

# Oracle

## Exam Questions 1Z0-809

Java SE 8 Programmer II



**NEW QUESTION 1**

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 2**

Given the code fragment:

```
5. IntConsumer consumer = e -> System.out.println(e);
6. Integer value = 90;
7. /* insert code fragment here */
8. consumer.accept(result);
```

Which code fragment, when inserted at line 7, enables printing 100?

- A. Function<Integer> funRef = e -> e + 10; Integer result = funRef.apply(value);
- B. IntFunction funRef = e -> e + 10; Integer result = funRef.apply (10);
- C. ToIntFunction<Integer> funRef = e -> e + 10; int result = funRef.applyAsInt (value);
- D. ToIntFunction funRef = e -> e + 10; int result = funRef.apply (value);

**Answer:** A

**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:

```
class Resource implements AutoCloseable {  
    public void close() throws Exception {  
        System.out.print("Close-");  
    }  
    public void open() {  
        System.out.print("Open-");  
    }  
}
```

and this code fragment:

```
Resource res1 = new Resource();  
try {  
    res1.open();  
    res1.close();  
} catch (Exception e) {  
    System.out.println("Exception - 1");  
}  
try (res1 = new Resource()) { // line n1  
    res1.open();  
} catch (Exception e) {  
    System.out.println("Exception - 2");  
}
```

What is the result?

- A. Open-Close- Exception – 1 Open-Close-
- B. Open-Close-Open-Close-
- C. A compilation error occurs at line n1.
- D. Open-Close-Open-

**Answer:** C

#### 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

Given the code fragment:

```
ProductCode<Number, Integer> c1 = new ProductCode<Number, Integer>(); /* c1  
instantiation */  
ProductCode<Number, String> c2 = new ProductCode<Number, String>(); /* c2  
instantiation */
```

You have been asked to define the ProductCode class. The definition of the ProductCode class must allow c1 instantiation to succeed and cause a compilation

error on c2 instantiation.

Which definition of ProductCode meets the requirement?

```
A. class ProductCode<T, S<Integer>> {
    T c1;
    S c2;
}

B. class ProductCode<T, S extends T> {
    T c1;
    S c2;
}

C. class ProductCode<T, S> {
    T c1;
    S c2;
}

D. class ProductCode<T, S super T> {
    T c1;
    S c2;
}
```

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

**Answer:** B

#### NEW QUESTION 7

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 8

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 9

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 10

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 10**

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 12**

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 14**

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 17**

Which two statements are true about localizing an application? (Choose two.)

- A. Support for new regional languages does not require recompilation of the code.
- B. Textual elements (messages and GUI labels) are hard-coded in the code.
- C. Language and region-specific programs are created using localized data.
- D. Resource bundle files include data and currency information.
- E. Language codes use lowercase letters and region codes use uppercase letters.

**Answer: AE**

### NEW QUESTION 22

Given the code fragment:

```
public void recDelete (String dirName) throws IOException { File [ ] listOfFiles = new File (dirName) .listFiles();
if (listOfFiles != null && listOfFiles.length >0) {
for (File aFile : listOfFiles) { if (aFile.isDirectory ()) {
recDelete (aFile.getAbsolutePath ());
} else {
if (aFile.getName ().endsWith (".class")) aFile.delete ();
}
}
}
}
```

Assume that Projects contains subdirectories that contain .class files and is passed as an argument to the recDelete () method when it is invoked. What is the result?

- A. The method deletes all the .class files in the Projects directory and its subdirectories.
- B. The method deletes the .class files of the Projects directory only.
- C. The method executes and does not make any changes to the Projects directory.
- D. The method throws an IOException.

