DUMPSARENA

Java SE 11 Programmer II

Oracle 1z0-816

Version Demo

Total Demo Questions: 10

Total Premium Questions: 80

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

QUESTION NO: 1

Assuming the Widget class has a getPrice method, this code does not compile:

```
List widgets = List.of(new Widget("Basic Widget", 19.55), // line 1
                       new Widget("Enhanced Widget", 35.00),
                       new Widget("Luxury Edition Widget", 55.45));
Stream widgetStream = widgets.stream();                    // line 4
widgetStream.filter(a -> a.getPrice() > 20.00)             // line 5
            .forEach(System.out::println);
```

Which two statements, independently, would allow this code to compile? (Choose two.)

A. Replace line 5 with widgetStream.filter(a −> ((Widget)a).getPrice() > 20.00).

B. Replace line 1 with List widgetStream = widgets.stream();.

C. Replace line 5 with widgetStream.filter((Widget a) −> a.getPrice() > 20.00).

D. Replace line 4 with Stream widgetStream = widgets.stream();.

ANSWER: A D

QUESTION NO: 2

Given the declaration:

```
@interface Resource {
   String name();
   int priority() default 0;
}
```

Examine this code fragment:

/* Loc1 */ class ProcessOrders { ... }

Which two annotations may be applied at Loc1 in the code fragment? (Choose two.)

A. @Resource(priority=100)

B. @Resource(priority=0)

C. @Resource(name="Customer1", priority=100)

D. @Resource(name="Customer1")

E. @Resource

ANSWER: A B

QUESTION NO: 3

Given:

```
public class Tester {
   static class Person implements /* line 1 */ {
      private String name;
      Person(String name) { this.name = name; }
      /* line 2 */
   }
   public static void main(String[] args) {
      Person[] people = {new Person("Joe"),
                         new Person("Jane"),
                         new Person("John")};
      Arrays-sort(people);
      for(Person person: people) {
         System.out.println(person.name);
      }
   }
}
```

You want the code to produce this output:

John

Joe

Jane

Which code fragment should be inserted on line 1 and line 2 to produce the output?

A. Insert Comparator on line 1. Insert
public int compare(Person p1, Person p2) { return p1.name.compare(p2.name);
} on line 2.

B. Insert Comparator on line 1. Insert
public int compareTo(Person person) { return person.name.compareTo(this.name);
} on line 2.

C. Insert Comparable on line 1. Insert
public int compare(Person p1, Person p2) { return p1.name.compare(p2.name);
} on line 2.

D. Insert Comparator on line 1. Insert
public int compare(Person person) { return person.name.compare(this.name);
} on line 2.

ANSWER: B

Explanation:

Reference: https://www.coursehero.com/file/p320ss6/Override-public-int-compareTo-Person-other-Compare-this-objects-name-to-others/

QUESTION NO: 4

Given an application with a main module that has this module-info.java file:

```
module main {
    exports country;
    uses country.CountryDetails;
}
```

Which two are true? (Choose two.)

A. A module providing an implementation of country.CountryDetails can be compiled and added without recompiling the main module.

B. A module providing an implementation of country.CountryDetails must have a requires main; directive in its module-info.java file.

C. An implementation of country.countryDetails can be added to the main module.

D. To compile without an error, the application must have at least one module in the module source path that provides an implementation of country.CountryDetails.

E. To run without an error, the application must have at least one module in the module path that provides an implementation of country.CountryDetails.

ANSWER: B D

Explanation:

Reference: https://stackoverflow.com/questions/49476559/java-9-error-not-in-a-module-on-the-module-source-path

QUESTION NO: 5

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.

