DUMPSARENA

Java SE 8 Programmer II

Oracle 1z0-809

Version Demo

Total Demo Questions: 15

Total Premium Questions: 207

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

QUESTION NO: 1

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: C E

QUESTION NO: 2

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

QUESTION NO: 3

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: C D

QUESTION NO: 4

Given:

```
public class Vehicle {
    int vId;
    String vName;
    public Vehicle(int vIdArg, String vNameArg) {
        this.vId = vIdArg;
        this.vName = vNameArg;
    }
    public int getVId() { return vId; }
    public String getVName() { return vName; }
    public String toString() {
        return vName;
    }
}
```

and the code fragment:

```
List<Vehicle> vehicle = Arrays.asList(
        new Vehicle(2, "Car"),
        new Vehicle(3, "Bike"),
        new Vehicle(1, "Truck"));
vehicle.stream()
        // line n1
        .forEach(System.out::print);
```

Which two code fragments, when inserted at line n1 independently, enable the code to print TruckCarBike?

A. .sorted ((v1, v2) -> v1.getVId() < v2.getVId())

B. .sorted (Comparable.comparing (Vehicle: :getVName)).reversed ()

C. .map (v -> v.getVid())
.sorted ()

D. .sorted((v1, v2) -> Integer.compare(v1.getVId(), v2.getVid()))

E. .sorted(Comparator.comparing ((Vehicle v) -> v.getVId()))

ANSWER: D E

QUESTION NO: 5

Given the code fragment:

List list1 = Arrays.asList(10, 20);

List 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

QUESTION NO: 6

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 public.

C. Remove the new keyword from the instantion of usCoin.

D. Make the getValue() method public.

E. Add the final keyword in the declaration of value.

ANSWER: B C

QUESTION NO: 7

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(scr));
```

What is the result?

A. [Java EE: Helen:Houston]
[Java ME: Jessy:Chicago, Java ME: Mark:Chicago]

B. Java EE
Java ME

C. [Java ME: Jessy:Chicago, Java ME: Mark:Chicago]
[Java EE: Helen:Houston]

D. A compilation error occurs.

ANSWER: D

Explanation:

Your Code ...

```
1  public class Student {
2      String course, name, city;
3      public Student (String name, String course, String cit
4          this.course = course; this.name = name; this.city
5      }
6      public String toString() {
7          return course + ":" + name + ":" + city;
8      }
9      public String getCourse() {return course; }
10     public String getName() {return name; }
11     public String getCity() {return city; }
12
13     List<Student> stds = Arrays.asList (
14         new Student ("Jessy", "Java ME", "Chicago"),
15         new Student ("Helen", "Java ME", "Houston"),
16         new Student ("Mark", "Java ME", "Chicago"));
17     stds.stream()
18         .collect (Collectors.groupBy(Student::getCourse))
19         .forEach (src, res) -> System.out.println(scr));
20 }
21
```

CommandLine Arguments ...

Stdin Inputs...

Execute Sa

Result...

CPU Time: sec(s), Memory: kilobyte(s)

```
/Student.java:17: error: <identifier> expected
  stds.stream()
             ^
/Student.java:17: error: ';' expected
  stds.stream()
             ^
2 errors
```

QUESTION NO: 8

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

QUESTION NO: 9

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: A E

Explanation:

Reference: http://docs.oracle.com/javase/7/docs/technotes/guides/intl/

QUESTION NO: 10

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 or
Java SEJava EE

D. Java EEJava SE

ANSWER: D

QUESTION NO: 11

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

QUESTION NO: 12

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 FuelNotAvailException { //line n2 super ride ();

}

}

and the code fragment:

public static void main (String[] args) throws 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 public void ride()throws FuelNotAvailException, Exception {

D. Replace line n2 with private void ride() throws FuelNotAvailException {

ANSWER: C

QUESTION NO: 13

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

QUESTION NO: 14

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()) {

f.open();

}

}

}

Which two modifications enable the code to print Open Close? (Choose two.)

A. Replace line n1 with: class Folder implements AutoCloseable {

B. Replace line n1 with: class Folder extends Closeable {

C. Replace line n1 with: class Folder extends Exception {

D. At line n2, insert:
final void close () {
System.out.print(“Close”);
}

E. At line n2, insert:
public void close () throws IOException {
System.out.print(“Close”);
}

ANSWER: A E

QUESTION NO: 15

Which two statements are true about the Fork/Join Framework? (Choose two.)

A. The RecursiveTask subclass is used when a task does not need to return a result.

B. The Fork/Join framework can help you take advantage of multicore hardware.

C. The Fork/Join framework implements a work-stealing algorithm.

D. The Fork/Join solution when run on multicore hardware always performs faster than standard sequential solution.

ANSWER: A C

Explanation:

Reference: https://www.logicbig.com/tutorials/core-java-tutorial/java-multi-threading/fork-and-join.html