**Answer:** A

### NEW QUESTION 27

Given:

```
1. abstract class Shape {
2. Shape ( ) { System.out.println ("Shape"); }
3. protected void area ( ) { System.out.println ("Shape"); } 4. }
5.
6. class Square extends Shape {
7. int side;
8. Square int side {
9. /* insert code here */
10. this.side = side;
11. }
12. public void area ( ) { System.out.println ("Square"); }
13. }
14. class Rectangle extends Square {
15. int len, br;
16. Rectangle (int x, int y) {
17. /* insert code here */
18. len = x, br = y;
19. }
20. void area ( ) { System.out.println ("Rectangle"); }
21. }
```

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

- A. At line 1, remove abstract
- B. At line 9, insert super ( );
- C. At line 12, remove public
- D. At line 17, insert super (x);
- E. At line 17, insert super (); super.side = x;
- F. At line 20, use public void area ( ) {

**Answer:** DF

### NEW QUESTION 29

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 that /green.txt and /colors/yellow.txt are accessible, and the code fragment: Path source = Paths.get("/green.txt"); Path target = Paths.get("/colors/yellow.txt"); Files.move(source, target, StandardCopyOption.ATOMIC\_MOVE); Files.delete(source); Which statement is true?

- A. The green.txt file content is replaced by the yellow.txt file content and the yellow.txt file is deleted.
- B. The yellow.txt file content is replaced by the green.txt file content and an exception is thrown.
- C. The file green.txt is moved to the /colors directory.
- D. A FileAlreadyExistsException is thrown at runtime.

**Answer: D**

**NEW QUESTION 34**

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 38**

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 41**

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 44

Given the code fragment:

```
//line n1
System.out.println(iP);
```

Which code fragment, when inserted at line n1, enables the code to print /First.txt?

- A. Path iP = new Paths ("/First.txt");
- B. Path iP = Paths.toPath ("/First.txt");
- C. Path iP = new Path ("/First.txt");
- D. Path iP = Paths.get ("/", "First.txt");

**Answer:** D

#### NEW QUESTION 48

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 52

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.writeIn();
- C. br.flush();
- D. bw.flush();

**Answer:** D

#### NEW QUESTION 55

Given:

```
interface Interfacel {
    public default void sayHi() {
        System.out.println("Hi Interface-1");
    }
}

interface Interface2 {
    public default void sayHi() {
        System.out.println("Hi Interface-2");
    }
}

public class MyClass implements Interfacel, Interface2 {
    public static void main(String[] args) {
        Interfacel obj = new MyClass();
        obj.sayHi();
    }
    public void sayHi() {
        System.out.println("Hi MyClass");
    }
}
```

What is the result?

- A. Hi Interface-2
- B. A compilation error occurs.
- C. Hi Interface-1
- D. Hi MyClass

**Answer:** D

#### NEW QUESTION 57

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 59

Given the code fragment: `Stream<List<String>> iStr= Stream.of ( Arrays.asList ("1", "John"), Arrays.asList ("2", null)0; Stream<<String> nlnSt = iStr.flatMapToInt ((x) -> x.stream ()); nlnSt.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 62

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))  
.f orEach(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 64

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 68

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 73

Given the code fragment:  
`List<Integer> nums = Arrays.asList (10, 20, 8); System.out.println (  
//line n1  
);`



Which code fragment must be inserted at line n1 to enable the code to print the maximum number in the nums list?

- A. `nums.stream().max(Comparator.comparing(a -> a)).get()`
- B. `nums.stream().max(Integer : : max).get()`
- C. `nums.stream().max()`
- D. `nums.stream().map(a -> a).max()`

**Answer:** A

#### NEW QUESTION 77

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 82

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 85

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 86

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 87

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 89

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 91**

Given the code fragment:

```
List<String> qwords = Arrays.asList("why ", "what ", "when ");
BinaryOperator<String> operator = (s1, s2) -> s1.concat(s2); // line n1
String sen = qwords.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 92**

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 96**

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 ())

**Answer: B**

#### NEW QUESTION 98

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 103

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 104**

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 108**

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 113**

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()  
.forEach(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 114**

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 115**

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 118

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 121

Given the code fragments :

