# 1z0-816

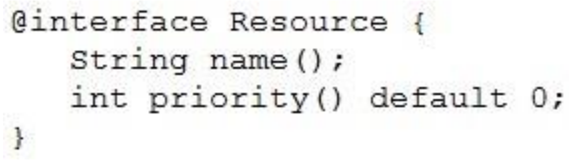
Number: 1z0-816
Passing Score: 800
Time Limit: 120 min

File Version: [1]

[1] z0-816

**Exam A**

**QUESTION 1**
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”)`

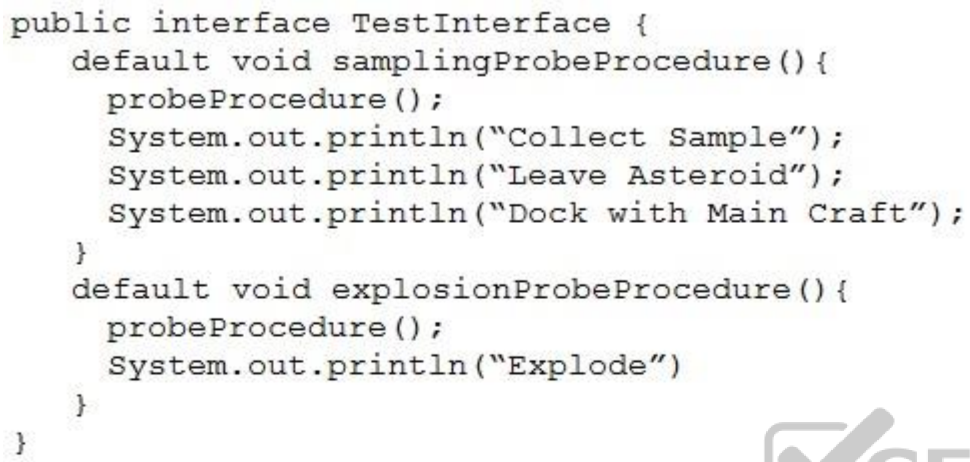
E. `@Resource`

**Correct Answer:** AB
**Section: (none)**
**Explanation**

**Explanation/Reference:**

**QUESTION 2**

Given:

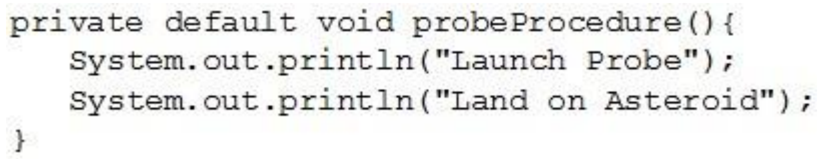


```
public interface TestInterface {
    default void samplingProbeProcedure(){
      probeProcedure();
      System.out.println(“Collect Sample”);
      System.out.println(“Leave Asteroid”);
      System.out.println(“Dock with Main Craft”);
    }
    default void explosionProbeProcedure(){
      probeProcedure();
      System.out.println(“Explode”)
    }
}
```

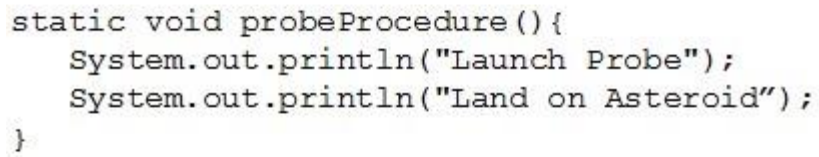
Examine these requirements:

- Eliminate code duplication.
- Keep constant the number of methods other classes may implement from this interface.

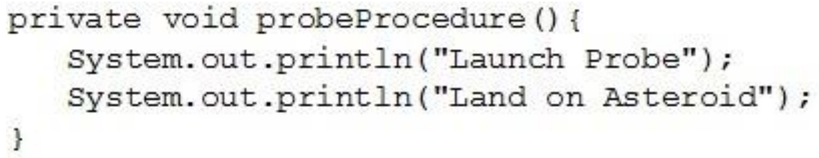
Which method can be added to meet these requirements?



```
private default void probeProcedure(){
    System.out.println("Launch Probe");
    System.out.println("Land on Asteroid");
}
```



```
static void probeProcedure(){
    System.out.println("Launch Probe");
    System.out.println("Land on Asteroid”);
}
```

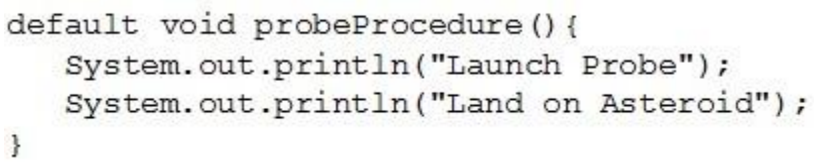


```
private void probeProcedure(){
    System.out.println("Launch Probe");
    System.out.println("Land on Asteroid");
}
```

A.

B.

C.



```
default void probeProcedure(){
    System.out.println("Launch Probe");
    System.out.println("Land on Asteroid");
}
```

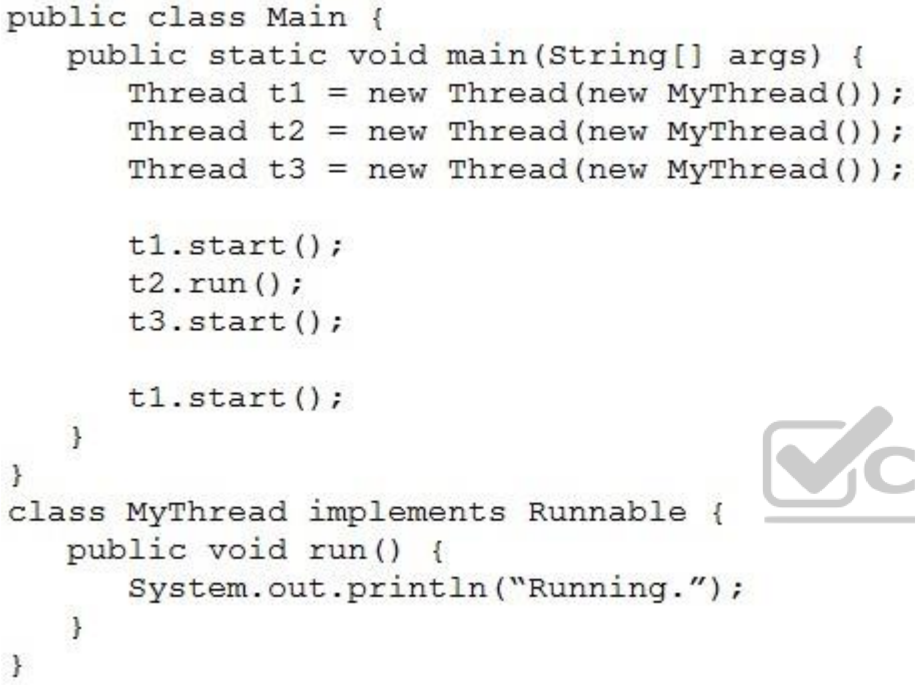
D.

**Correct Answer:** B
**Section: (none)**
**Explanation**

**Explanation/Reference:**

**QUESTION 3**

Given:



```
public class Main {
    public static void main(String[] args) {
        Thread t1 = new Thread(new MyThread());
        Thread t2 = new Thread(new MyThread());
        Thread t3 = new Thread(new MyThread());

        t1.start();
        t2.run();
        t3.start();

        t1.start();
    }
}
class MyThread implements Runnable {
    public void run() {
        System.out.println(“Running.”);
    }
}
```

Which one is correct?

A. An `IllegalThreadStateException` is thrown at run time.

B. Three threads are created.

C. The compilation fails.

D. Four threads are created.

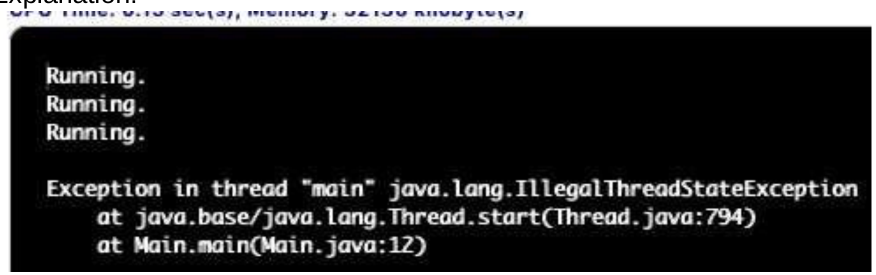
**Correct Answer:** A

**Section: (none)**

**Explanation**

**Explanation/Reference:**

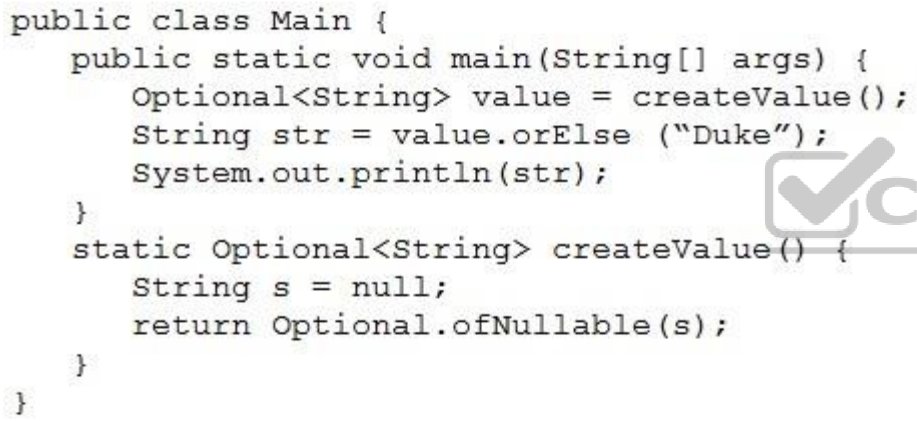
Explanation:



```
CPU Time: 0.15 sec(s), Memory: 32136 kilobyte(s)

Running.
Running.
Running.

Exception in thread "main" java.lang.IllegalThreadStateException
    at java.base/java.lang.Thread.start(Thread.java:794)
    at Main.main(Main.java:12)
```

**QUESTION 4**

Given:



```
public class Main {
    public static void main(String[] args) {
        Optional<String> value = createValue();
        String str = value.orElse ("Duke");
        System.out.println(str);
    }
    static Optional<String> createValue() {
        String s = null;
        return Optional.ofNullable(s);
    }
}
```

What is the output?

A. `null`

B. A `NoSuchElementException` is thrown at run time.

C. `Duke`

D. A `NullPointerException` is thrown at run time.

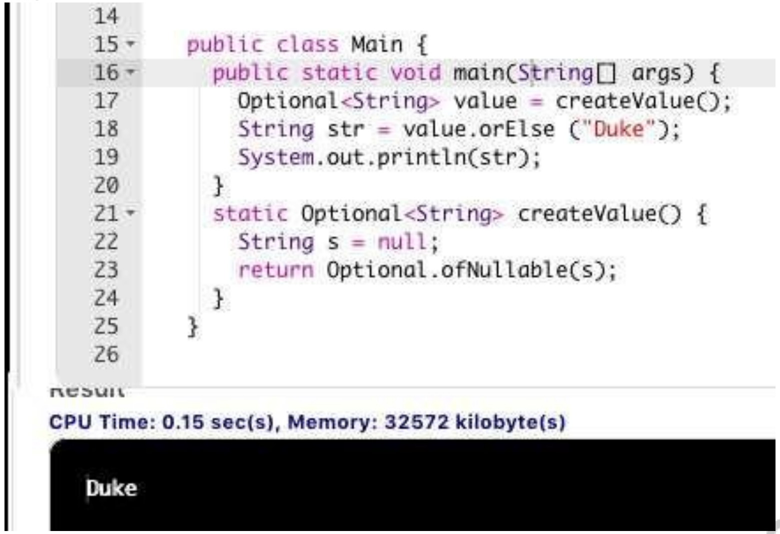
**Correct Answer:** C

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:



```
public class Main {
  public static void main(String[] args) {
    Optional<String> value = createValue();
    String str = value.orElse ("Duke");
    System.out.println(str);
  }
  static Optional<String> createValue() {
    String s = null;
    return Optional.ofNullable(s);
  }
}
```

Result

CPU Time: 0.15 sec(s), Memory: 32572 kilobyte(s)

```
Duke
```

## QUESTION 5

Assume `ds` is a `DataSource` and the EMP table is defined appropriately.

```
try (Connection conn = ds.getConnection();
     PreparedStatement ps = conn.prepareStatement(“INSERT INTO EMP VALUES(?, ?, ?)”)) {
     ps.setObject(1, 101, JDBCType.INTEGER);
     ps.setObject(2, “SMITH”, JDBCType.VARCHAR);
     ps.setObject(3, “HR”, JDBCType.VARCHAR);
     ps.executeUpdate();
     ps.setInt(1, 102);
     ps.setString(2, “JONES”);
     ps.executeUpdate();
}
```

What does executing this code fragment do?

