D. A compilation error occurs at line n1.

**Answer:** A

#### NEW QUESTION 18

Given the content:

```
MessagesBundle.properties file:  
  
inquiry = How are you?  
  
MessagesBundle_de_DE.properties file:  
  
inquiry = Wie geht's?
```

and given the code fragment:

```
Locale currentLocale;  
// line 1  
ResourceBundle messages = ResourceBundle.getBundle("MessagesBundle", currentLocale);  
System.out.println(messages.getString("inquiry"));
```

Which two code fragments, when inserted at line 1 independently, enable the code to print "Wie geht's?"

- A. currentLocale = new Locale ("de", "DE");
- B. currentLocale = new Locale.Builder ().setLanguage ("de").setRegion ("DE").build ();
- C. currentLocale = Locale.GERMAN;
- D. currentLocale = new Locale(); currentLocale.setLanguage ("de"); currentLocale.setRegion ("DE");
- E. currentLocale = Locale.getInstance(Locale.GERMAN,Locale.GERMANY);

**Answer:** B

#### NEW QUESTION 20

Given:

```
public class Foo<K, V> {  
    private K key;  
    private V value;  
  
    public Foo(K key, V value) { this.key = key; this.value = value; }  
  
    public static <T> Foo<T, T> twice(T value) { return new Foo<T, T>(value, value); }  
  
    public K getKey() { return key; }  
    public V getValue() { return value; }  
}
```

Which option fails?

- A. Foo<String, Integer> mark = new Foo<String, Integer> ("Steve", 100);
- B. Foo<String, String> pair = Foo.<String>twice ("Hello World!");
- C. Foo<Object, Object> percentage = new Foo<String, Integer>("Steve", 100);
- D. Foo<String, String> grade = new Foo <> ("John", "A");

**Answer:** A

#### NEW QUESTION 25

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 29

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 33

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 37

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 39

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 43

Given the records from the Employee table:



eid	ename
111	Tom
112	Jerry
113	Donald

and given the code fragment: try {  
Connection conn = DriverManager.getConnection (URL, userName, passWord); Statement st = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_UPDATABLE);  
st.execute("SELECT\*FROM Employee"); ResultSet rs = st.getResultSet();  
while (rs.next()) {  
if (rs.getInt(1) ==112) { rs.updateString(2, "Jack");  
}  
}  
rs.absolute(2);  
System.out.println(rs.getInt(1) + " " + rs.getString(2));  
} catch (SQLException ex) { System.out.println("Exception is raised");  
}  
Assume that:

The required database driver is configured in the classpath.

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

- A. The Employee table is updated with the row: 112 Jackand the program prints: 112 Jerry
- B. The Employee table is updated with the row: 112 Jackand the program prints: 112 Jack
- C. The Employee table is not updated and the program prints: 112 Jerry
- D. The program prints Exception is raised.

**Answer:** A

#### NEW QUESTION 45

Given the content of /resources/Message.properties: welcome1="Good day!"

and given the code fragment: Properties prop = new Properties ();

FileInputStream fis = new FileInputStream ("/resources/Message.properties"); prop.load(fis);

System.out.println(prop.getProperty("welcome1")); System.out.println(prop.getProperty("welcome2", "Test"));/line n1

System.out.println(prop.getProperty("welcome3"));

What is the result?

- A. Good day!Testfollowed by an Exception stack trace
- B. Good day!followed by an Exception stack trace
- C. Good day!Test null
- D. A compilation error occurs at line n1.

**Answer:** C

#### NEW QUESTION 50

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 54

Given the code fragments:

class MyThread implements Runnable {

private static AtomicInteger count = new AtomicInteger (0); public void run ( ) {

int x = count.incrementAndGet(); System.out.print (x+" ");

}

}

and

Thread thread1 = new Thread(new MyThread()); Thread thread2 = new Thread(new MyThread()); Thread thread3 = new Thread(new MyThread()); Thread [ ] ta = {thread1, thread2, thread3};

```
for (int x= 0; x < 3; x++) { ta[x].start();  
}
```

Which statement is true?

- A. The program prints 1 2 3 and the order is unpredictable.
- B. The program prints 1 2 3.
- C. The program prints 1 1 1.
- D. A compilation error occurs.