```
public class Product {  
    String name;  
    Integer price;  
    Product(String name, Integer price) {  
        this.name = name;  
        this.price = price;  
    }  
    public void printVal(){ System.out.print(name + " Price:" + price + " "); }  
    public void setPrice(int price) { this.price = price; }  
    public Integer getPrice() { return price; }  
}
```

and

```
List<Product> li = Arrays.asList(new Product("TV", 1000), new Product("Refrigerator",  
2000));  
Consumer<Product> raise = e -> e.setPrice(e.getPrice() + 100);  
li.forEach(raise);  
li.stream().forEach(Product::printVal);
```

What is the result?

- A. TV Price :110 Refrigerator Price :2100
- B. A compilation error occurs.
- C. TV Price :1000 Refrigerator Price :2000
- D. The program prints nothing.

**Answer:** C

#### NEW QUESTION 126

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 127

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 128

Given:

```
class UserException extends Exception { }
```

```
class AgeOutOfLimitException extends UserException { }
```

 and the code fragment:

```
class App {
```

```
    public void doRegister(String name, int age) throws UserException, AgeOutOfLimitException { if (name.length () < 6) {
        throw new UserException ();
```

```
    } else if (age >= 60) {
```

```
        throw new AgeOutOfLimitException ();
```

```
    } else {
```

```
        System.out.println("User is registered.");
```

```
    }
```

```
}
```

```
public static void main(String[] args) throws UserException { App t = new App ();
```

- A. t.doRegister("Mathew", 60);}
- B. User is registered.
- C. An AgeOutOfLimitException is thrown.
- D. A UserException is thrown.
- E. A compilation error occurs in the main method.

**Answer:** B

#### NEW QUESTION 131

Which statement is true about the single abstract method of the java.util.function.Function interface?

- A. It accepts one argument and returns void.
- B. It accepts one argument and returns boolean.
- C. It accepts one argument and always produces a result of the same type as the argument.
- D. It accepts an argument and produces a result of any data type.

**Answer:** D

#### NEW QUESTION 134

Given the code fragment:

```
List<String> str = Arrays.asList ("my", "pen", "is", "your", "pen"); Predicate<String> test = s -> {
```

```
    int i = 0;
```

```
    boolean result = s.contains ("pen");
```

```
    System.out.print(i++) + ":"; return result;
```

```
};
```

```
str.stream()
```

```
    .filter(test)
```

```
    .findFirst()
```

```
    .ifPresent(System.out ::print);
```

 What is the result?



- A. 0 : 0 : pen
- B. 0 : 1 : pen
- C. 0 : 0 : 0 : 0 : 0 : pen
- D. 0 : 1 : 2 : 3 : 4 :
- E. A compilation error occurs.

**Answer:** A

#### NEW QUESTION 136

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 137

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 139

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 143

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 144

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 146

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 150

Given the code fragment:

```
List<Integer> li = Arrays.asList(10, 20, 30);  
Function<Integer, Integer> fn = f1 -> f1 + f1;  
Consumer<Integer> conVal = s -> System.out.print("Val:" + s + " ");  
li.stream().map(fn).forEach(conVal);
```

What is the result?

- A. Val:20 Val:40 Val:60
- B. Val:10 Val:20 Val:30
- C. A compilation error occurs.
- D. Val: Val: Val:

**Answer: B**

#### NEW QUESTION 154

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 155

.....

## Thank You for Trying Our Product

### We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

### 1Z0-809 Practice Exam Features:

- \* 1Z0-809 Questions and Answers Updated Frequently
- \* 1Z0-809 Practice Questions Verified by Expert Senior Certified Staff
- \* 1Z0-809 Most Realistic Questions that Guarantee you a Pass on Your First Try
- \* 1Z0-809 Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

**100% Actual & Verified — Instant Download, Please Click**  
**[Order The 1Z0-809 Practice Test Here](#)**