A. inserts two rows (`101, 'SMITH', 'HR'`) and (`102, 'JONES', NULL`)

B. inserts two rows (`101, 'SMITH', 'HR'`) and (`102, 'JONES', 'HR'`)

C. inserts one row (`101, 'SMITH', 'HR'`)

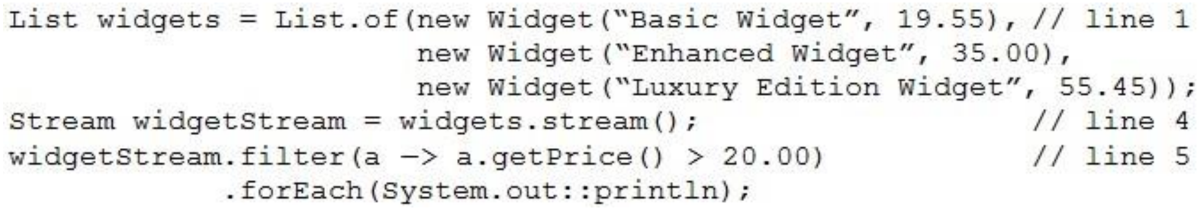
D. throws a `SQLException`

**Correct Answer:** C
**Section: (none)**
**Explanation**

**Explanation/Reference:**

**QUESTION 6**

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<Widget> widgetStream = widgets.stream();.`

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

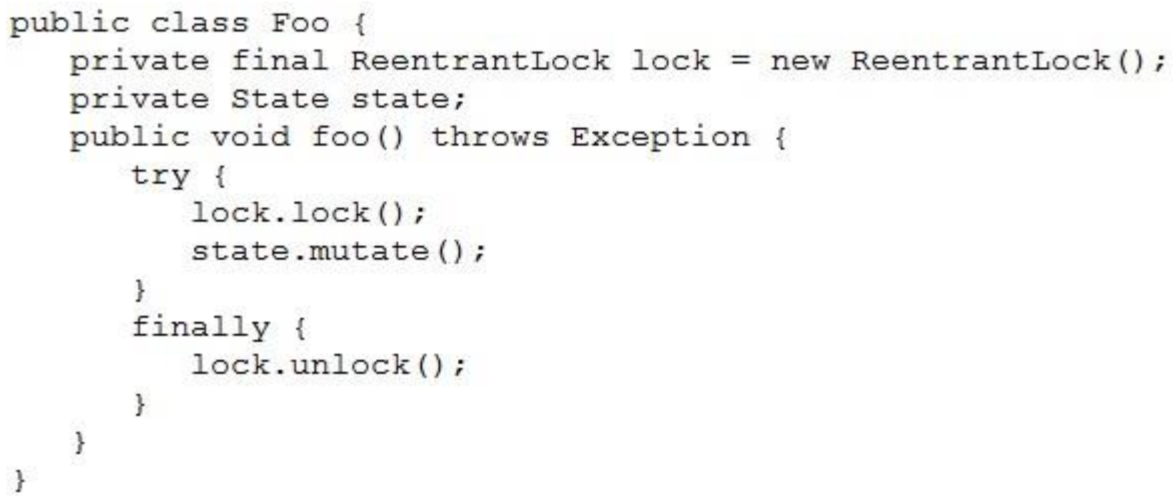
D. Replace line 4 with `Stream<Widget> widgetStream = widgets.stream();.`

**Correct Answer:** AD
**Section: (none)**
**Explanation**

**Explanation/Reference:**

**QUESTION 7**

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.

**Correct Answer:** C

**Section: (none)**

**Explanation**

**Explanation/Reference:**

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

**QUESTION 8**

Given:

```
var data = new ArrayList<>();
data.add("Peter");
data.add(30);
data.add("Market Road");
data.set(1, 25);
data.remove(2); data.set(3,
1000L);
System.out.print(data);
```

What is the output?

A. `[Market Road, 1000]`

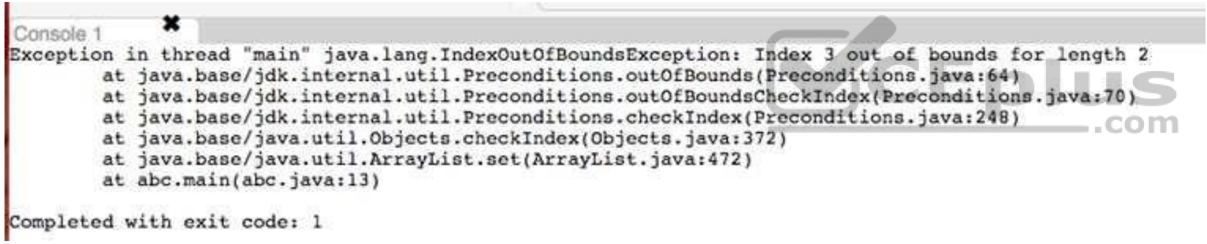
B. `[Peter, 30, Market Road]`

C. `[Peter, 25, null, 1000]`

D. An exception is thrown at run time.

**Correct Answer:** D
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Explanation:



```
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index 3 out of bounds for length 2
        at java.base/jdk.internal.util.Preconditions.outOfBounds(Preconditions.java:64)
        at java.base/jdk.internal.util.Preconditions.outOfBoundsCheckIndex(Preconditions.java:70)
        at java.base/jdk.internal.util.Preconditions.checkIndex(Preconditions.java:248)
        at java.base/java.util.Objects.checkIndex(Objects.java:372)
        at java.base/java.util.ArrayList.set(ArrayList.java:472)
        at abc.main(abc.java:13)

Completed with exit code: 1
```

**QUESTION 9**
Which two are successful examples of autoboxing? (Choose two.)

A. `String a = “A”;`

B. `Integer e = 5;`

C. `Float g = Float.valueOf(null);`

D. `Double d = 4;`

E. `Long c = 23L;`

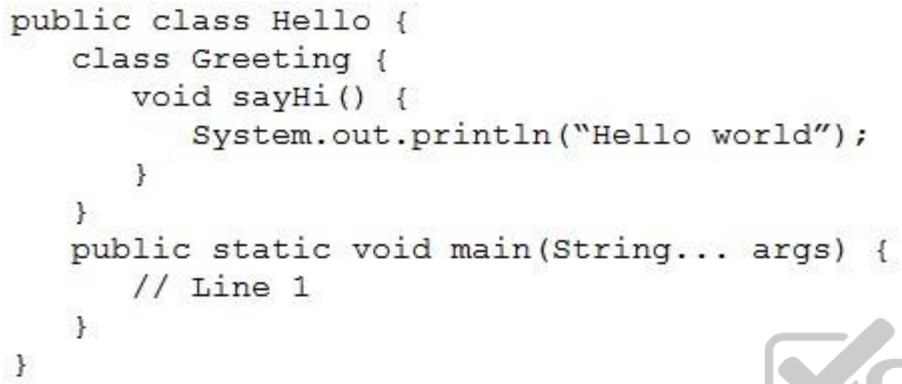
F. `Float f = 6.0;`

**Correct Answer:** AB

**Section: (none)**
**Explanation**

**Explanation/Reference:**

**QUESTION 10**
Given:



```
public class Hello {
    class Greeting {
        void sayHi() {
            System.out.println("Hello world");
        }
    }
    public static void main(String... args) {
        // Line 1
    }
}
```

What code must you insert on Line 1 to enable the code to print `Hello world`?

A. `Hello.Greeting myG = new Hello.Greeting()`
`myG.sayHi();`

B. `Hello myH = new Hello(); Hello.Greeting myG =`
`myH.new Greeting(); myG.sayHi();`

C. `Hello myH = new Hello();`

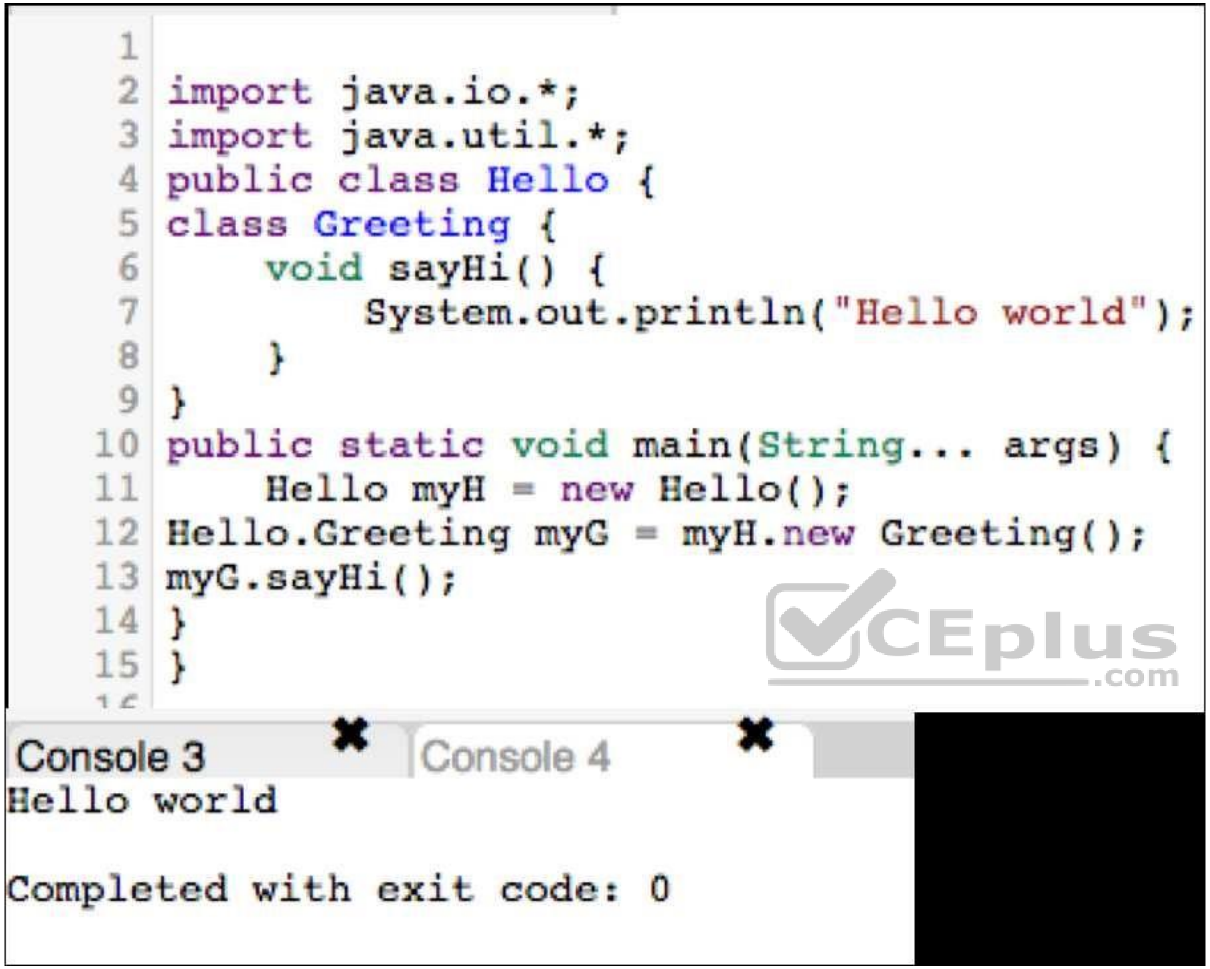
`Hello.Greeting myG = myH.new Hello.Greeting();`
`myG.sayHi();`

D. `Hello myH = new Hello(); Greeting myG = new`
`Greeting(); myG.sayHi ();`

**Correct Answer:** B
**Section: (none)**
**Explanation**

**Explanation/Reference:**

Explanation:



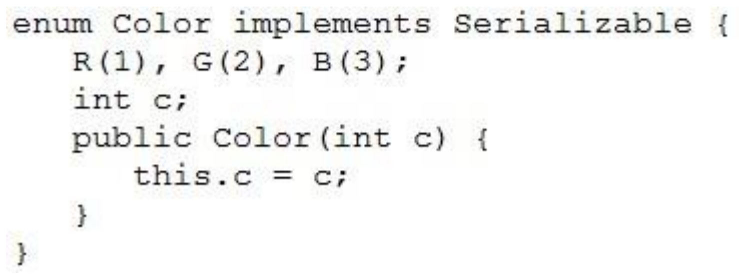
```java

import java.io.*;
import java.util.*;
public class Hello {
class Greeting {
    void sayHi() {
        System.out.println("Hello world");
    }
}
public static void main(String... args) {
    Hello myH = new Hello();
Hello.Greeting myG = myH.new Greeting();
myG.sayHi();
}
}
```

```
Hello world

Completed with exit code: 0
```

[redacted]

**QUESTION 11**

Given:



```
enum Color implements Serializable {
    R(1), G(2), B(3);
    int c;
    public Color(int c) {
        this.c = c;
    }
}
```

What action ensures successful compilation?

