1z0-816.VCEplus.premium.exam.80q

Number: 1z0-816
Passing Score: 800
Time Limit: 120 min
File Version: 1.0

1z0-816

Java SE 11 Programmer II

Version 1.0

Exam A

QUESTION 1

Given the code fragment:

```
Path currentFile = Paths.get("/scratch/exam/temp.txt");
Path outputFile = Paths get("/scratch/exam/new.txt");
Path directory = Paths.get("/scratch/");

Files.copy(currentFile, outputFile);
Files.copy(outputFile, directory);
Files.delete (outputFile);
```

The `/scratch/exam/temp.txt` file exists. The `/scratch/exam/new.txt` and `/scratch/new.txt` files do not exist.

What is the result?

A. `/scratch/exam/new.txt` and `/scratch/new.txt` are deleted.
B. The program throws a `FileaAlreadyExistsException`.
C. The program throws a `NoSuchFileException`.
D. A copy of `/scratch/exam/new.txt` exists in the `/scratch` directory and `/scratch/exam/new.txt` is deleted.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Expalanation:

```
27  public class Main {
28    public static void main(String[] args) {
29    Path currentFile = Paths.get("/scratch/exam/temp.txt");
30    Path outputFile = Paths.get("/scratch/exam/new.txt");
31    Path directory = Paths.get("/scratch/");
32
33    Files.copy(currentFile, outputFile);
34    Files.copy(outputFile, directory);
35    Files.delete (outputFile);
36   }
37   }
38
```

QUESTION 2 Which two are functional interfaces?

(Choose two.) A.

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

B.

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

```
interface MyRunnable {
   public default void run() {}
   public void run(String s);
}
@FunctionalInterface
interface MyRunnable {
}
interface MyRunnable {
   @FunctionalInterface
   public void run();
}
```

C.

D.

E.

Correct Answer: CE
Section: (none)
Explanation

Explanation/Reference:
Reference: http://tutorials.jenkov.com/java-functional-programming/functional-interfaces.html

QUESTION 3
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 4
Given:

```
interface MyInterface1 {
   public int method() throws Exception;
   private void pMethod() { /* an implementation of pMethod */ }
}
interface MyInterface2 {
   public static void sMethod() { /* an implementation of sMethod */ }
   public boolean equals();
}
interface MyInterface3 {
   public void method();
   public void method(String str);
}
interface MyInterface4 {
   public void dMethod() { /* an implementation of dMethod */ }
   public void method();
}
interface MyInterface5 {
   public static void sMethod();
   public void method(String str);
}
```

Which two interfaces can be used in lambda expressions? (Choose two.)

A. `MyInterface1`
B. `MyInterface3`
C. `MyInterface5`
D. `MyInterface2`
E. `MyInterface4`

Correct Answer: CD
Section: (none)
Explanation

Explanation/Reference:
Reference: https://dzone.com/articles/functional-interface-and-lambda-expression

QUESTION 5
Given this enum declaration:

```
1. enum Alphabet {
2.    A, B, C
3.
4. }
```

Examine this code:

```
System.out.println(Alphabet.getFirstLetter());
```

What code should be written at line 3 to make this code print `A`?

A. `final String getFirstLetter() { return A.toString(); }`
B. `static String getFirstLetter() { return Alphabet.values()[1].toString(); }`
C. `static String getFirstLetter() { return A.toString(); }`
D. `String getFirstLetter() { return A.toString(); }`

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 6
Given these two classes:

```
public class Resource {
   public Worker owner;
   public synchronized boolean claim(Worker worker) {
      if  (owner == null) {
           owner = worker;
           return true;
      }
      else return false;
   }
   public synchronized void release() {
      owner = null;
   }
}

public class Worker {
   public synchronized void work(Resource... resources) {
      for (int i = 0; i < 10; i++) {
           while (!resources[0].claim(this)) { }
           while (!resources[1].claim(this)) { }
           // do work with resource
           resources[1].release();
           resources[0].release();
      }
   }
}
```

And given this fragment:

```
Worker w1 = new Worker();
Worker w2 = new Worker();
Resource r1 = new Resource();
Resource r2 = new Resource();
new Thread( () -> {
     w1.work(r1, r2);
} ).start();
new Thread( () -> {
     w2.work(r2, r1);
} ).start();
```

Which describes the fragment?

