DUMPSARENA

Java SE 11 Developer

Oracle 1z0-819

Version Demo

Total Demo Questions: 15

Total Premium Questions: 248

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

QUESTION NO: 1

Given:

```
public class Foo {
    private void print() {
        System.out.println("Bonjour le monde!");
    }
    public void foo() {
        print();
    }
}

public class Bar extends Foo {
    private void print() {
        System.out.println("Hello world!");
    }
    public void bar() {
        print();
    }
    public static void main(String... args) {
        Bar b = new Bar();
        b.foo();
        b.bar();
    }
}
```

What is the output?

A. Hello world!Bonjour le monde!

B. Hello world!Hello world!

C. Bonjour le monde!Hello world!

D. Bonjour le monde!Bonjour le monde!

ANSWER: C

QUESTION NO: 2

Given:

```
public interface Builder {
    public A build(String str);
}

and

public class BuilderImpl implements Builder {
    @Override
    public B build(String str) {
        return new B(str);
    }
}
```

Assuming that this code compiles correctly, which three statements are true? (Choose three.)

A. B cannot be abstract.

B. B is a subtype of A.

C. A cannot be abstract.

D. A cannot be final.

E. B cannot be final.

F. A is a subtype of B.

ANSWER: A B D

QUESTION NO: 3

Given these declarations:

```
String eName = "SMITH";
String empId = "42";
```

and these two code fragments:

Fragment 1:

```
Statement stmt = conn.createStatement();
String sql = "INSERT INTO EMP VALUES ('" + eName + "', '" + empId + "')";
stmt.executeUpdate(sql);
```

Fragment 2:

```
String sql = "INSERT INTO EMP VALUES (?, ?)";
PreparedStatement pStmt = conn.prepareStatement(sql);
pStmt.setObject(1, eName, JDBCType.VARCHAR);
pStmt.setObject(2, empId, JDBCType.VARCHAR);
pStmt.executeUpdate();
```

Which code fragment is preferred and why?

A. Fragment 1 because it is shorter.

B. Fragment 2 because it prevents SQL injection.

C. Fragment 2 because it explicitly specifies the SQL types of the column values.

D. Fragment 1 because it is more performant.

ANSWER: B

QUESTION NO: 4

Given the code fragment:

```
public class Main {
  public static void main(String[] args) {
    try {
      Path path = Paths.get("/u01/work");
      // line 1
      System.out.println(attributes.isDirectory());
    } catch (IOException e) {
      e.printStackTrace();
    }
  }
}
```

You want to examine whether path is a directory. Which code inserted on line 1 will accomplish this?

A. BasicFileAttributes attributes = Files isDirectory (path);

B. BasicFileAttributes attributes =Files.getAttribute (path, ‘’insdirectory’’);

C. BasicFileAttributes attributes = Files.readAttributes(path, BasicFileAttributes.class

D. BasicFileAttributes attributes = Files, readAttributes (path, FileAttributes, class);

ANSWER: D

QUESTION NO: 5

Given:

```
interface Abacus{
  public int calc (int a, int b);
}

public class Main {
  public static void main (String[] args) {
    int result = 0;
    // line 1
    result = aba.calc(10, 20);
    System.out.println(result);
  }
}
```

Which two codes, independently, can be inserted in line to 1 compile?

A. Abacus aba = (int m, int n) -> { m * n };

B. Abacus aba = (int e, int f) -> { return e * f; };

C. Abacus aba = (a, b) -> a * b;

D. Abacus aba = v, w -> x * y;

E. Abacus aba = (int i, j) -> (return i * j; };

ANSWER: C E

QUESTION NO: 6

Which two statements independently compile? (Choose two.)

A. List list = new ArrayList();

B. List list = new ArrayList();

C. List list = new ArrayList();

D. List list = new ArrayList();

E. List list = new ArrayList();

ANSWER: A C

QUESTION NO: 7

Given:

```
public class SerializedMessage implements Serializable {
   String message;
   LocalDateTime createdTime;
   transient LocalDateTime updatedDateTime;;
   SerializedMessage(String message) {
      this.message = message;
      this.createdTime = LocalDateTime.now();
   }
   private void readObject (ObjectInputStream in) {
      try {
         in.defaultReadObject();
         this.updatedDateTime = LocalDateTime.now();
      } catch (IOException |ClassNotFoundException e) {
        e.printStackTrace();
      }
   }
}
```

When is the readObject method called?