A. Replace `public Color(int c)` with `private Color(int c)`.

B. Replace `int c`; with `private int c;`.

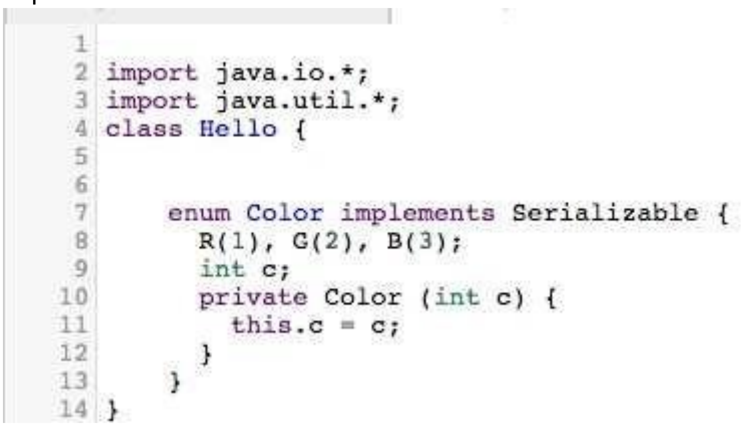
C. Replace `int c`; with `private final int c;`.

D. Replace `enum Color implements Serializable` with `public enum Color`.

E. Replace `enum Color` with `public enum Color`.

**Correct Answer:** A
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Explanation:



```

import java.io.*;
import java.util.*;
class Hello {


    enum Color implements Serializable {
      R(1), G(2), B(3);
      int c;
      private Color (int c) {
        this.c = c;
      }
    }
}
```

**QUESTION 12** `var numbers = List.of(0,1,2,3,4,5,6,7,8,9);`

You want to calculate the average of `numbers`.

Which two codes will accomplish this? (Choose two.)

A. `double avg = numbers.stream().parallel().averagingDouble(a -> a);`

B. `double avg = numbers.parallelStream().mapToInt (m -> m).average().getAsDouble();`

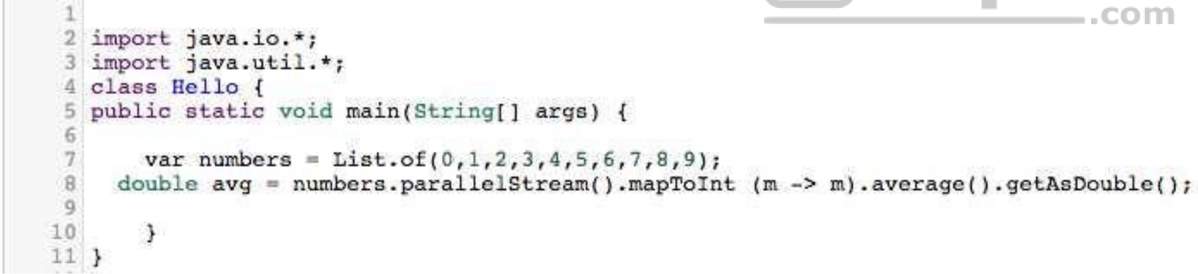
C. `double avg = numbers.stream().mapToInt (i -> i).average().parallel();`

D. `double avg = numbers.stream().average().getAsDouble();`

E. `double avg = numbers.stream().collect(Collectors.averagingDouble(n -> n));`

**Correct Answer:** BD
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Explanation:



```
import java.io.*;
import java.util.*;
class Hello {
public static void main(String[] args) {

    var numbers = List.of(0,1,2,3,4,5,6,7,8,9);
  double avg = numbers.parallelStream().mapToInt (m -> m).average().getAsDouble();

    }
}
```

**QUESTION 13**
Given:

```
// line 1
List<String> fruits = new ArrayList<>(List.of("apple", "orange","banana"));
fruits.replaceAll(fucntion);
```

Which statement on line 1 enables this code fragment to compile?

A. `Function function = String::toUpperCase;`

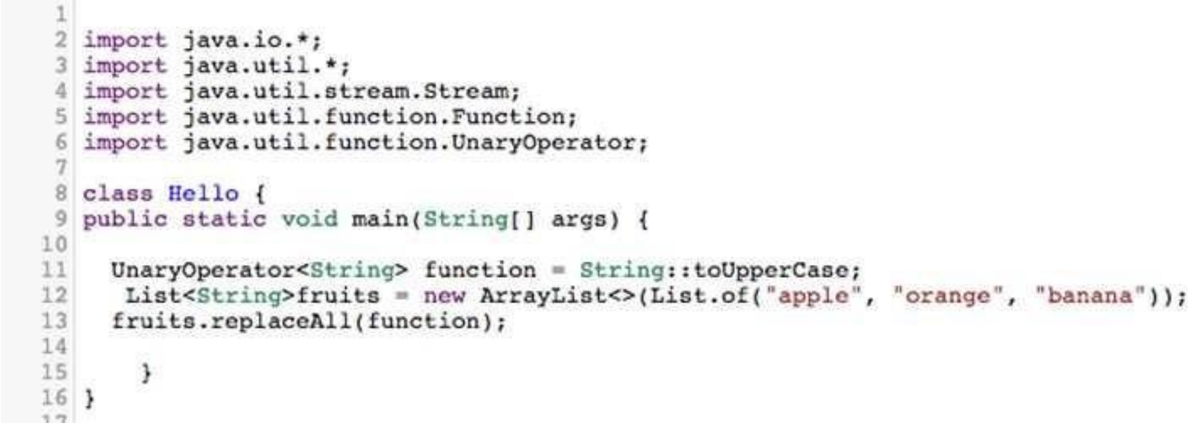
B. `UnaryOperator function = s -> s.toUpperCase();`

C. `UnaryOperator<String> function = String::toUpperCase;`

D. `Function<String> function = m -> m.toUpperCase();`

**Correct Answer:** C
**Section: (none)**
**Explanation**

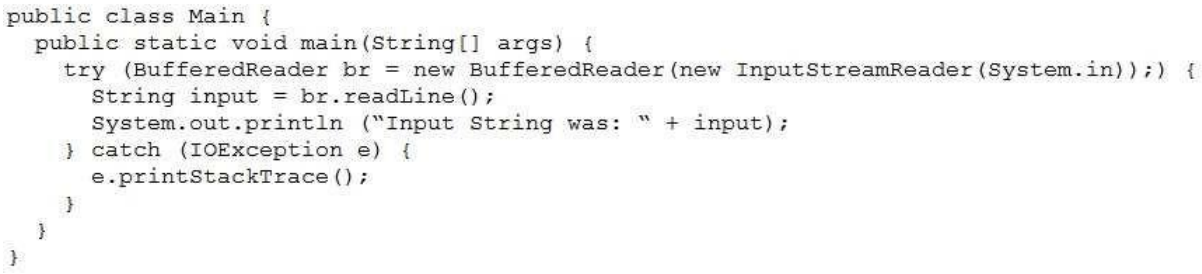
**Explanation/Reference:**
Explanation:



```

import java.io.*;
import java.util.*;
import java.util.stream.Stream;
import java.util.function.Function;
import java.util.function.UnaryOperator;

class Hello {
public static void main(String[] args) {

  UnaryOperator<String> function = String::toUpperCase;
   List<String>fruits = new ArrayList<>(List.of("apple", "orange", "banana"));
  fruits.replaceAll(function);

    }
}
```

**QUESTION 14**

Given:



```
public class Main {
  public static void main(String[] args) {
    try (BufferedReader br = new BufferedReader(new InputStreamReader(System.in));) {
      String input = br.readLine();
      System.out.println (“Input String was: ” + input);
    } catch (IOException e) {
      e.printStackTrace();
    }
  }
}
```

Which is true?

A. `System.out` is the standard output stream. The stream is open only when `System.out` is called.

B. `System.in` cannot reassign the other stream.

C. `System.out` is an instance of `java.io.OutputStream` by default.

D. `System.in` is the standard input stream. The stream is already open.

**Correct Answer:** D

**Section: (none)**

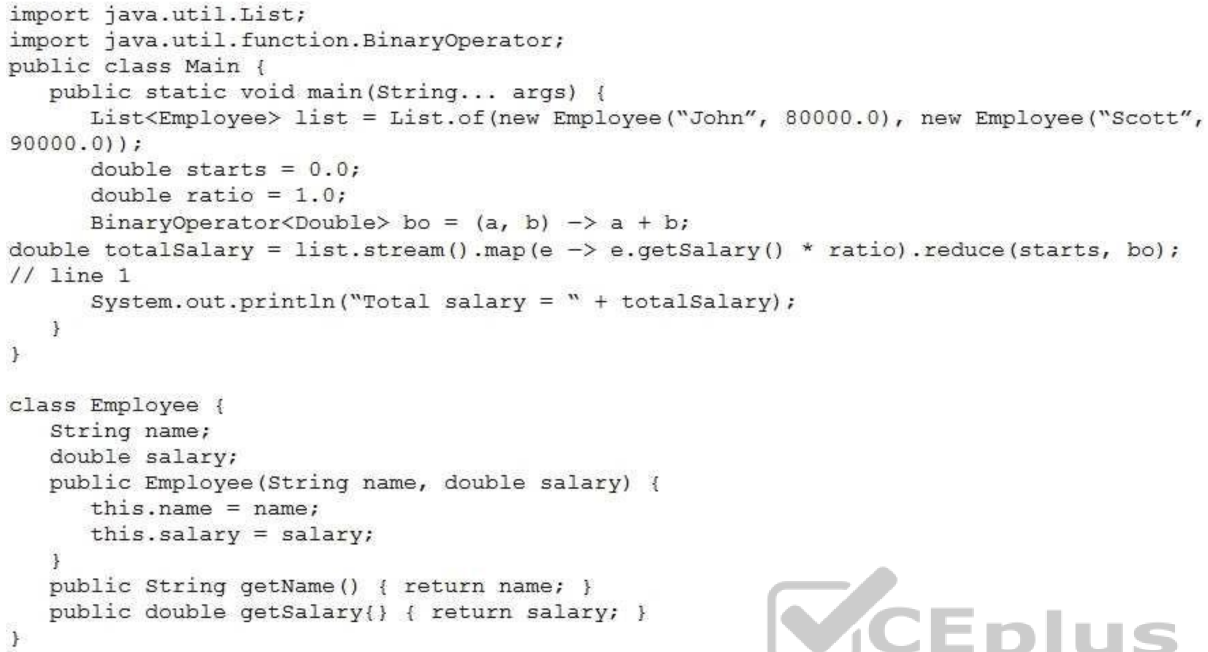
**Explanation**

**Explanation/Reference:**

Reference: https://www.geeksforgeeks.org/java-lang-system-class-java/

**QUESTION 15**

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());`

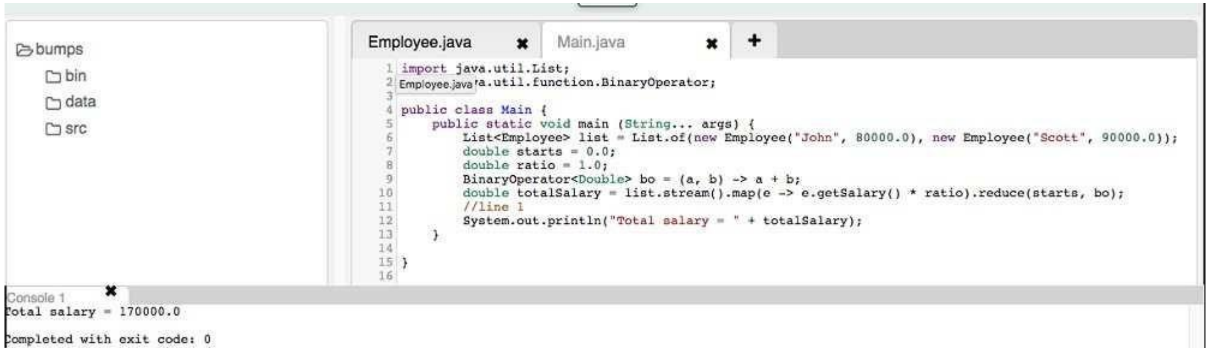
B. `double totalSalary = list.stream().mapToDouble(e -> e.getSalary() * ratio).sum;`

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);`

**Correct Answer:** C
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Explanation:



```
import java.util.List;
Employee.java'a.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);
    }

}
```

```
Total salary = 170000.0

Completed with exit code: 0
```

**QUESTION 16**

Given:

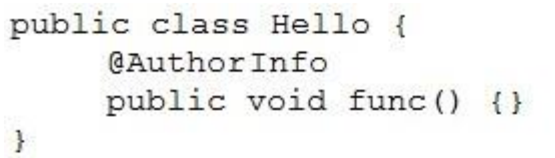
```
@Target(ElementType.METHOD)
@Retention(RetentionPolicy.RUNTIME)
public @interface AuthorInfo {
     String author() default “”;
     String date();
     String[] comments() default {};
}
```

Which two are correct? (Choose two.)