A. It throws `IllegalMonitorStateException`.
B. It is subject to deadlock.
C. It is subject to livelock.
D. The code does not compile.

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 7
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");
}
default void probeProcedure(){
   System.out.println("Launch Probe");
   System.out.println("Land on Asteroid");
}
```

A.

B. C.

D.

Correct Answer: B
Section: (none)
Explanation
Explanation/Reference:

QUESTION 8
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:

```
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 9 Which code fragment does a service use to load the service provider with a `Print` interface?

A. `private Print print = com.service.Provider.getInstance();`
B. `private java.util.ServiceLoader<Print> loader = ServiceLoader.load(Print.class);`
C. `private java.util.ServiceLoader<Print> loader = new java.util.ServiceLoader<>();`
D. `private Print print = new com.service.Provider.PrintImpl();`

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

Reference: https://docs.oracle.com/javase/8/docs/api/?java/util/ServiceLoader.html

QUESTION 10
Examine these module declarations:

```
module ServiceAPI {
    exports com.example.api;
}

module ServiceProvider {
    requires ServiceAPI;
    provides com.example.api with com.myimpl.Impl;
}

module Consumer {
    requires ServiceAPI;
    uses com.example.api;
}
```

Which two statements are correct? (Choose two.)

A. The `ServiceProvider` module is the only module that, at run time, can provide the `com.example.api` API.
B. The placement of the `com.example.api` API in a separate module, `ServiceAPI`, makes it easy to install multiple provider modules.
C. The `Consumer` module should require the `ServiceProvider` module.
D. The `ServiceProvider` module should export the `com.myimpl` package.
E. The `ServiceProvider` module does not know the identity of a module (such as Consumer) that uses the `com.example.api` API.

Correct Answer: AC
Section: (none)
Explanation

Explanation/Reference:

QUESTION 11
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:

```
14
15  public class Main {
16    public static void main(String[] args) {
17      Optional<String> value = createValue();
18      String str = value.orElse ("Duke");
19      System.out.println(str);
20    }
21    static Optional<String> createValue() {
22      String s = null;
23      return Optional.ofNullable(s);
24    }
25  }
26
```

Result

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

```
Duke
```

QUESTION 12
Given:

```
1. public class Test {
2.    private static class Greet {
3.       private void print() {
4.          System.out.println("Hello World");
5.       }
6.    }
7.    public static void main(String[] args) {
8.       Test.Greet i = new Greet();
9.       i.print();
10.   }
11. }
```

What is the result?

A. The compilation fails at line 9.
B. The compilation fails at line 2.
C. `Hello World`
D. The compilation fails at line 8.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Explanation:

```
1  public class Test {
2      private static class Greet {
3          private void print() {
4              System.out.println("Hello World");
5          }
6      }
7      public static void main(String[] args) {
8          Test.Greet i = new Greet();
9          i.print();
10     }
11 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.16 sec(s), Memory: 32504 kilobyte(s)

```
Hello World
```

QUESTION 13

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 14

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 15
Given:

```
var numbers = List.of(1,2,3,4,5,6,7,8,9,10);
// line 1
StringBuilder sb = new StringBuilder();
for(int a: numbers) {
    sb.append(f.apply(a));
    sb.append(" ");
}
System.out.println(sb.toString());
```

Which statement on line 1 enables this code to compile?

A. `Function<Integer, Integer> f = n -> n * 2;`
B. `Function<Integer> f = n -> n * 2;`
C. `Function<int> f = n -> n * 2;`
D. `Function<int, int> f = n -> n * 2;`
E. `Function f = n -> n * 2;`

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
Explanation:

```
15
16  public class Main {
17    public static void main(String[] args) {
18      var numbers = List.of(1,2,3,4,5,6,7,8,9,10);
19      Function<Integer, Integer> f = n -> n * 2;
20      StringBuilder sb = new StringBuilder();
21      for(int a: numbers) {
22          sb.append(f.apply(a));
23          sb.append(" ");
24      }
25      System.out.println(sb.toString());
26  }
27  }
28
```

Result

CPU Time: 0.22 sec(s), Memory: 33056 kilobyte(s)

```
2 4 6 8 10 12 14 16 18 20
```

QUESTION 16
Given:

```
var fruits = List.of("apple", "orange", "banana", "lemon");
```

You want to examine the first element that contains the character `n`.

Which statement will accomplish this?

A. `String result = fruits.stream().filter(f -> f.contains("n")).findAny();`
B. `fruits.stream().filter(f -> f.contains("n")).forEachOrdered(System.out::print);`
C. `Optional<String> result = fruits.stream().filter(f -> f.contains("n")).findFirst ();`
D. `Optional<String> result = fruits.stream().anyMatch(f -> f.contains("n"));`

Correct Answer: B
Section: (none)
Explanation
Explanation/Reference:
Explanation:

```
1  import java.io.*;
2  import java.util.*;
3  public class abc {
4    public static void main(String[] args) {
5
6      var fruits = List.of("apple", "orange", "banana", "lemon");
7
8  fruits.stream().filter(f -> f.contains("n")).forEachOrdered(System.out::print);
9
10   }
11 }
12
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4 | Interactive | Stdin Inp

CommandLine Arguments

Execute

Result

CPU Time: 0.19 sec(s), Memory: 33200 kilobyte(s)

```
orangebananalemon
```

QUESTION 17

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 18
Given:

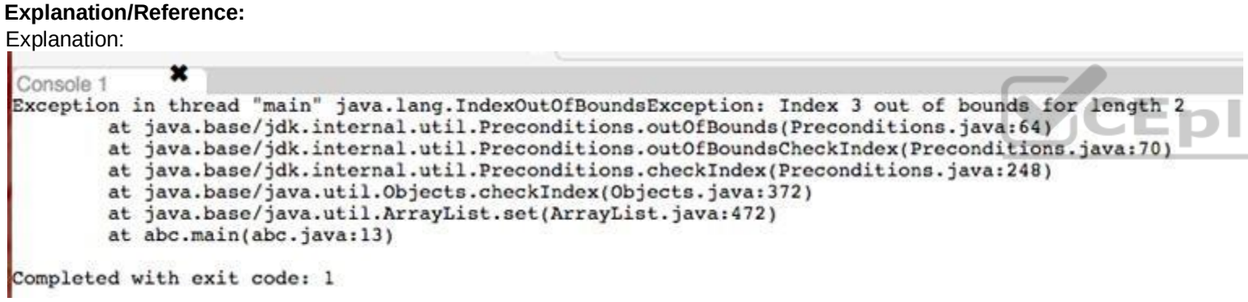
```
var data = new ArrayList<>();
data.add("Peter");
data.add(30);
data.add("Market Road");
data.set(1, 25);
data.remove(2); data.set(3,
1000L);
System.out.print(data);
```

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:



QUESTION 19 Which code

fragment compiles? A.

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

B.

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

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

C.

D.

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation:

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

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

  }
}
```

QUESTION 20 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 21
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:

```
1
2  import java.io.*;
3  import java.util.*;
4  public class Hello {
5  class Greeting {
6      void sayHi() {
7          System.out.println("Hello world");
8      }
9  }
10 public static void main(String... args) {
11     Hello myH = new Hello();
12 Hello.Greeting myG = myH.new Greeting();
13 myG.sayHi();
14 }
15 }
```

Console 3 | Console 4

```
Hello world

Completed with exit code: 0
```

QUESTION 22

Which code fragment prints 100 random numbers? A.