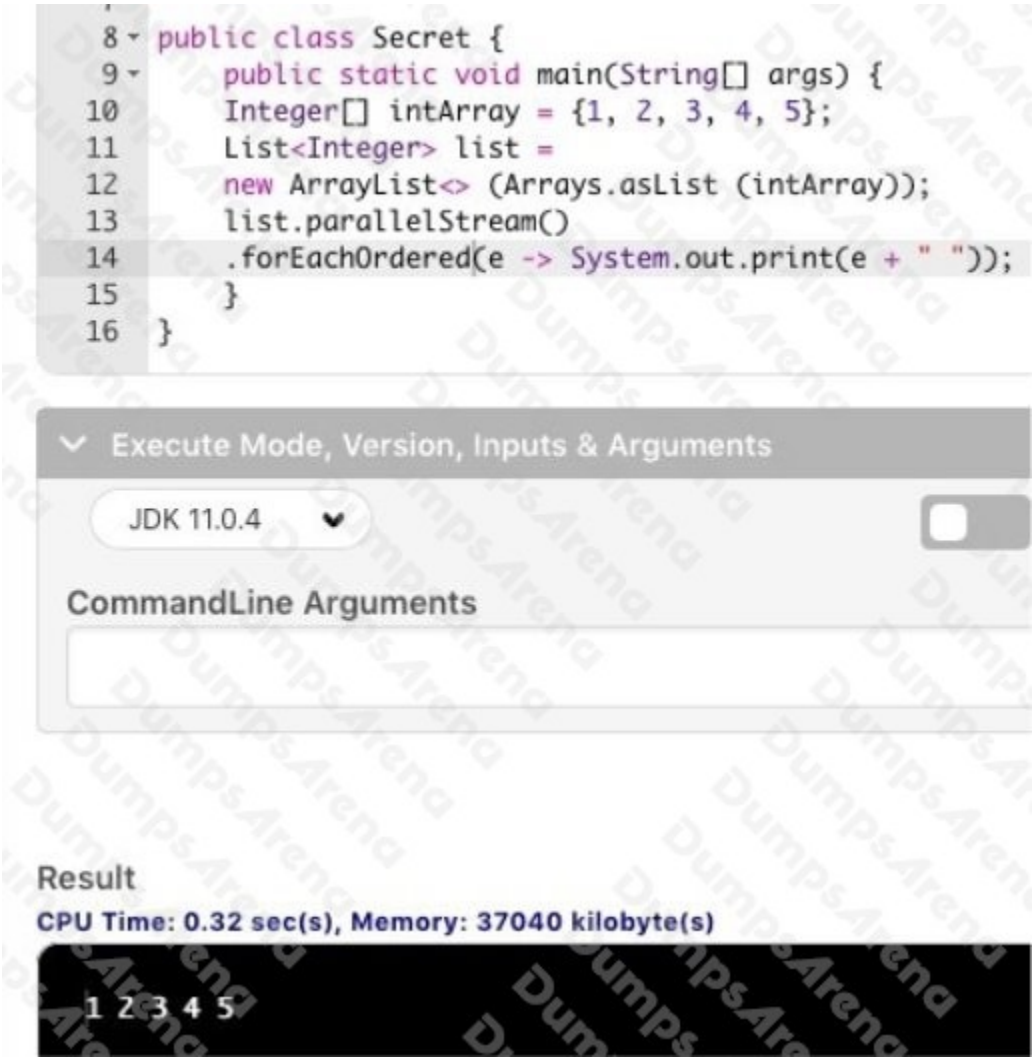
C. Replacing forEach() with forEachOrdered(), the program prints 2 1 3 4 5, but the order is unpredictable.

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

Explanation:



QUESTION NO: 6

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

A. Supplier

B. Predicate

C. Function

D. Consumer

ANSWER: D

Explanation:

Reference: https://www.geeksforgeeks.org/java-8-consumer-interface-in-java-with-examples/

QUESTION NO: 7

Given:

```
public class FunctionalInterfaceTest {
   public static void main(String[] args) {
      List fruits = Arrays.asList("apple", "orange", "banana");
      Consumer<String> c = System.out::print;
      Consumer<String> output = c.andThen(x -> System.out.println(":" + x.toUpperCase
()));
      fruits.forEach(output);
   }
}
```

What is the output?

A. :APPLE:ORANGE:BANANA appleorangebanana

B. :APPLE:ORANGE:BANANA

C. APPLE:apple
ORANGE:orange
BANANA:banana

D. appleorangebanana
:APPLE:ORANGE:BANANA

E. apple:APPLE orange:ORANGE banana:BANANA

ANSWER: E

Explanation:

```java
import java.util.*;
import java.io.*;
import java.lang.Thread;
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;
import java.util.function.Consumer;

public class FunctionalInterfaceTest {
 public static void main (String[] args) {
        List fruits = Arrays.asList("apple", "orange", "banana");
        Consumer<String> c = System.out::print;
        Consumer<String> output = c.andThen(x -> System.out.println(":" + x.toUpperCase()));

   fruits.forEach(output);

  }
}
```

QUESTION NO: 8

Given:

```java
@Target(ElementType.METHOD)
@Retention(RetentionPolicy.RUNTIME)
public @interface AuthorInfo {
    String author() default "";
    String date();
    String[] comments() default {};
}
```

Which two are correct? (Choose two.)