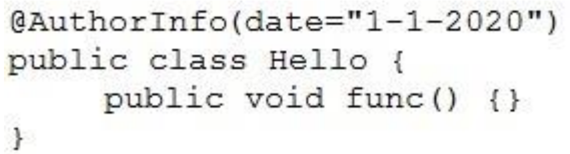
```
@AuthorInfo(date="1-1-2020", comments={ null })
public class Hello {
     public void func() {}
}
public class Hello {
@AuthorInfo (date="1-1-2020. comments=“Hello")
     public void func() {}
}
```

A.

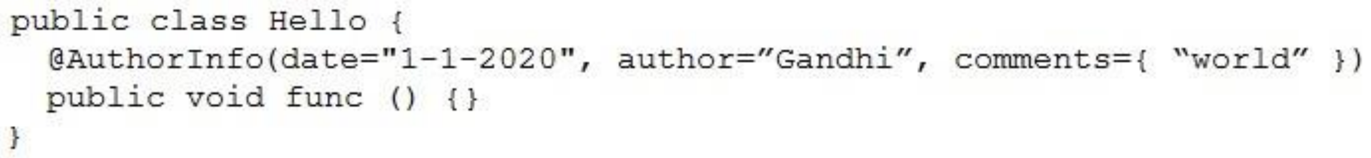
B.



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



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



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

C.

D.

E.

**Correct Answer:** CD

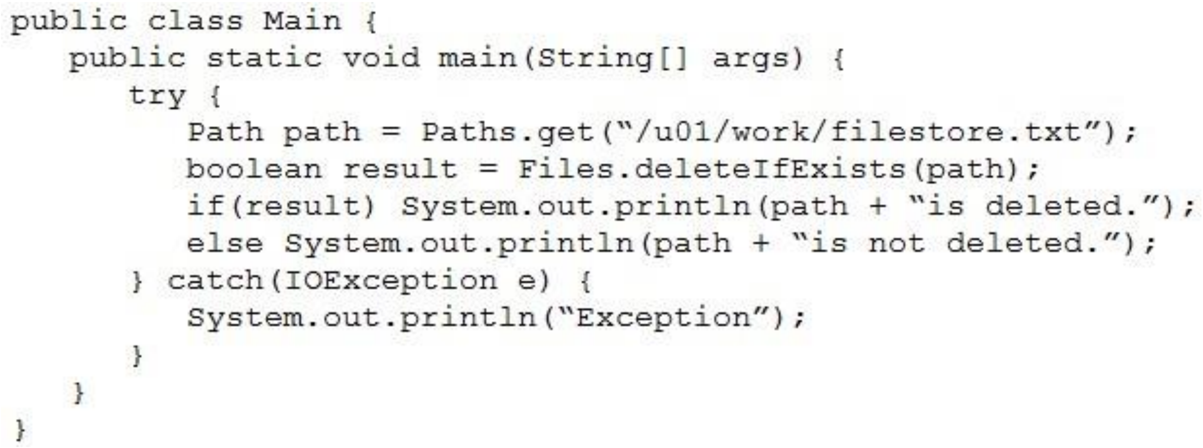
**Section: (none)**

**Explanation**

**Explanation/Reference:**

**QUESTION 17**

Given:



```
public class Main {
    public static void main(String[] args) {
        try {
            Path path = Paths.get(“/u01/work/filestore.txt”);
            boolean result = Files.deleteIfExists(path);
            if(result) System.out.println(path + “is deleted.”);
            else System.out.println(path + “is not deleted.”);
        } catch(IOException e) {
            System.out.println(“Exception”);
        }
    }
}
```

Assume the file on `path` does not exist.

What is the result?

A. The compilation fails.

B. `/u01/work/filestore.txt is not deleted`.

C. `Exception`

D. `/u01/work/filestore.txt is deleted`.

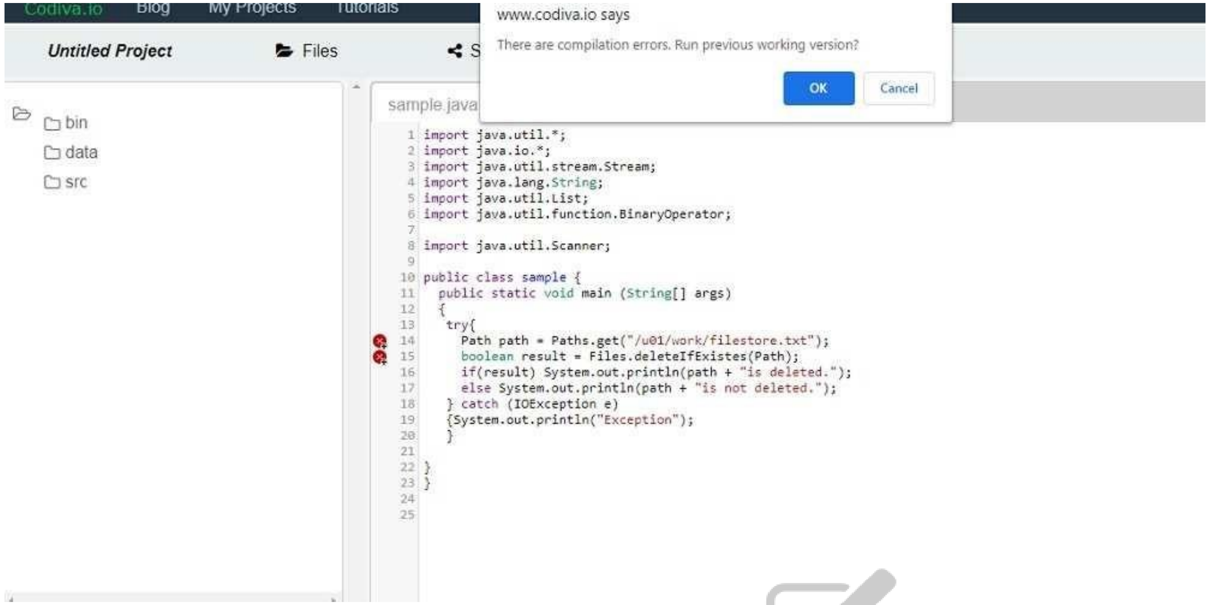
**Correct Answer:** A

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation:



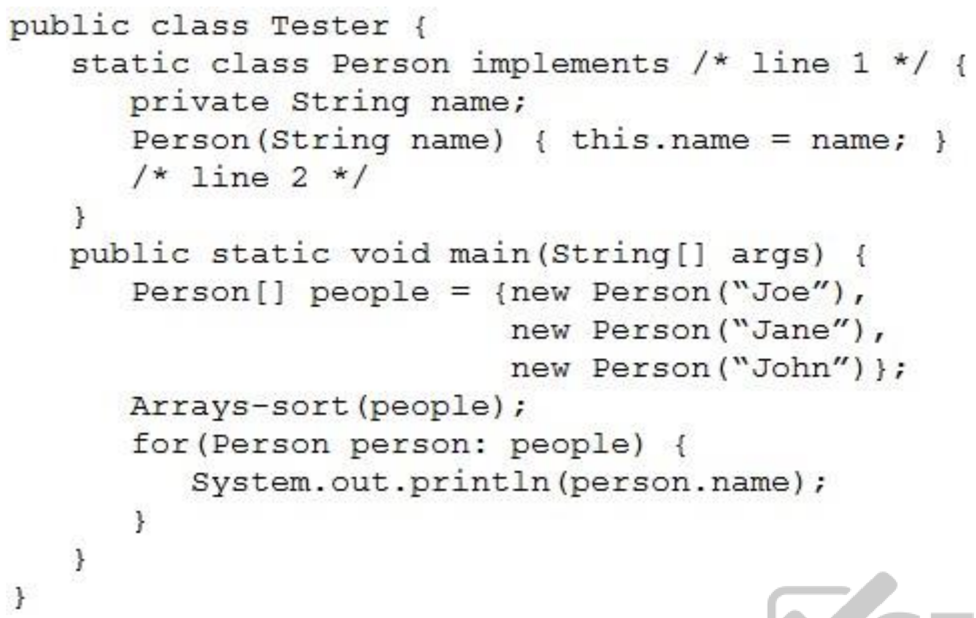
www.codiva.io says

There are compilation errors. Run previous working version?

```java
import java.util.*;
import java.io.*;
import java.util.stream.Stream;
import java.lang.String;
import java.util.List;
import java.util.function.BinaryOperator;

import java.util.Scanner;

public class sample {
  public static void main (String[] args)
  {
   try{
     Path path = Paths.get("/u01/work/filestore.txt");
     boolean result = Files.deleteIfExistes(Path);
     if(result) System.out.println(path + "is deleted.");
     else System.out.println(path + "is not deleted.");
   } catch (IOException e)
   {System.out.println("Exception");
   }

}
}
```

**QUESTION 18**

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

B. Insert `Comparator<Person>` on line 1.
   Insert
   `public int compareTo(Person person) {`
   `return person.name.compareTo(this.name);`

} on line
2.

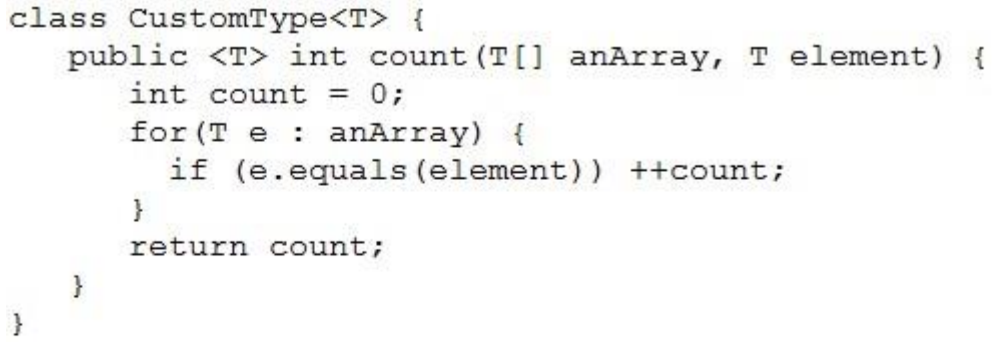
C. Insert `Comparable<Person>` on line 1.
Insert
```
public int compare(Person p1, Person p2) {
return p1.name.compare(p2.name);
```
} on line
2.