**Answer:** A

#### NEW QUESTION 56

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 60

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 65

Given:

```
class FuelNotAvailException extends Exception { }  
class Vehicle {  
void ride() throws FuelNotAvailException { //line n1  
System.out.println("Happy Journey!");  
}  
}
```

```
class SolarVehicle extends Vehicle {  
public void ride () throws Exception { //line n2  
super.ride ();  
}  
}
```

and the code fragment:

```
public static void main (String[] args) throws FuelNotAvailException, Exception  
{  
Vehicle v = new SolarVehicle ();  
v.ride();  
}
```

Which modification enables the code fragment to print Happy Journey!?

- A. Replace line n1 with public void ride() throws FuelNotAvailException {
- B. Replace line n1 with protected void ride() throws Exception {
- C. Replace line n2 with void ride() throws Exception {
- D. Replace line n2 with private void ride() throws FuelNotAvailException {

**Answer:** B

#### NEW QUESTION 68

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

```
public class Test {  
static Connection newConnection =null;  
public static Connection get DBConnection () throws SQLException { try (Connection con = DriverManager.getConnection(URL, username, password)) {  
newConnection = con;  
}  
}
```

```
return newConnection;  
}  
public static void main (String [] args) throws SQLException { get DBConnection ();  
Statement st = newConnection.createStatement(); st.executeUpdate("INSERT INTO student VALUES (102, 'Kelvin')");  
}  
}
```

Assume that:

The required database driver is configured in the classpath.

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

What is the result?

- A. The program executes successfully and the STUDENT table is updated with one record.
- B. The program executes successfully and the STUDENT table is NOT updated with any record.
- C. A SQLException is thrown as runtime.
- D. A NullPointerException is thrown as runtime.

**Answer:** C

#### NEW QUESTION 72

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 73

Given:

```
class Counter extends Thread {
    int i = 10;
    public synchronized void display(Counter obj) {
        try {
            Thread.sleep(5);
            obj.increment(this);
            System.out.println(i);
        } catch (InterruptedException ex) { }
    }
    public synchronized void increment (Counter obj) {
        i++;
    }
}

public class Test {
    public static void main(String[] args) {
        final Counter obj1 = new Counter();
        final Counter obj2 = new Counter();
        new Thread(new Runnable() {
            public void run() {obj1.display(obj2);
            }
        }).start();
        new Thread(new Runnable() {
            public void run() { obj2.display(obj1); }
        }).start();
    }
}
```

From what threading problem does the program suffer?

- A. race condition
- B. deadlock
- C. starvation
- D. livelock

**Answer:** B

#### NEW QUESTION 74

Given the code fragment:

```
ZonedDateTime depart = ZonedDateTime.of(2015, 1, 15, 3, 0, 0, 0, ZoneID.of("UTC-7"));
```

```
ZonedDateTime arrive = ZonedDateTime.of(2015, 1, 15, 9, 0, 0, 0, ZoneID.of("UTC-5"));
```

```
long hrs = ChronoUnit.HOURS.between(depart, arrive); //line n1 System.out.println("Travel time is" + hrs + "hours");
```

What is the result?

- A. Travel time is 4 hours
- B. Travel time is 6 hours
- C. Travel time is 8 hours
- D. An exception is thrown at line n1.

**Answer:** A

#### NEW QUESTION 78

Given:



```
class Student {
    String course, name, city;
    public Student(String name, String course, String city) {
        this.course = course; this.name = name; this.city = city;
    }
    public String toString() {
        return course + ":" + name + ":" + city;
    }
    public String getCourse() { return course; }
    public String getName() { return name; }
    public String getCity() { return city; }
}
```

and the code fragment:

```
List<Student> stds = Arrays.asList(
    new Student ("Jessy", "Java ME", "Chicago"),
    new Student ("Helen", "Java EE", "Houston"),
    new Student ("Mark", "Java ME", "Chicago"));
stds.stream()
    .collect(Collectors.groupingBy(Student::getCourse))
    .forEach(src, res) -> System.out.println(src));
```

What is the result?