```
var r=  new Random();
new DoubleStream(r::nextDouble).limit(100).forEach(System.out::print);
```

```
DoubleStream.generate(Random::nextDouble)
            .limit (100).forFach(System.out::print);
```

```
Doublestream.generate(Random.nextDouble).limit(100).forEach(System.out.print);
```

```
var r = new Random(); DoubleStream.generate(r::nextDouble).limit(100).forEach(System.out::print);
```

B.

C.

D.

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Reference: https://www.javacodegeeks.com/2014/01/java-util-random-in-java-8.html

QUESTION 23
You are working on a functional bug in a tool used by your development organization. In your investigation, you find that the tool is executed with a security policy file containing this grant.

```
grant codebase "file:${klib.home}/j2se/home/klib.jar" {
    permission java.security.AllPermission;
};
```

What action should you take?

A. Nothing, because it is an internal tool and not exposed to the public.
B. Remove the grant because it is excessive.
C. Nothing, because it is not related to the bug you are investigating.
D. File a security bug against the tool referencing the excessive permission granted.
E. Nothing, because listing just the required permissions would be an ongoing maintenance challenge.

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Reference: https://wiki.sei.cmu.edu/confluence/display/java/ENV03-J.+Do+not+grant+dangerous+combinations+of+permissions

QUESTION 24
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 `requiresmain`; 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`.

Correct Answer: BD
Section: (none)
Explanation

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

QUESTION 25
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:

```
1
2  import java.io.*;
3  import java.util.*;
4  class Hello {
5
6
7      enum Color implements Serializable {
8        R(1), G(2), B(3);
9        int c;
10       private Color (int c) {
11         this.c = c;
12       }
13     }
14 }
```

QUESTION 26 `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:

```
1
2  import java.io.*;
3  import java.util.*;
4  class Hello {
5  public static void main(String[] args) {
6
7      var numbers = List.of(0,1,2,3,4,5,6,7,8,9);
8    double avg = numbers.parallelStream().mapToInt (m -> m).average().getAsDouble();
9
10     }
11 }
```

QUESTION 27
Given:

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

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:

```
1
2  import java.io.*;
3  import java.util.*;
4  import java.util.stream.Stream;
5  import java.util.function.Function;
6  import java.util.function.UnaryOperator;
7
8  class Hello {
9  public static void main(String[] args) {
10
11   UnaryOperator<String> function = String::toUpperCase;
12    List<String>fruits = new ArrayList<>(List.of("apple", "orange", "banana"));
13   fruits.replaceAll(function);
14
15     }
16 }
```

QUESTION 28
Given:

```
try {
  // line 1
  lines.map(l -> l.toUpperCase())
    .forEach (line --> {
       try {
          Files.write(Paths.get("outputFile_to_path"),
line.getBytes(),StandardOpenOption.CREATE);
       } catch (IOExeption e) {
         e.printStackTrace();
       }
    });
} catch (IOException e) {
  e.printStackTrace();
}
```

You want to obtain the Stream object on reading the file.

Which code inserted on line 1 will accomplish this?

A. `var lines = Files.lines(Paths.get(INPUT_FILE_NAME));`
B. `Stream lines = Files.readAllLines(Paths.get(INPUT_FILE_NAME));`
C. `var lines = Files.readAllLines(Paths.get(INPUT_FILE_NAME));`
D. `Stream<String> lines = Files.lines(INPUT_FILE_NAME);`

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 29
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 30
Given:

```
public class Employee {
   private String name;
   private LocalDate birthday;
   // the constructors, getters, and setters methods go here
}
```

and

```
List<Employee> roster = new ArrayList<>();
// ...
Predicate<Employee> y = (Employee e) -> e.getBirthday()
    .isBefore(IsoChronology.INSTANCE.date(1989, 1, 1));
Set<String> s1 = roster.stream()
// Line 1
```

Which code fragment on line 1 makes the `s1` set contain the names of all employees born before January 1, 1989? A.

```
.collect(Collectors.partitioningBy(y))
.get(true)
.stream()
.map(Employee::getName)
.collect(Collectors.toCollection(TreeSet::new));
```

B.

```
.collect(Collectors.partitioningBy(y))
.get(true)
.map(Employee::getName)
.collect(Collectors.toSet());
```

C.