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

**Correct Answer:** B
**Section: (none)**
**Explanation**

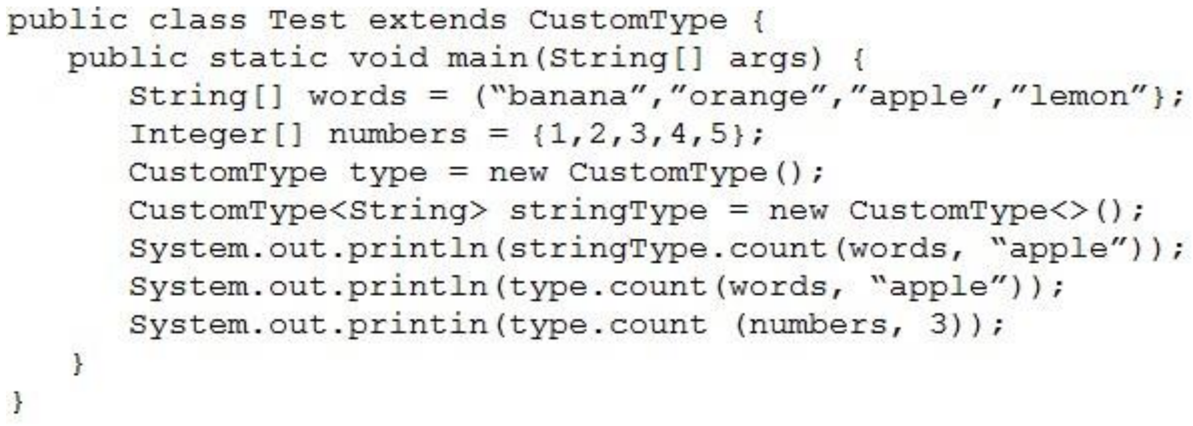
**Explanation/Reference:**
Reference: https://www.coursehero.com/file/p320ss6/Override-public-int-compareTo-Person-other-Compare-this-objects-name-to-others/

**QUESTION 19**
Given:



```
class CustomType<T> {
   public <T> int count(T[] anArray, T element) {
      int count = 0;
      for(T e : anArray) {
        if (e.equals(element)) ++count;
      }
      return count;
   }
}
```

and



```
public class Test extends CustomType {
   public static void main(String[] args) {
      String[] words = (“banana”,”orange”,”apple”,”lemon”};
      Integer[] numbers = {1,2,3,4,5};
      CustomType type = new CustomType();
      CustomType<String> stringType = new CustomType<>();
      System.out.println(stringType.count(words, “apple”));
      System.out.println(type.count(words, “apple”));
      System.out.printin(type.count (numbers, 3));
   }
}
```

What is the result?

A. A `NullPointerException` is thrown at run time.

B. The compilation fails.

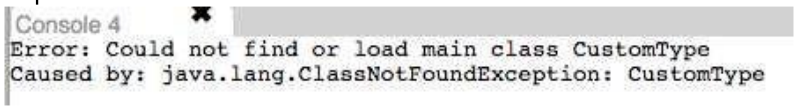
C. `1`
`Null`
`null`

D. `1`
`1`
`1`

E. A `ClassCastException` is thrown at run time.

**Correct Answer:** B
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Explanation:



```
Error: Could not find or load main class CustomType
Caused by: java.lang.ClassNotFoundException: CustomType
```

**QUESTION 20**

Given:

```
public class X {
}
```

and

```
public final class Y extends X {
}
```

What is the result of compiling these two classes?

A. The compilation fails because there is no zero args constructor defined in class `X`.

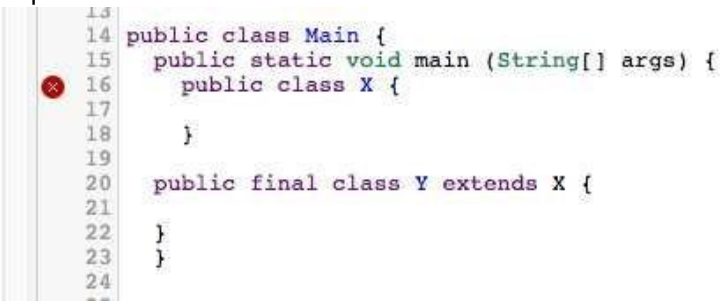
B. The compilation fails because either class `X` or class `Y` needs to implement the `toString()` method.

C. The compilation fails because a final class cannot extend another class.

D. The compilation succeeds.

**Correct Answer:** B
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Explanation:



```
public class Main {
  public static void main (String[] args) {
    public class X {

    }

  public final class Y extends X {

  }
  }
```

**QUESTION 21**
Which code is correct?

A. `Runnable r = "Message" -> System.out.println();`

B. `Runnable r = () -> System.out::print;`

C. `Runnable r = () -> {System.out.println(“Message”);};`

D. `Runnable r = -> System.out.println(“Message”);`

E. `Runnable r = {System.out.println(“Message”)};`

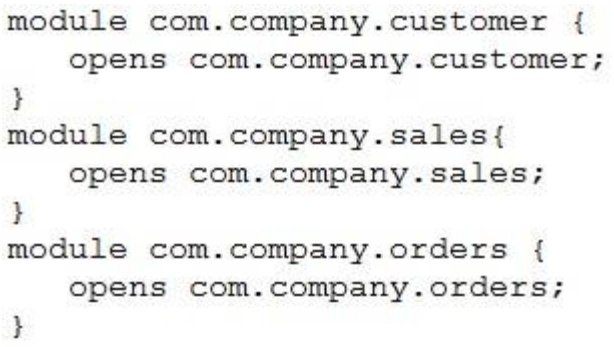
**Correct Answer:** C
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Reference: https://www.oracle.com/technical-resources/articles/java/architect-lambdas-part1.html

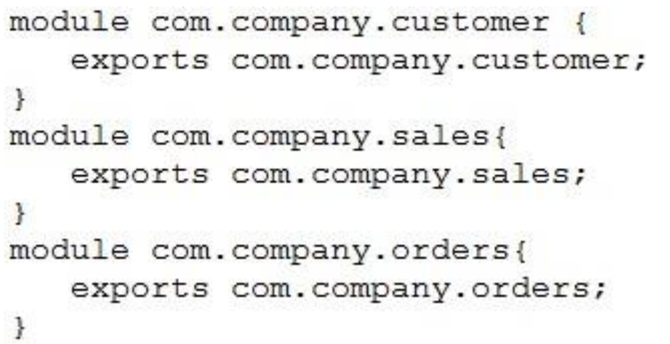
**QUESTION 22**
A company has an existing sales application using a Java 8 jar file containing packages:

```
com.company.customer;
com.company.customer.orders;
com.company.customer.info;
com.company.sales;
com.company.sales.leads;
com.company.sales.closed;
com.company.orders;
com.company.orders.pending;
com.company.orders.shipped.
```

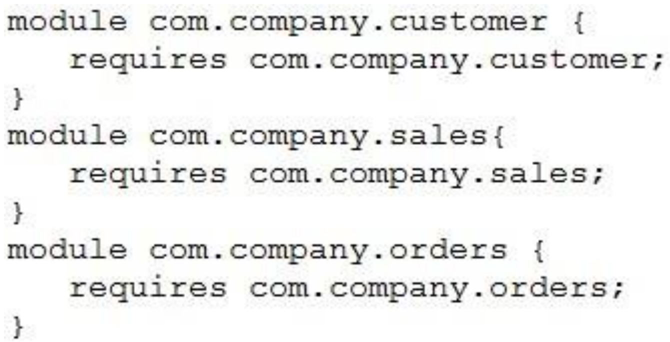
To modularize this jar file into three modules, `customer`, `sales`, and `orders`, which `module-info.java` would be correct?

A. 

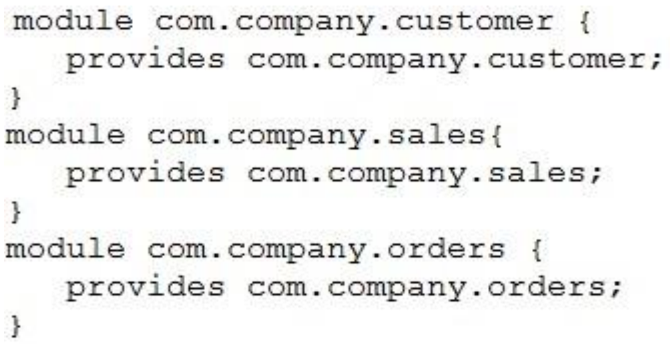
```
module com.company.customer {
    opens com.company.customer;
}
module com.company.sales{
    opens com.company.sales;
}
module com.company.orders {
    opens com.company.orders;
}
```



```
module com.company.customer {
   exports com.company.customer;
}
module com.company.sales{
   exports com.company.sales;
}
module com.company.orders{
   exports com.company.orders;
}
```



```
module com.company.customer {
   requires com.company.customer;
}
module com.company.sales{
   requires com.company.sales;
}
module com.company.orders {
   requires com.company.orders;
}
```



```
module com.company.customer {
   provides com.company.customer;
}
module com.company.sales{
   provides com.company.sales;
}
module com.company.orders {
   provides com.company.orders;
}
```

B. C.

D.

**Correct Answer:** C
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Reference: https://developer.ibm.com/tutorials/java-modularity-3/

**QUESTION 23**
Which is a proper JDBC URL?

A. `jdbe.mysql.com://localhost:3306/database`

B. `http://localhost.mysql.com:3306/database`

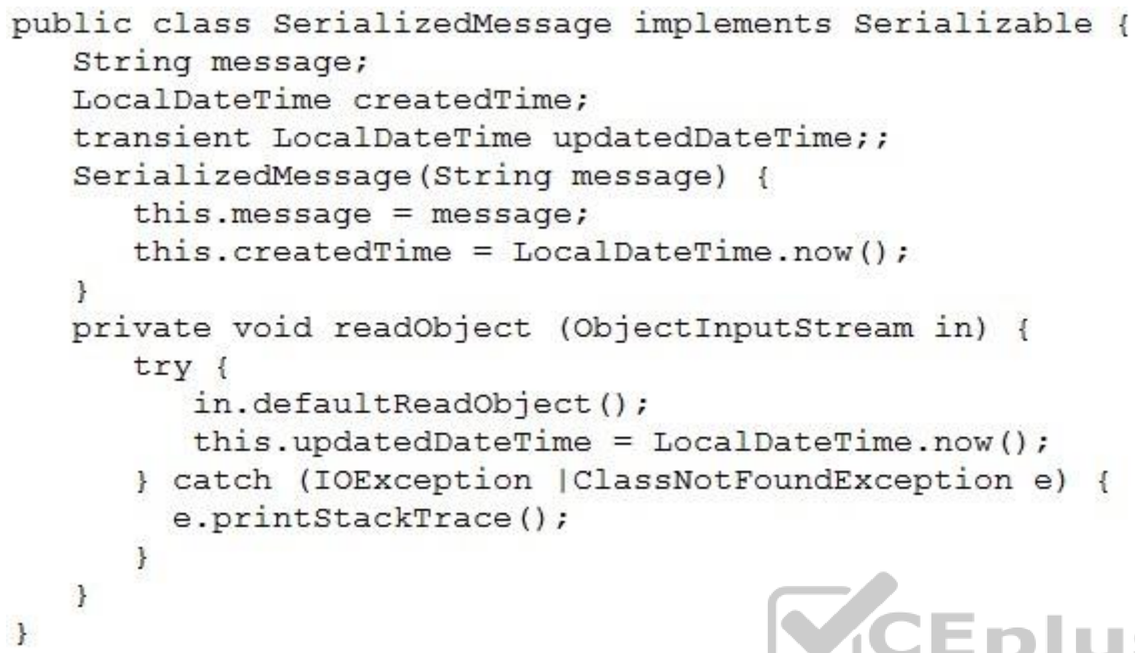
C. `http://localhost mysql.jdbc:3306/database`

D. `jdbc:mysql://localhost:3306/database`

**Correct Answer:** D
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Reference: https://vladmihalcea.com/jdbc-driver-connection-url-strings/

**QUESTION 24**
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

**Correct Answer:** B
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Reference: https://www.oracle.com/technical-resources/articles/java/javaserial.html

**QUESTION 25**
Given:

```
List<String> list1 = new ArrayList<>();
list1.add(“A”); list1.add(“B”); List
list2 = List.copyOf(list1);
list2.add(“C”);
List<List<String>> list3 = List.of(list1, list2);
System.out.println(list3);
```

What is the result?