- A. [Java EE: Helen:Houston][Java ME: Jessy:Chicago, Java ME: Mark:Chicago]
- B. Java EEJava ME
- C. [Java ME: Jessy:Chicago, Java ME: Mark:Chicago] [Java EE: Helen:Houston]
- D. A compilation error occurs.

**Answer:** D

#### NEW QUESTION 83

Given the code fragment: Stream<List<String>> iStr= Stream.of ( Arrays.asList ("1", "John"),  
Arrays.asList ("2", null)0;  
Stream<<String> nInSt = iStr.flatMapToInt ((x) -> x.stream ()); nInSt.forEach (System.out :: print);  
What is the result?

- A. 1John2null
- B. 12
- C. A NullPointerException is thrown at run time.
- D. A compilation error occurs.

**Answer:** D

#### NEW QUESTION 84

Given:  
class Student {  
String course, name, city;  
public Student (String name, String course, String city) { this.course = course; this.name = name; this.city = city;  
}  
public String toString() {  
return course + ":" + name + ":" + city;  
}  
and the code fragment: List<Student> stds = Arrays.asList(  
new Student ("Jessy", "Java ME", "Chicago"), new Student ("Helen", "Java EE", "Houston"), new Student ("Mark", "Java ME", "Chicago")); stds.stream()  
.collect(Collectors.groupingBy(Student::getCourse))  
.forEach(src, res) -> System.out.println(src)); What is the result?

- A. [Java EE: Helen:Houston][Java ME: Jessy:Chicago, Java ME: Mark:Chicago]
- B. Java EEJava ME
- C. [Java ME: Jessy:Chicago, Java ME: Mark:Chicago] [Java EE: Helen:Houston]
- D. A compilation error occurs.

**Answer:** B

#### NEW QUESTION 87

Given the code fragment:

```
List<String> colors = Arrays.asList("red", "green", "yellow"); Predicate<String> test = n -> { System.out.println("Searching...");
return n.contains("red");
};
colors.stream()
.f ilter(c -> c.length() > 3)
.allMatch(test); What is the result?
```

- A. Searching...
- B. Searching...Searching...
- C. Searching...Searching... Searching...
- D. A compilation error occurs.

**Answer:** A

#### NEW QUESTION 91

Given the code fragment:

```
List<Integer> codes = Arrays.asList (10, 20); UnaryOperator<Double> uo = s -> s +10.0; codes.replaceAll(uo);
codes.forEach(c -> System.out.println(c)); What is the result?
```

- A. 20.030.0
- B. 1020
- C. A compilation error occurs.
- D. A NumberFormatException is thrown at run time.

**Answer:** C

#### NEW QUESTION 95

Given:

```
public enum USCurrency { PENNY (1),
NICKLE(5), DIME (10), QUARTER(25);
private int value;
public USCurrency(int value) { this.value = value;
}
public int getValue() {return value;}
}
public class Coin {
public static void main (String[] args) { USCurrency usCoin =new USCurrency.DIME; System.out.println(usCoin.getValue());
}
}
```

Which two modifications enable the given code to compile? (Choose two.)

- A. Nest the USCurrency enumeration declaration within the Coin class.
- B. Make the USCurrency enumeration constructor private.
- C. Remove the new keyword from the instantiation of usCoin.
- D. Make the getter method of value as a static method.
- E. Add the final keyword in the declaration of value.

**Answer:** BC

#### NEW QUESTION 99

Given the code fragment:

```
LocalDate valentinesDay =LocalDate.of(2015, Month.FEBRUARY, 14); LocalDate nextYear = valentinesDay.plusYears(1); nextYear.plusDays(15); //line n1
System.out.println(nextYear); What is the result?
```

- A. 2016-02-14
- B. A DateTimeException is throw
- C. 2016-02-29
- D. A compilation error occurs at line n1.