A.
```
@AuthorInfo(date="1-1-2020", comments={ null })
public class Hello {
     public void func() {}
}
```

B.
```
public class Hello {
@AuthorInfo (date="1-1-2020. comments="Hello")
     public void func() {}
}
```

C.
```
public class Hello {
     @AuthorInfo
     public void func() {}
}
```

D.
```
@AuthorInfo(date="1-1-2020")
public class Hello {
     public void func() {}
}
```

E.
```
public class Hello {
  @AuthorInfo(date="1-1-2020", author="Gandhi", comments={ "world" })
  public void func () {}
}
```

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

ANSWER: C D

QUESTION NO: 9

Given:

```
import java.util.List;
import java.util.function.BinaryOperator;
public class Main {
   public static void main(String... args) {
      List<Employee> list = List.of(new Employee("John", 80000.0), new Employee("Scott",
90000.0));
      double starts = 0.0;
      double ratio = 1.0;
      BinaryOperator<Double> bo = (a, b) -> a + b;
double totalSalary = list.stream().map(e -> e.getSalary() * ratio).reduce(starts, bo);
// line 1
      System.out.println("Total salary = " + totalSalary);
   }
}

class Employee {
   String name;
   double salary;
   public Employee(String name, double salary) {
      this.name = name;
      this.salary = salary;
   }
   public String getName() { return name; }
   public double getSalary{} { return salary; }
}
```

Which statement is equivalent to line 1?

A. double totalSalary = list.stream().map(e −> e.getSalary() * ratio).reduce(bo).ifPresent (p −> p.doubleValue());

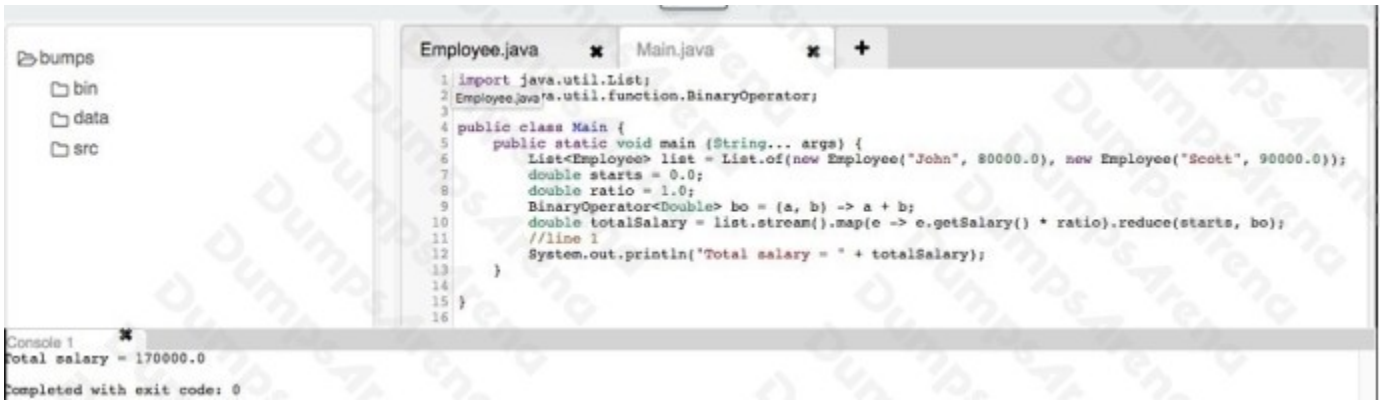
B. double totalSalary = list.stream().mapToDouble(e −> e.getSalary() * ratio).sum;

C. double totalSalary = list.stream().map(Employee::getSalary * ratio).reduce(bo).orElse(0.0);

D. double totalSalary = list.stream().mapToDouble(e −> e.getSalary() * ratio).reduce(starts, bo);

ANSWER: C

Explanation:



QUESTION NO: 10

Given:

```
public class Foo {
   private final ReentrantLock lock = new ReentrantLock();
   private State state;
   public void foo() throws Exception {
      try {
         lock.lock();
         state.mutate();
      }
      finally {
         lock.unlock();
      }
   }
}
```

What is required to make the Foo class thread safe?

A. No change is required.

B. Make the declaration of lock static.

C. Replace the lock constructor call with new ReentrantLock (true).

D. Move the declaration of lock inside the foo method.

ANSWER: C

Explanation:

Reference: https://stackoverflow.com/questions/55134811/how-to-make-java-class-thread-safe