A. `[[A, B],[A, B]]`

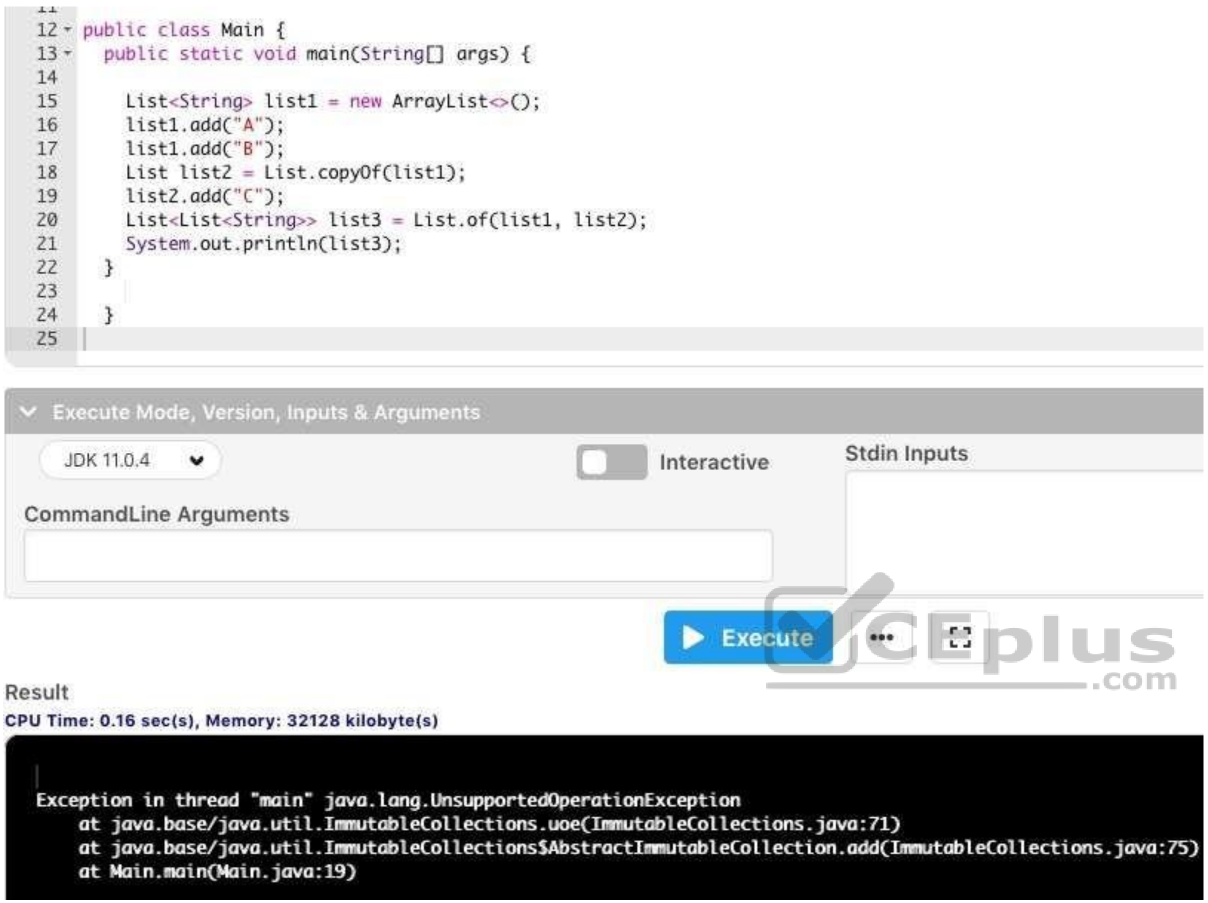
B. An exception is thrown at run time.

C. `[[A, B], [A, B, C]]`

D. `[[A, B, C], [A, B, C]]`

**Correct Answer:** B
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Explanation:



```
public class Main {
  public static void main(String[] args) {

    List<String> list1 = new ArrayList<>();
    list1.add("A");
    list1.add("B");
    List list2 = List.copyOf(list1);
    list2.add("C");
    List<List<String>> list3 = List.of(list1, list2);
    System.out.println(list3);
  }

  }
```

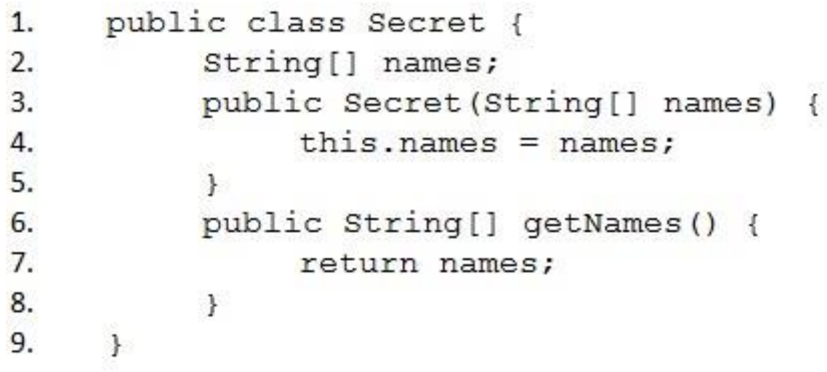
Result

CPU Time: 0.16 sec(s), Memory: 32128 kilobyte(s)

```
Exception in thread "main" java.lang.UnsupportedOperationException
    at java.base/java.util.ImmutableCollections.uoe(ImmutableCollections.java:71)
    at java.base/java.util.ImmutableCollections$AbstractImmutableCollection.add(ImmutableCollections.java:75)
    at Main.main(Main.java:19)
```

**QUESTION 26**

Given:



```
1.    public class Secret {
2.          String[] names;
3.          public Secret(String[] names) {
4.                this.names = names;
5.          }
6.          public String[] getNames() {
7.                return names;
8.          }
9.    }
```

Which three actions implement Java SE security guidelines? (Choose three.)

A. Change line 7 to `return names.clone();`.

B. Change line 4 to `this.names = names.clone();`.

C. Change the `getNames()` method name to `get$Names()`.

D. Change line 6 to `public synchronized String[] getNames() {`.

E. Change line 2 to `private final String[] names;`.

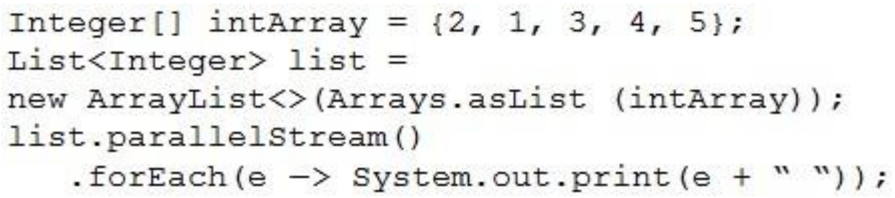
F. Change line 3 to `private Secret(String[] names) {`.

G. Change line 2 to `protected volatile String[] names;`.

**Correct Answer:** EFG
**Section: (none)**
**Explanation**

**Explanation/Reference:**

**QUESTION 27**
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.

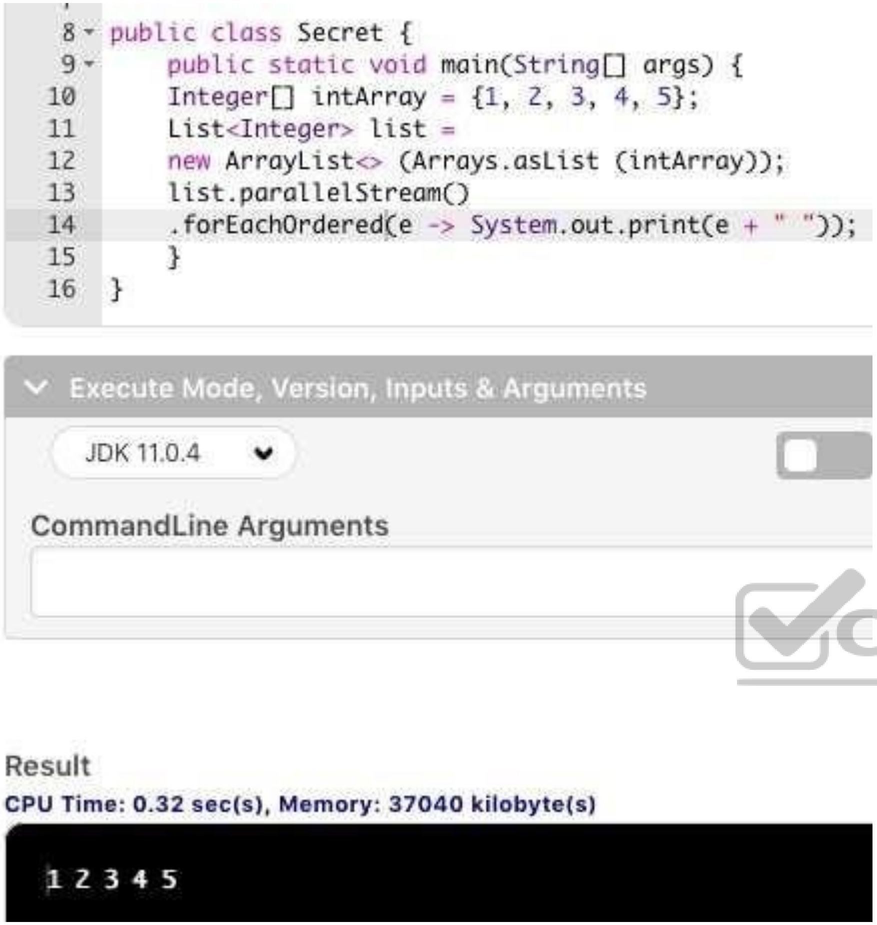
C. Replacing `forEach()` with `forEachOrdered()`, the program prints `2 1 3 4 5`, but the order is unpredictable.

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`.

**Correct Answer:** BD
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Explanation:



```java
public class Secret {
    public static void main(String[] args) {
    Integer[] intArray = {1, 2, 3, 4, 5};
    List<Integer> list =
    new ArrayList<> (Arrays.asList (intArray));
    list.parallelStream()
    .forEachOrdered(e -> System.out.print(e + " "));
    }
}
```

JDK 11.0.4

CommandLine Arguments

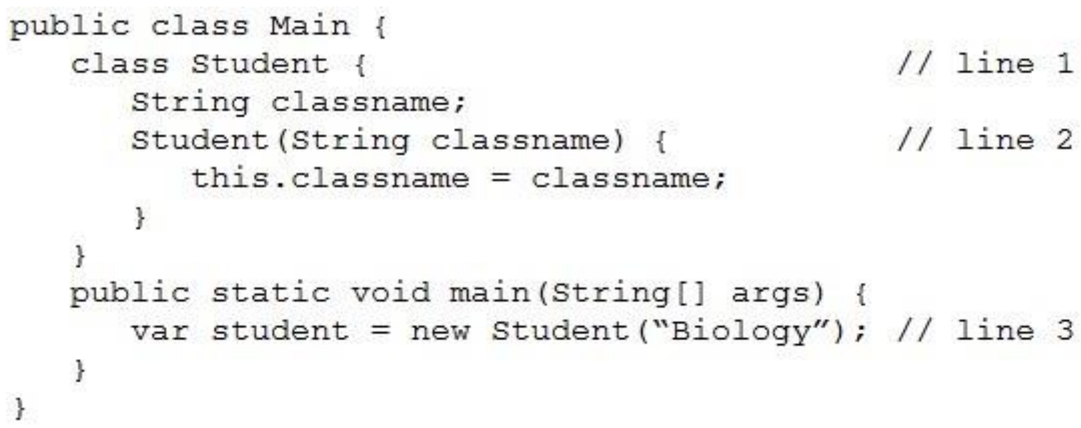
Result

CPU Time: 0.32 sec(s), Memory: 37040 kilobyte(s)

```
1 2 3 4 5
```

**QUESTION 28**

Given:



```
public class Main {
    class Student {                              // line 1
        String classname;
        Student(String classname) {              // line 2
            this.classname = classname;
        }
    }
    public static void main(String[] args) {
        var student = new Student(“Biology”);    // line 3
    }
}
```

Which two independent changes will make the `Main` class compile? (Choose two.)