**Answer:** A

#### NEW QUESTION 101

Given the definition of the Country class: public class country {

```
public enum Continent {ASIA, EUROPE} String name;
Continent region;
public Country (String na, Continent reg) { name = na, region = reg;
}
public String getName () {return name;} public Continent getRegion () {return region;}
}
```

and the code fragment:

```
List<Country> couList = Arrays.asList (
new Country ("Japan", Country.Continent.ASIA), new Country ("Italy", Country.Continent.EUROPE),
new Country ("Germany", Country.Continent.EUROPE)); Map<Country.Continent, List<String>> regionNames = couList.stream ()
.c ollect(Collectors.groupingBy (Country ::getRegion, Collectors.mapping(Country::getName, Collectors.toList())))); System.out.println(regionNames);
```

- A. {EUROPE = [Italy, Germany], ASIA = [Japan]}
- B. {ASIA = [Japan], EUROPE = [Italy, Germany]}
- C. {EUROPE = [Germany, Italy], ASIA = [Japan]}

D. {EUROPE = [Germany], EUROPE = [Italy], ASIA = [Japan]}

**Answer:** B

#### NEW QUESTION 105

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 106

Given the code fragment:

```
List<Integer> prices = Arrays.asList(3, 4, 5);
prices.stream()
    .filter(e -> e > 4)
    .peek(e -> System.out.print("Price " + e))           // line n1
    .map(n -> n - 1)                                     // line n2
    .peek(n -> System.out.println(" New Price " + n));   // line n3
```

Which modification enables the code to print Price 5 New Price 4?

- A. Replace line n2 with .map (n -> System.out.println ("New Price" + n –1)) and remove line n3
- B. Replace line n2 with .mapToInt (n -> n – 1);
- C. Replace line n1 with .forEach (e -> System.out.print ("Price" + e))
- D. Replace line n3 with .forEach (n -> System.out.println ("New Price" + n));

**Answer:** A

#### NEW QUESTION 109

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 112

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 113

Given the code fragment:

```
Map<Integer, Integer> mVal = new HashMap<>();
mVal.put(1, 10);
mVal.put(2, 20);
//line n1
c.accept(1, 2);
mVal.forEach(c);
```

Which statement can be inserted into line n1 to print 1,2; 1,10; 2,20;?

- A. BiConsumer<Integer,Integer> c = (i, j) -> {System.out.print (i + "," + j+ "; ");};
- B. BiFunction<Integer, Integer, String> c = (i, j) -> {System.out.print (i + "," + j+ "; ");};
- C. BiConsumer<Integer, Integer, String> c = (i, j) -> {System.out.print (i + "," + j+ "; ");};
- D. BiConsumer<Integer, Integer, Integer> c = (i, j) -> {System.out.print (i + "," + j+ "; ");};

**Answer:** B

#### NEW QUESTION 118

Given:

```
public class StrMan {
    public static void doStuff(String s) {
        try {
            if (s == null) {
                throw new NullPointerException();
            }
        } finally {
            System.out.println("-finally-");
        }
        System.out.println("-doStuff-");
    }
    public static void main (String[] args) {
        try {
            doStuff(null);
        } catch (NullPointerException npe) {
            System.out.println("-catch-");
        }
    }
}
```

What is the result?



- A. –catch--finally--dostuff-
- B. –catch-
- C. –finally--catch-
- D. –finally-dostuff--catch-

**Answer: C**

#### NEW QUESTION 119

The data.doc, data.txt and data.xml files are accessible and contain text. Given the code fragment:

```
Stream<Path> paths = Stream.of (Paths. get("data.doc"),  
Paths. get("data.txt"),  
Paths. get("data.xml"));  
paths.filter(s-> s.toString().endsWith("txt")).forEach( s -> {  
try { Files.readAllLines(s)  
.stream()  
.forEach(System.out::println); //line n1  
} catch (IOException e) { System.out.println("Exception");  
}  
}  
);  
What is the result?
```

- A. The program prints the content of data.txt file.
- B. The program prints: Exception<<The content of the data.txt file>> Exception
- C. A compilation error occurs at line n1.
- D. The program prints the content of the three files.

**Answer: A**

#### NEW QUESTION 124

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 128

Given:

```
class Engine {  
    double fuelLevel;  
    Engine(int fuelLevel) { this.fuelLevel = fuelLevel; }  
    public void start() {  
        // line n1  
        System.out.println("Started");  
    }  
    public void stop() { System.out.println("Stopped"); }  
}
```

Your design requires that:

- ▶ fuelLevel of Engine must be greater than zero when the start() method is invoked.
- ▶ The code must terminate if fuelLevel of Engine is less than or equal to zero.