A. before this object is deserialized

B. after this object is deserialized

C. before this object Is serialized

D. The method is never called.

E. after this object is serialized

ANSWER: B

QUESTION NO: 8

Given:

```
public class X {
  private Collection collection;
  public void set(Collection collection) {
    this.collection = collection;
  }
}

and

public class Y extends X {
  public void set(Map<String,String> map) {
    super.set(map); // line 1
  }
}
```

Which two lines can replace line 1 so that the Y class compiles? (Choose two.)

A. map.forEach((k, v) -> set(v)));

B. set(map.values());

C. super.set(List map)

D. super.set(map.values());

E. set(map)

ANSWER: B D

QUESTION NO: 9

Given:

```
Integer[] intArray = {2, 1, 3, 4, 5};
List<Integer> list =
new ArrayList<>(Arrays.asList (intArray));
list.parallelStream()
   .forEach(e -> System.out.print(e + " "));
```

Which two are correct? (Choose two.)

A. The output will be exactly 2 1 3 4 5.

B. The program prints 1 4 2 3, but the order is unpredictable.

C. Replacing forEach() with forEachOrdered(), the program prints 2 1 3 4 5, but the order is unpredictable.

D. Replacing forEach() with forEachOrdered(), the program prints 1 2 3 4 5.

E. Replacing forEach() with forEachOrdered(), the program prints 2 1 3 4 5.

ANSWER: B D

QUESTION NO: 10

Given:

```
public static void main(String[] args) {
   try (Reader reader1 = new FileReader("File1.txt");
        Reader reader2 = new FileReader("File2.txt");
        Reader reader3 = new FileReader("File3_txt")) {

   } catch (IOException ex) {
      Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
   }
   // Line 1
   System.out.println("Done");
}
```

When run and all three files exist, what is the state of each reader on Line 1?

A. All three readers are still open.

B. All three readers have been closed.

C. The compilation fails.

D. Only reader1 has been closed.

ANSWER: C

QUESTION NO: 11

Which code fragment compiles?

A.
```
Comparator comparator = new Comparator<?>() {
  public int compare(Integer i, Integer j) {
    return i.compareTo(j);
  }
};
```

B.
```
var comparator = new Comparator<>() {
  public int compare(Integer i, Integer j) {
    return i.compareTo(j);
  }
};
```

C.
```
Comparator<> comparator = new Comparator<Integer>() {
  public int compare(Integer i, Integer j) {
    return i.compareTo(j);
  }
};
```

D.
```
Comparator<Integer> comparator = new Comparator<>() {
  public int compare(Integer i, Integer j) {
    return i.compareTo(j);
  }
};
```

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: D

QUESTION NO: 12

Given:

```
public class DNASynth {
    int aCount;
    int tCount;
    int cCount;
    int gCount;

    int getACount(int aCount){
        return aCount;
    }
    int getTCount(int tCount){
        return this.tCount;
    }
    int getCCount(){
        return getTotalCount() - this.aCount - getTCount(0) - gCount;
    }
    int getGCount(){
        return getGCount();
    }
    int getTotalCount(){
        return aCount + getTCount(0) + this.cCount + this.gCount;
    }
}
```

Which two methods facilitate valid ways to read instance fields? (Choose two.)

A. getTCount

B. getACount

C. getTotalCount

D. getCCount

E. getGCount

ANSWER: C D

QUESTION NO: 13

Which interface in the java.util.function package will return a void return type?

A. Supplier

B. Predicate

C. Function

D. Consumer

ANSWER: D

QUESTION NO: 14

Which two are functional interfaces? (Choose two.)

A.
```
@FunctionalInterface
interface MyRunnable {
   public void run();
}
```

B.
```
@FunctionalInterface
interface MyRunnable {
   public void run();
   public void call();
}
```

C.
```
interface MyRunnable {
   public default void run() {}
   public void run(String s);
}
```

D.
```
@FunctionalInterface
interface MyRunnable {
}
```

E.
```
interface MyRunnable {
   @FunctionalInterface
   public void run();
}
```

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

ANSWER: C E

QUESTION NO: 15

Given:

```
public interface EulerInterface {
    double getEulerValue();
}

public class EulerLambda {
    public static void main(String[] args) {
        EulerInterface  myEulerInterface;
        myEulerInterface = () -> "2.71828";
        System.out.println("Value of Euler = " + myEulerInterface.getEulerValue());
    }
}
```

What is the result?

A. It throws a runtime exception.

B. Value of Euler = 2.71828

C. The code does not compile.

D. Value of Euler = “2.71828”

ANSWER: C