

# 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 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 3**

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 4

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 5

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 6

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 7

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 8

Given the definition of the Emp class: public class Emp

```
private String eName; private Integer eAge;  
Emp(String eN, Integer eA) { this.eName = eN;  
this.eAge = eA;  
}  
public Integer getEAge () {return eAge;} public String getEName () {return eName;}  
}
```

and code fragment:

```
List<Emp>li = Arrays.asList(new Emp("Sam", 20), New Emp("John", 60), New Emp ("Jim", 51));  
Predicate<Emp> agVal = s -> s.getEAge() > 50; //line n1 li = li.stream().filter(agVal).collect(Collectors.toList());  
Stream<String> names = li.stream().map.(Emp::getEName); //line n2 names.forEach(n -> System.out.print(n + " "));  
What is the result?
```

- A. Sam John Jim
- B. John Jim
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

**Answer:** B

#### NEW QUESTION 9

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 10

Given the code fragments:  
interface CourseFilter extends Predicate<String> { public default boolean test (String str) {  
return str.equals ("Java");  
}  
}  
and  
List<String> strs = Arrays.asList("Java", "Java EE", "Java ME"); Predicate<String> cf1 = s -> s.length() > 3;  
Predicate cf2 = new CourseFilter() { //line n1 public boolean test (String s) {  
return s.contains ("Java");  
}  
};  
long c = strs.stream()  
.filter(cf1)  
.f ilter(cf2 //line n2  
.count(); System.out.println(c); What is the result?

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

**Answer:** B

#### NEW QUESTION 10

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 15

Given the code fragments:  
4. void doStuff() throws ArithmeticException, NumberFormatException, Exception  
{  
5. if (Math.random() >-1 throw new Exception ("Try again"); 6. }  
and  
24. try {  
25. doStuff ( );  
26. } catch (ArithmeticException | NumberFormatException | Exception e) {  
27. System.out.println (e.getMessage()); }  
28. catch (Exception e) {  
29. System.out.println (e.getMessage()); }  
30. }



Which modification enables the code to print Try again?

- A. Comment the lines 28, 29 and 30.
- B. Replace line 26 with:} catch (Exception | ArithmeticException | NumberFormatException e) {
- C. Replace line 26 with:} catch (ArithmeticException | NumberFormatException e) {
- D. Replace line 27 with: throw e;

**Answer: C**

#### NEW QUESTION 19

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 24

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 28

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

Given the code fragment:

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

Which is the result?

- A. 3:00 – difference: 2
- B. 2:00 – difference: 1

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

**Answer: B**

#### NEW QUESTION 31

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 33

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 37

Given:

```
public class Canvas implements Drawable { public void draw () { }  
}  
public abstract class Board extends Canvas { }  
public class Paper extends Canvas { protected void draw (int color) { }  
}  
public class Frame extends Canvas implements Drawable { public void resize () { }  
}  
public interface Drawable { public abstract void draw ();  
}
```

Which statement is true?

- A. Board does not compile.
- B. Paper does not compile.
- C. Frame does not compile.
- D. Drawable does not compile.
- E. All classes compile successfully.

**Answer:** E

### NEW QUESTION 40

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 42

Given:

```
interface Rideable {Car getCar (String name); } class Car {  
    private String name; public Car (String name) { this.name = name;  
    }  
}
```

Which code fragment creates an instance of Car?

- A. Car auto = Car ("MyCar"): : new;
- B. Car auto = Car : : new;Car vehicle = auto : : getCar("MyCar");
- C. Rideable rider = Car : : new;Car vehicle = rider.getCar("MyCar");
- D. Car vehicle = Rideable : : new : : getCar("MyCar");

**Answer:** C



**NEW QUESTION 45**

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 47**

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 ()
```

```
.collect(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 51**

Given the code fragment:

```
public static void main(String[] args) {  
    Console console = System.console();  
    char[] pass = console.readPassword("Enter password:"); // line n1  
    String password = new String(pass); // line n2  
}
```

What is the result?

- A. A compilation error occurs at line n1.
- B. A compilation error occurs at line n2.
- C. The code reads the password without echoing characters on the console.
- D. A compilation error occurs because the IOException isn't declared to be thrown or caught?

**Answer: D**

**NEW QUESTION 54**

Assume customers.txt is accessible and contains multiple lines. Which code fragment prints the contents of the customers.txt file?

- A. `Stream<String> stream = Files.find (Paths.get ("customers.txt")); stream.forEach((String c) -> System.out.println(c));`
- B. `Stream<Path> stream = Files.find (Paths.get ("customers.txt")); stream.forEach( c) -> System.out.println(c));`
- C. `Stream<Path> stream = Files.list (Paths.get ("customers.txt")); stream.forEach( c) -> System.out.println(c));`
- D. `Stream<String> lines = Files.lines (Paths.get ("customers.txt")); lines.forEach( c) -> System.out.println(c));`

**Answer: A**

**NEW QUESTION 59**

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 63

Given:

```
interface Doable {  
    public void doSomething (String s);  
}
```

Which two class definitions compile? (Choose two.)

- A. public abstract class Task implements Doable { public void doSomethingElse(String s) { }}
- B. public abstract class Work implements Doable { public abstract void doSomething(String s) { } public void doYourThing(Boolean b) { }}
- C. public class Job implements Doable { public void doSomething(Integer i) { }}
- D. public class Action implements Doable { public void doSomething(Integer i) { } public String doThis(Integer j) { }}
- E. public class Do implements Doable { public void doSomething(Integer i) { } public void doSomething(String s) { } public void doThat (String s) { }}

**Answer:** AE

#### NEW QUESTION 64

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 69

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 71

Given the code fragments: class TechName {

```
    String techName;  
    TechName (String techName) { this.techName=techName;  
    }  
}
```

and

```
List<TechName> tech = Arrays.asList ( new TechName("Java-"),  
    new TechName("Oracle DB-"), new TechName("J2EE-")  
);  
Stream<TechName> stre = tech.stream();  
//line n1
```

Which should be inserted at line n1 to print Java-Oracle DB-J2EE-?

- A. stre.forEach(System.out::print);
- B. stre.map(a-> a.techName).forEach(System.out::print);
- C. stre.map(a-> a).forEachOrdered(System.out::print);
- D. stre.forEachOrdered(System.out::print);

**Answer:** B

#### NEW QUESTION 76

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 77

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 82

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 84

Given that version.txt is accessible and contains: 1234567890  
and given the code fragment:

```
try (FileInputStream fis = new FileInputStream("version.txt");
    InputStreamReader isr = new InputStreamReader(fis);
    BufferedReader br = new BufferedReader(isr);) {
    if (br.markSupported()) {
        System.out.print((char) br.read());
        br.mark(2);
        System.out.print((char) br.read());
        br.reset();
        System.out.print((char) br.read());
    }
} catch (Exception e) {
    e.printStackTrace();
}
```

What is the result?

- A. 121
- B. 122
- C. 135
- D. The program prints nothing.

**Answer:** B

#### NEW QUESTION 86

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 90

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 94**

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 96**

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);}}What is the result?
- 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 98**

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 101**

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).reversed()).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).reversed

**Answer:** A

**NEW QUESTION 106**

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 109

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 111

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 116

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 120

.....

## 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](#)**