Which code fragment should be added at line n1 to express this invariant condition?

- A. assert (fuelLevel) : "Terminating...";
- B. assert (fuelLevel > 0) : System.out.println ("Impossible fuel");
- C. assert fuelLevel < 0: System.exit(0);
- D. assert fuelLevel > 0: "Impossible fuel" ;

**Answer: C**

**NEW QUESTION 132**

Given the code fragment:

```
List<Integer> list1 = Arrays.asList(10, 20); List<Integer> list2 = Arrays.asList(15, 30);  
//line n1
```

Which code fragment, when inserted at line n1, prints 10 20 15 30?

- A. `Stream.of(list1, list2).flatMap(list -> list.stream()).forEach(s -> System.out.print(s + " "));`
- B. `Stream.of(list1, list2).flatMap(list -> list.intStream()).forEach(s -> System.out.print(s + " "));`
- C. `list1.stream().flatMap(list2.stream()).flatMap(e1 -> e1.stream()).forEach(s -> System.out.println(s + " "));`
- D. `Stream.of(list1, list2).flatMapToInt(list -> list.stream()).forEach(s -> System.out.print(s + " "));`

**Answer:** A

**NEW QUESTION 133**

For which three objects must a vendor provide implementations in its JDBC driver? (Choose three.)

- A. Time
- B. Date
- C. Statement
- D. ResultSet
- E. Connection
- F. SQLException
- G. DriverManager

**Answer:** CDE

**Explanation:**

Database vendors support JDBC through the JDBC driver interface or through the ODBC connection. Each driver must provide implementations of `java.sql.Connection`, `java.sql.Statement`, `java.sql.PreparedStatement`, `java.sql.CallableStatement`, and `java.sql.ResultSet`. They must also implement the `java.sql.Driver` interface for use by the generic `java.sql.DriverManager` interface.

**NEW QUESTION 136**

Given:

```
final class Folder { //line n1  
//line n2  
public void open () { System.out.print("Open");  
}  
}  
public class Test {  
public static void main (String [] args) throws Exception { try (Folder f = new Folder()) {
```

- A. `f.open();}}`Which two modifications enable the code to print Open Close? (Choose two.)
- B. Replace line n1 with:`class Folder implements AutoCloseable {`
- C. Replace line n1 with:`class Folder extends Closeable {`
- D. Replace line n1 with:`class Folder extends Exception {`
- E. At line n2, insert: `final void close () {System.out.print("Close");}`
- F. At line n2, insert:`public void close () throws IOException { System.out.print("Close");}`

**Answer:** AE

**NEW QUESTION 138**

Given the code fragment:

```
List<String> gwords = Arrays.asList("why ", "what ", "when ");  
BinaryOperator<String> operator = (s1, s2) -> s1.concat(s2); // line n1  
String sen = gwords.stream()  
    .reduce("Word: ", operator);  
System.out.println(sen);
```

What is the result?

- A. Word: why what when
- B. Word: why Word: why what Word: why what when
- C. Word: why Word: what Word: when
- D. Compilation fails at line n1.

**Answer:** A

**NEW QUESTION 143**

Given the definition of the Vehicle class: `class Vehicle {`

```
String name;  
void setName (String name) { this.name = name;  
}  
String getName() { return name;  
}  
}
```

Which action encapsulates the Vehicle class?

- A. Make the Vehicle class public.
- B. Make the name variable public.
- C. Make the setName method public.
- D. Make the name variable private.
- E. Make the setName method private.
- F. Make the getName method private.

**Answer:** D

#### NEW QUESTION 144

Given the code fragment: UnaryOperator<Integer> uo1 = s -> s\*2; line n1  
List<Double> loanValues = Arrays.asList(1000.0, 2000.0); loanValues.stream()  
.filter(lv -> lv >= 1500)  
.map(lv -> uo1.apply(lv))  
.forEach(s -> System.out.print(s + " ")); What is the result?