A. Move the entire `Student` class declaration to a separate Java file, `Student.java`.

B. Change line 2 to `public Student(String classname)`.

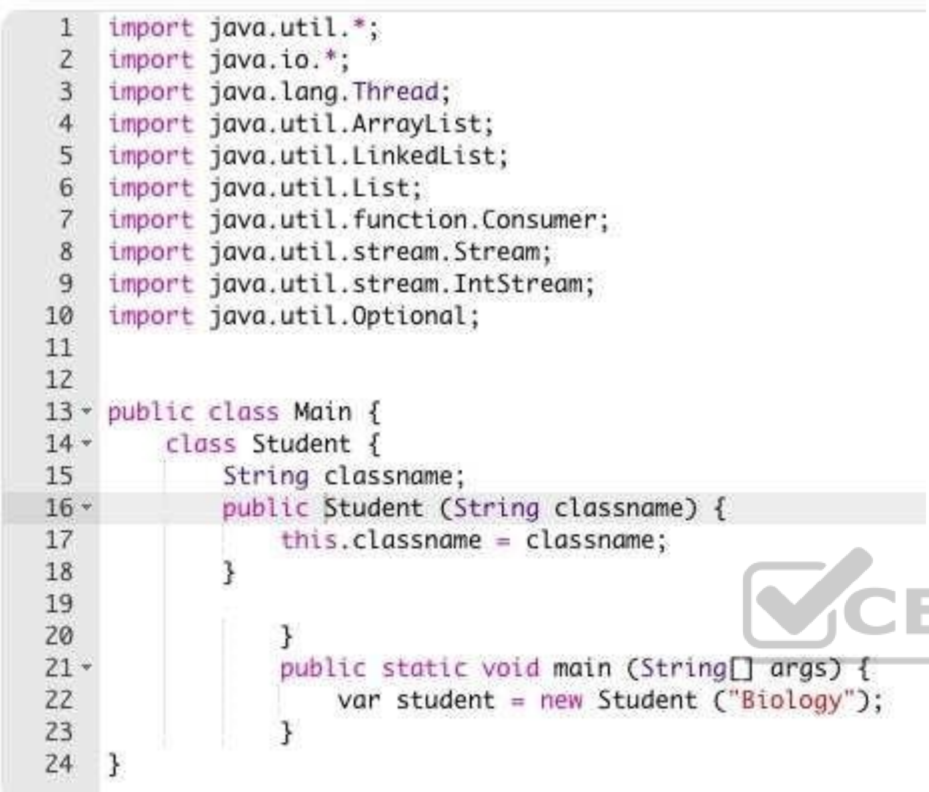
C. Change line 1 to `public class Student {`.

D. Change line 3 to `Student student = new Student(“Biology”);`.

E. Change line 1 to `static class Student {`.

**Correct Answer:** BD
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Explanation:



```
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;
import java.util.stream.Stream;
import java.util.stream.IntStream;
import java.util.Optional;


public class Main {
    class Student {
        String classname;
        public Student (String classname) {
            this.classname = classname;
        }

            }
            public static void main (String[] args) {
                var student = new Student ("Biology");
            }
}
```

**QUESTION 29**

Given the code fragment:

```
var pool = Executors.newFixedThreadPool(5);
Future outcome = pool.submit(() -> 1);
```

Which type of lambda expression is passed into `submit()`?

A. `java.lang.Runnable`

B. `java.util.function.Predicate`

C. `java.util.function.Function`

D. `java.util.concurrent.Callable`

**Correct Answer:** D
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Reference: https://www.codota.com/code/java/methods/java.util.concurrent.Executors/newFixedThreadPool

**QUESTION 30**
Which two statements set the default locale used for formatting numbers, currency, and percentages? (Choose two.)

A. `Locale.setDefault(Locale.Category.FORMAT, “zh-CN”);`

B. `Locale.setDefault(Locale.Category.FORMAT, Locale.CANADA_FRENCH);`

C. `Locale.setDefault(Locale.SIMPLIFIED_CHINESE);`

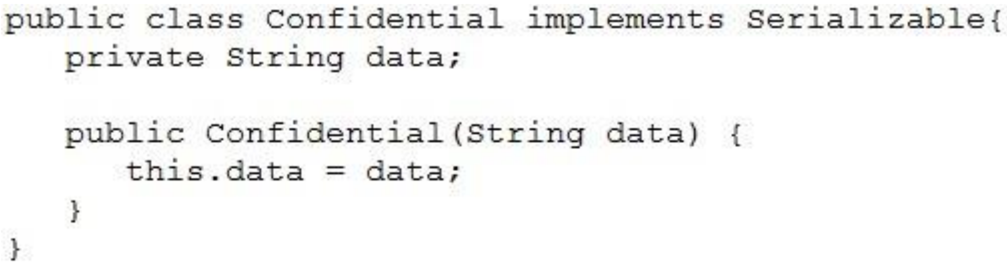
D. `Locale.setDefault(“en_CA”);`

E. `Locale.setDefault(“es”, Locale.US);`

**Correct Answer:** BD
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Reference: https://www.oracle.com/technical-resources/articles/javase/locale.html

**QUESTION 31**
Given:



```
public class Confidential implements Serializable{
   private String data;

   public Confidential(String data) {
      this.data = data;
   }
}
```

Which two are secure serialization of these objects? (Choose two.)

A. Define the `serialPersistentFields` array field.
B. Declare fields `transient`.
C. Implement only `readResolve` to replace the instance with a serial proxy and not `writeReplace`.
D. Make the class abstract.
E. Implement only `writeReplace` to replace the instance with a serial proxy and not `readResolve`.

**Correct Answer:** AC
**Section: (none)**
**Explanation**

**Explanation/Reference:**

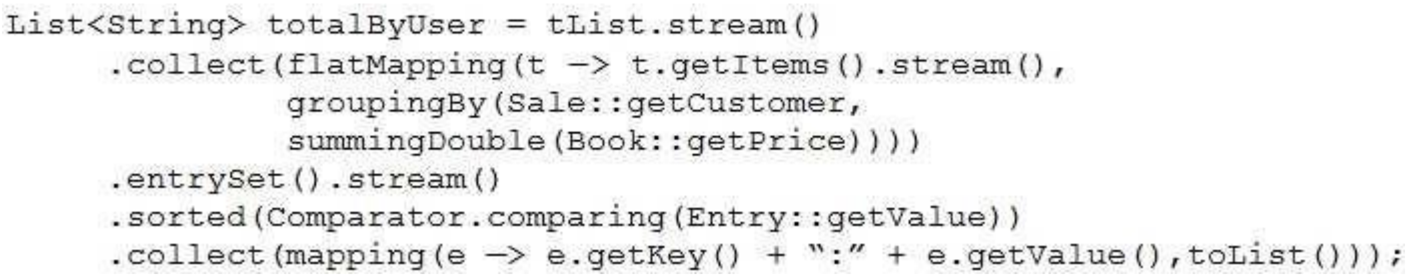
**QUESTION 32**
A bookstore's sales are represented by a list of `Sale` objects populated with the name of the customer and the books they purchased.

```
public class Sale {
private String customer;
private List<Book> items;
// constructor, setters and getters not shown
}
```

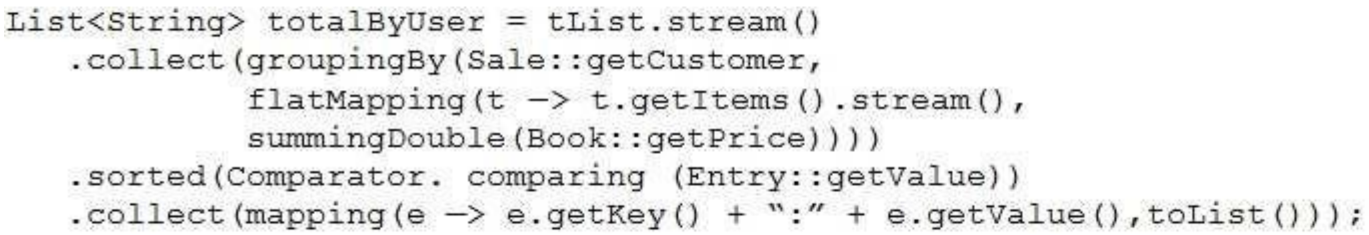
```
public class Book {
private String name;
private double price;
// constructor, setters and getters not shown
}
```

Given a list of `Sale` objects, `tList`, which code fragment creates a list of total sales for each customer in ascending order?

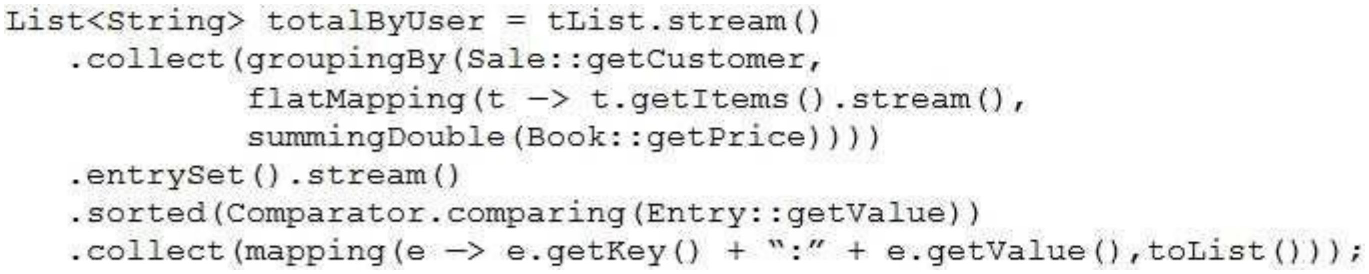
A.



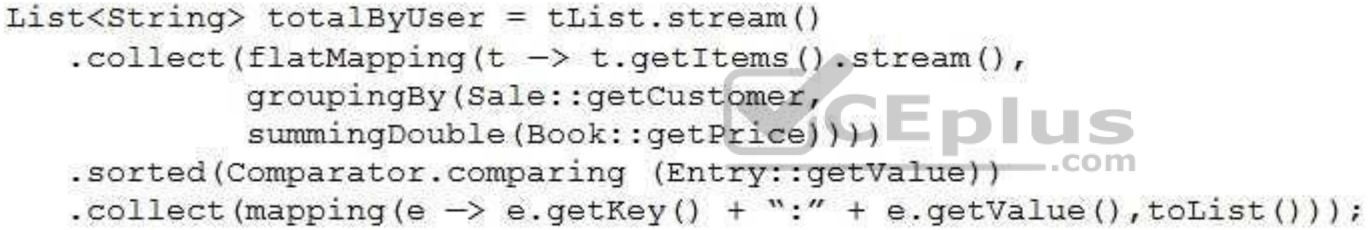
```
List<String> totalByUser = tList.stream()
     .collect(flatMapping(t -> t.getItems().stream(),
              groupingBy(Sale::getCustomer,
              summingDouble(Book::getPrice))))
     .entrySet().stream()
     .sorted(Comparator.comparing(Entry::getValue))
     .collect(mapping(e -> e.getKey() + ":" + e.getValue(),toList()));
```



```
List<String> totalByUser = tList.stream()
   .collect(groupingBy(Sale::getCustomer,
            flatMapping(t -> t.getItems().stream(),
            summingDouble(Book::getPrice))))
   .sorted(Comparator. comparing (Entry::getValue))
   .collect(mapping(e -> e.getKey() + “:” + e.getValue(),toList()));
```



```
List<String> totalByUser = tList.stream()
   .collect(groupingBy(Sale::getCustomer,
            flatMapping(t -> t.getItems().stream(),
            summingDouble(Book::getPrice))))
   .entrySet().stream()
   .sorted(Comparator.comparing(Entry::getValue))
   .collect(mapping(e -> e.getKey() + “:” + e.getValue(),toList()));
```



```
List<String> totalByUser = tList.stream()
   .collect(flatMapping(t -> t.getItems().stream(),
            groupingBy(Sale::getCustomer,
            summingDouble(Book::getPrice))))
   .sorted(Comparator.comparing (Entry::getValue))
   .collect(mapping(e -> e.getKey() + “:” + e.getValue(),toList()));
```

B.

C.

D.

**Correct Answer:** C
**Section: (none)**
**Explanation**

**Explanation/Reference:**

**QUESTION 33**
Which three annotation uses are valid? (Choose three.)

A. `Function<String, String> func = (@NonNull x) -> x.toUpperCase();`

B. `var v = “Hello” + (@Interned) “World”`

C. `Function<String, String> func = (var @NonNull x) -> x.toUpperCase();`

D. `Function<String, String> func = (@NonNull var x) -> x.toUpperCase();`

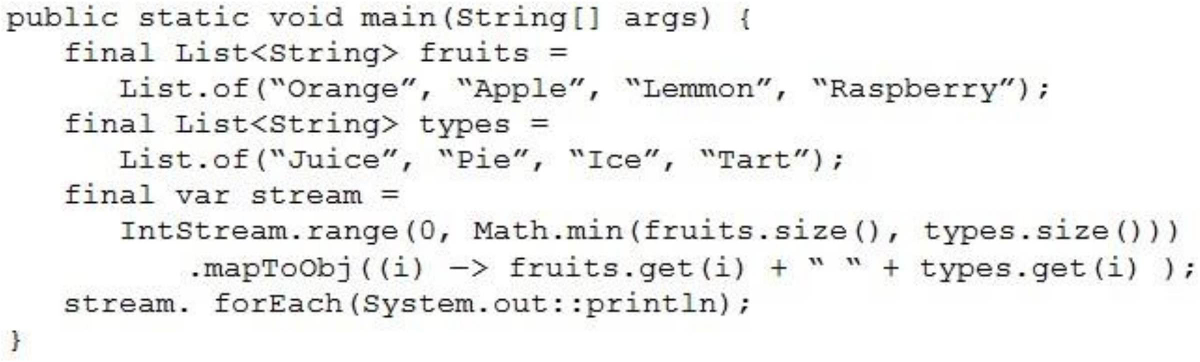
E. `var myString = (@NonNull String) str;`

F. `var obj = new @Interned MyObject();`

**Correct Answer:** ACF
**Section: (none)**
**Explanation**

**Explanation/Reference:**

**QUESTION 34**
Given:



```
public static void main(String[] args) {
    final List<String> fruits =
       List.of(“Orange”, “Apple”, “Lemmon”, “Raspberry”);
    final List<String> types =
       List.of(“Juice”, “Pie”, “Ice”, “Tart”);
    final var stream =
       IntStream.range(0, Math.min(fruits.size(), types.size()))
           .mapToObj((i) -> fruits.get(i) + “ “ + types.get(i) );
    stream. forEach(System.out::println);
}
```

What is the result?