- A. 4000.0
- B. 4000
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

**Answer:** D

#### NEW QUESTION 149

Given:

```
class Product {  
    String name;  
    int qty;  
    public String toString(){  
        return name;  
    }  
    public Product(String name, int qty) {  
        this.name = name;  
        this.qty = qty;  
    }  
    static class ProductFilter {  
        public boolean isAvailable(Product p) {    // line n1  
            return p.qty >= 10;  
        }  
    }  
}
```

and the code fragment:

```
List<Product> products = Arrays.asList(  
    new Product("MotherBoard", 5),  
    new Product("Speaker", 20));  
products.stream()  
    .filter(Product.ProductFilter::isAvailable) // line n2  
    .forEach(p -> System.out.println(p));
```

Which modification enables the code fragment to print Speaker?

- A. Implement Predicate in the Product.ProductFilter class and replace line n2 with .filter (p-> p.ProductFilter.test (p))
- B. Replace line n1 with:public static boolean isAvailable (Product p) {
- C. Replace line n2 with:.filter (p -> p.ProductFilter: :isAvailable (p))
- D. Replace line n2 with:.filter (p -> Product: :ProductFilter: :isAvailable (p))

**Answer:** B

#### NEW QUESTION 153

Given the code fragment:

- 9. Connection conn = DriverManager.getConnection(dbURL, userName, passWord);
- 10. String query = "SELECT id FROM Employee";
- 11. try (Statement stmt = conn.createStatement()) {
- 12. ResultSet rs = stmt.executeQuery(query);
- 13. stmt.executeQuery("SELECT id FROM Customer");
- 14. while (rs.next()) {
- 15. //process the results

```
16. System.out.println("Employee ID: "+ rs.getInt("id"));
17. }
18. } catch (Exception e) {
19. System.out.println ("Error");
20. }
```

Assume that:

The required database driver is configured in the classpath.

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

The Employee and Customer tables are available and each table has id column with a few records and the SQL queries are valid.

What is the result of compiling and executing this code fragment?

- A. The program prints employee IDs.
- B. The program prints customer IDs.
- C. The program prints Error.
- D. compilation fails on line 13.

**Answer: C**

#### NEW QUESTION 158

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 159

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 164

Which statement is true about java.util.stream.Stream?

- A. A stream cannot be consumed more than once.
- B. The execution mode of streams can be changed during processing.
- C. Streams are intended to modify the source data.
- D. A parallel stream is always faster than an equivalent sequential stream.

**Answer: B**

#### NEW QUESTION 169

Given the code fragment:

```
List<String> empDetails = Arrays.asList("100, Robin, HR", "200, Mary, AdminServices",
"101, Peter, HR");
empDetails.stream()
.filter(s-> s.contains("1"))
.sorted()
.f orEach(System.out::println); //line n1
What is the result?
```

- A. 100, Robin, HR101, Peter, HR
- B. A compilation error occurs at line n1.
- C. 100, Robin, HR101, Peter, HR200, Mary, AdminServices
- D. 100, Robin, HR200, Mary, AdminServices101, Peter, HR

**Answer: A**



**NEW QUESTION 172**

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 176**

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 180**

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 185**

Given the code fragment:

```
List<String> nums = Arrays.asList("EE", "SE");
String ans = nums
    .parallelStream()
    .reduce("Java ", (a, b) -> a.concat(b));
System.out.print(ans);
```

What is the result?

- A. Java EEJava EESE
- B. Java EESE
- C. The program prints either:Java EEJava SE orJava SEJava EE
- D. Java EEJava SE

**Answer:** D

#### NEW QUESTION 187

Given the content of Operator.java, EngineOperator.java, and Engine.java files:

```
Operator.java:
public abstract class Operator {
    protected void turnON();
    protected void turnOFF();
}

EngineOperator.java:
public class EngineOperator extends Operator{
    public final void turnON() { System.out.print("ON "); }
    public final void turnOFF() { System.out.println("OFF"); }
}

Engine.java:
public class Engine{
    Operator m = new EngineOperator();
    public void operate() {
        m.turnON();
        m.turnOFF();
    }
}
```

and the code fragment:

```
Engine carEngine = new Engine();
carEngine.operate();
```

What is the result?

- A. The Engine.java file fails to compile.
- B. The EngineOperator.java file fails to compile.
- C. The Operator.java file fails to compile.
- D. ON OFF

**Answer:** A

#### NEW QUESTION 191

Given the code fragments:

```
class Caller implements Callable<String> { String str;
public Caller (String s) {this.str=s;}
public String call()throws Exception { return str.concat ("Caller");}
}
```

```
class Runner implements Runnable { String str;
public Runner (String s) {this.str=s;}
public void run () { System.out.println (str.concat ("Runner"));}
}
```

and

```
public static void main (String[] args) InterruptedException, ExecutionException
{
    ExecutorService es = Executors.newFixedThreadPool(2); Future f1 = es.submit (new Caller ("Call"));
    Future f2 = es.submit (new Runner ("Run")); String str1 = (String) f1.get();
}
```

```
String str2 = (String) f2.get(); //line n1 System.out.println(str1+ ":" + str2);
}
```

What is the result?

- A. The program prints: Run RunnerCall Caller : nullAnd the program does not terminate.
- B. The program terminates after printing: Run RunnerCall Caller : Run
- C. A compilation error occurs at line n1.
- D. An Execution is thrown at run time.

**Answer:** A

#### NEW QUESTION 193

Given the code fragment:

```
Deque<Integer> nums = new ArrayDeque<>();
nums.add(1000);
nums.push(2000);
nums.add(3000);
nums.push(4000);
Integer i1 = nums.remove();
Integer i2 = nums.pop();
System.out.println(i1 + " : " + i2);
```

What is the result?

- A. 4000 : 2000
- B. 4000 : 1000
- C. 1000 : 4000
- D. 1000 : 2000

**Answer:** B

#### NEW QUESTION 197

Given the code fragment:

```
class CallerThread implements Callable<String> { String str;
public CallerThread(String s) {this.str=s;} public String call() throws Exception { return str.concat("Call");
}
}
and
public static void main (String[] args) throws InterruptedException, ExecutionException
{
ExecutorService es = Executors.newFixedThreadPool(4); //line n1 Future f1 = es.submit (newCallerThread("Call"));
String str = f1.get().toString(); System.out.println(str);
}
```

Which statement is true?

- A. The program prints Call Call and terminates.
- B. The program prints Call Call and does not terminate.
- C. A compilation error occurs at line n1.
- D. An ExecutionException is thrown at run time.

**Answer:** B

#### NEW QUESTION 200

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 on non-final fields.
- E. You want to take advantage of multiple inheritance of type.

**Answer:** BE

#### NEW QUESTION 201

Given: Book.java:

```
public class Book {
private String read(String bname) { return "Read" + bname }
}
```

EBook.java:

```
public class EBook extends Book {
public class String read (String url) { return "View" + url }
}
```

Test.java:

```
public class Test {
public static void main (String[] args) { Book b1 = new Book();
```

```
b1.read("Java Programing"); Book b2 = new EBook();
b2.read("http://ebook.com/ebook");
}
}
```

What is the result?

- A. Read Java Programming View [http:// ebook.com/ebook](http://ebook.com/ebook)
- B. Read Java Programming Read [http:// ebook.com/ebook](http://ebook.com/ebook)
- C. The EBook.java file fails to compile.
- D. The Test.java file fails to compile.

**Answer: D**

#### NEW QUESTION 204

Given:

```
public class Emp { String fName; String lName;
public Emp (String fn, String ln) { fName = fn;
lName = ln;
}
public String getfName() { return fName; } public String getlName() { return lName; }
}
```

and the code fragment: `List<Emp> emp = Arrays.asList ( new Emp ("John", "Smith"),  
new Emp ("Peter", "Sam"),  
new Emp ("Thomas", "Wale")); emp.stream()  
//line n1`

`.collect(Collectors.toList());`

Which code fragment, when inserted at line n1, sorts the employees list in descending order of fName and then ascending order of lName?