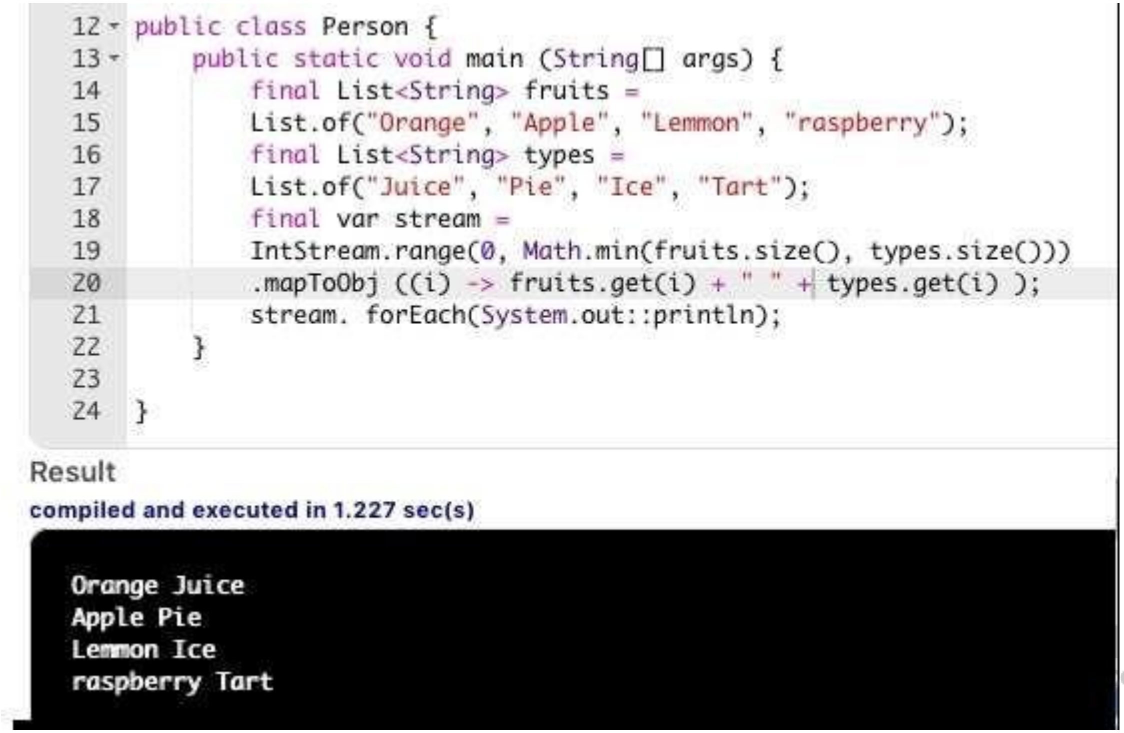
A. `Orange Juice`

B. The compilation fails.

C.
```
Orange Juice
Apple Pie
Lemmon Ice
Raspberry Tart
```

D. The program prints nothing.

**Correct Answer:** C
**Section: (none)**
**Explanation**
**Explanation/Reference:**
Explanation:



```
public class Person {
    public static void main (String[] args) {
        final List<String> fruits =
        List.of("Orange", "Apple", "Lemmon", "raspberry");
        final List<String> types =
        List.of("Juice", "Pie", "Ice", "Tart");
        final var stream =
        IntStream.range(0, Math.min(fruits.size(), types.size()))
        .mapToObj ((i) -> fruits.get(i) + " " + types.get(i) );
        stream. forEach(System.out::println);
    }

}
```

Result

compiled and executed in 1.227 sec(s)

```
Orange Juice
Apple Pie
Lemmon Ice
raspberry Tart
```

**QUESTION 35**

Which interface in the `java.util.function` package can return a primitive type?

A. `ToDoubleFunction`

B. `Supplier`

C. `BiFunction`

D. `LongConsumer`

**Correct Answer:** A

**Section: (none)**

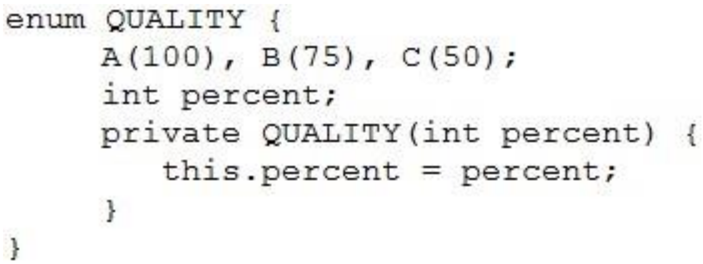
**Explanation**

**Explanation/Reference:**

Reference: http://java.boot.by/ocpjp8-upgrade-guide/ch02s07.html

**QUESTION 36**

Given:

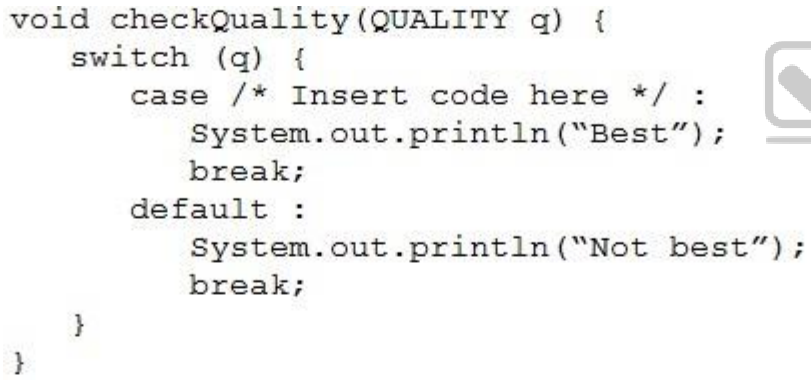


```
enum QUALITY {
     A(100), B(75), C(50);
     int percent;
     private QUALITY(int percent) {
        this.percent = percent;
     }
}
```

and

```
checkQuality(QUALITY.A);
```

and



```
void checkQuality(QUALITY q) {
   switch (q) {
      case /* Insert code here */ :
         System.out.println("Best");
         break;
      default :
         System.out.println("Not best");
         break;
   }
}
```

Which code fragment can be inserted into the switch statement to print `Best`?

A. `QUALITY.A.ValueOf()`

B. `A`

C. `A.toString()`

D. `QUALITY.A`

**Correct Answer:** B
**Section: (none)**
**Explanation**

**Explanation/Reference:**

**QUESTION 37**
Given:

```
LocalDate d1 = LocalDate.of(1997,2,7);
DateTimeFormatter dtf =
DateTimeFormatter.ofPattern( /*insert code here*/ );
System.out.println(dtf.format (d1));
```

Which pattern formats the date as `Friday 7th of February 1997?`

A. `“eeee dd+”th of”+ MMM yyyy”`

B. `“eeee dd'th of' MMM yyyy”`

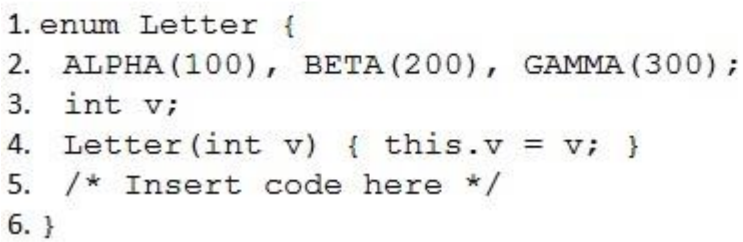
C. `“eeee d+”th of”+ MMMM yyyy”`

D. `“eeee d’th of’ MMMM yyyy”`

**Correct Answer:** B
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Reference: https://books.google.com.pk/books?id=PmiO65T9hF0C&pg=PA385&lpg=PA385&dq=java+pattern+formats+eeee+d%2Bth+of%2B+MMMM+yyyy&source=bl&ots=IJN_-AnWQj&sig=ACfU3U2RJf7iuK3t_SKARwLSaak9xxV09A&hl=en&sa=X&ved=2ahUKEwi4m6LL3vLoAhVgTRUIHURpC38Q6AEwDHoECBQQAQ#v=onepage&q=jav a%20pattern%20formats%20eeee%20d%2Bth%20of%2B%20MMMM%20yyyy&f=false

**QUESTION 38**
Given this enum declaration:



```
1. enum Letter {
2.  ALPHA(100), BETA(200), GAMMA(300);
3.  int v;
4.  Letter(int v) { this.v = v; }
5.  /* Insert code here */
6. }
```

Examine this code:

```
System.out.println(Letter.values()[1]);
```

What code should be written at line 5 for this code to print 200?

A. `public String toString() { return String.valueOf(ALPHA.v); }`

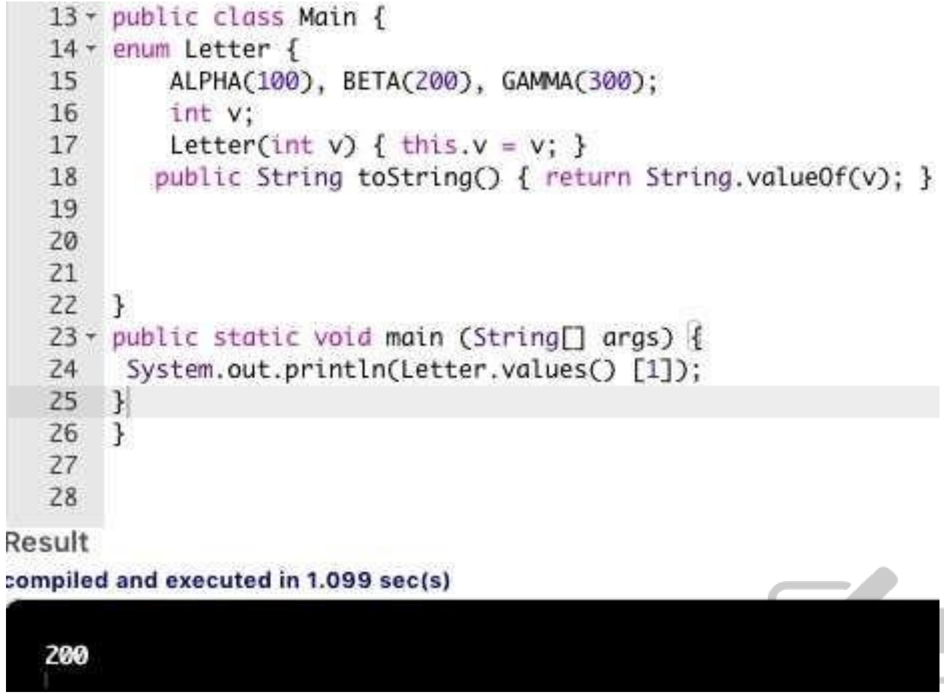
B. `public String toString() { return String.valueOf(Letter.values()[1]); }`

C. `public String toString() { return String.valueOf(v); }`

D. `String toString() { return “200”; }`

**Correct Answer:** C
**Section: (none)**
**Explanation**

**Explanation/Reference:**
Explanation:



```
public class Main {
enum Letter {
    ALPHA(100), BETA(200), GAMMA(300);
    int v;
    Letter(int v) { this.v = v; }
   public String toString() { return String.valueOf(v); }



}
public static void main (String[] args) {
 System.out.println(Letter.values() [1]);
}
}
```

Result

compiled and executed in 1.099 sec(s)

```
200
```