- A. `.sorted (Comparator.comparing(Emp::getfName).reserved().thenComparing(Emp::getlName))`
- B. `.sorted (Comparator.comparing(Emp::getfName).thenComparing(Emp::getlName))`
- C. `.map(Emp::getfName).sorted(Comparator.reserveOrder())`
- D. `.map(Emp::getfName).sorted(Comparator.reserveOrder()).map (Emp::getlName).reserved`

**Answer: A**

#### NEW QUESTION 209

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 210

Given:

```
class Product {
    String pname;
    public Product(String pname) {
        this.pname = pname;
    }
}
```

and the code fragment:

```
Product p1 = new Product ("PowerCharger");
Product p2 = p1;
System.out.println(p1.equals(p2));
Product p3 = new Product ("PowerCharger");
System.out.println(p1.equals(p3));
```

What is the result?

- A. truetrue
- B. falsetrue
- C. falsefalse
- D. truefalse

**Answer: B**



**NEW QUESTION 211**

Given the code fragments:

```
class Employee { Optional<Address> address;
Employee (Optional<Address> address) { this.address = address;
}
public Optional<Address> getAddress() { return address; }
}
class Address {
String city = "New York";
public String getCity { return city; } public String toString() {
return city;
}
}
and
Address address = null;
Optional<Address> addrs1 = Optional.ofNullable (address);
Employee e1 = new Employee (addrs1);
String eAddress = (addrs1.isPresent()) ? addrs1.get().getCity() : "City Not available";
What is the result?
```

- A. New York
- B. City Not available
- C. null
- D. A NoSuchElementException is thrown at run time.

**Answer:** B

**NEW QUESTION 214**

Given:

```
class ImageScanner implements AutoCloseable { public void close () throws Exception { System.out.print ("Scanner closed.");
}
public void scanImage () throws Exception { System.out.print ("Scan.");
throw new Exception("Unable to scan.");
}
}
class ImagePrinter implements AutoCloseable { public void close () throws Exception { System.out.print ("Printer closed.");
}
public void printImage () {System.out.print("Print."); }
}
and this code fragment:
try (ImageScanner ir = new ImageScanner(); ImagePrinter iw = new ImagePrinter()) { ir.scanImage();
iw.printImage();
} catch (Exception e) { System.out.print(e.getMessage());
}
What is the result?
```

- A. Scan.Printer close
- B. Scanner close
- C. Unable to scan.
- D. Scan.Scanner close
- E. Unable to scan.
- F. Sca
- G. Unable to scan.
- H. Sca
- I. Unable to sca
- J. Printer closed.

**Answer:** A

**NEW QUESTION 217**

Which action can be used to load a database driver by using JDBC3.0?

- A. Add the driver class to the META-INF/services folder of the JAR file.
- B. Include the JDBC driver class in a jdbc.properties file.
- C. Use the java.lang.Class.forName method to load the driver class.
- D. Use the DriverManager.getDriver method to load the driver class.

**Answer:** C

**NEW QUESTION 221**

Given the code fragments:

```
class R implements Runnable {  
    public void run() { System.out.println("Run..."); }  
}  
  
class C implements Callable<String> {  
    public String call() throws Exception { return "Call..."; }  
}
```

and

```
ExecutorService es = Executors.newSingleThreadExecutor();  
es.execute(new R()); // line n1  
Future<String> f1 = es.submit(new C()); // line n2  
System.out.println(f1.get());  
es.shutdown();
```

What is the result?

- A. The program prints Run... and throws an exception.
- B. A compilation error occurs at line n1.
- C. Run...Call...
- D. A compilation error occurs at line n2.

**Answer:** B

#### NEW QUESTION 223

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 225

Given:

```
class Person {  
    String name;  
    int age;  
    public Person(String name, int age) {  
        this.name = name;  
        this.age = age;  
    }  
    public String getName(){ return name; }  
    public int getAge(){ return age; }  
}
```

and the code fragment:

```
List<Person> sts = Arrays.asList(  
    new Person("Jack", 30),  
    new Person("Mike Hill", 21),  
    new Person("Thomas Hill", 24));  
Stream<Person> resList = sts.stream().filter(s -> s.getAge() >= 25);    // line n1  
long count = resList.filter(s -> s.getName().contains("Hill")).count();  
System.out.print(count);
```

What is the result?

- A. A compilation error occurs at line n1.
- B. An Exception is thrown at run time.
- C. 2

**Answer:** B

**NEW QUESTION 226**

.....

## 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>