```
.collect(Collectors.partitioningBy(y, Collectors.mapping(
    Employee::getName, Collectors.toSet())));
```

D.

```
.collect(Collectors.partitioningBy(y, Collectors.groupingBy(
    Employee::getName, Collectors.toCollection(TreeSet::new))));
```

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 31
Given:

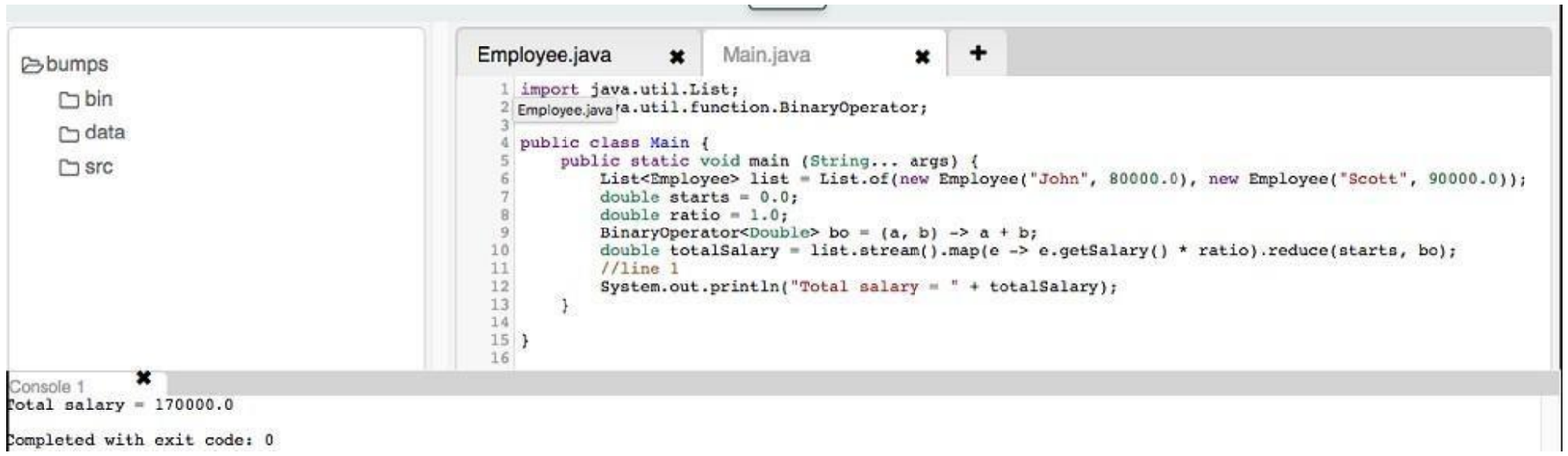
```
import java.util.List;
import java.util.function.BinaryOperator;
public class Main {
   public static void main(String... args) {
      List<Employee> list = List.of(new Employee("John", 80000.0), new Employee("Scott",
90000.0));
      double starts = 0.0;
      double ratio = 1.0;
      BinaryOperator<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:



QUESTION 32 Which interface in the `java.util.function` package will return a void return type?

A. `Supplier`
B. `Predicate`
C. `Function`
D. `Consumer`

Correct Answer: D
Section: (none)
Explanation

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

QUESTION 33
Given:

```
public class MyResource {
    public MyResource () {
    }
    // Resource methods
}
```

You want to use the `myResource` class in a try-with-resources statement.

Which change will accomplish this?

A. Extend `AutoCloseable` and override the `close` method.
B. Implement `AutoCloseable` and override the `autoClose` method.
C. Extend `AutoCloseable` and override the `autoClose` method.
D. Implement `AutoCloseable` and override the `close` method.

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

Reference: https://stackabuse.com/the-try-with-resources-statement-in-java/

QUESTION 34
Given:

```
@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. C.

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

D.

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

E.

```
@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 () {}
}
```

Correct Answer: CD
Section: (none)
Explanation

Explanation/Reference:

QUESTION 35
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");
      }
   }
}
```

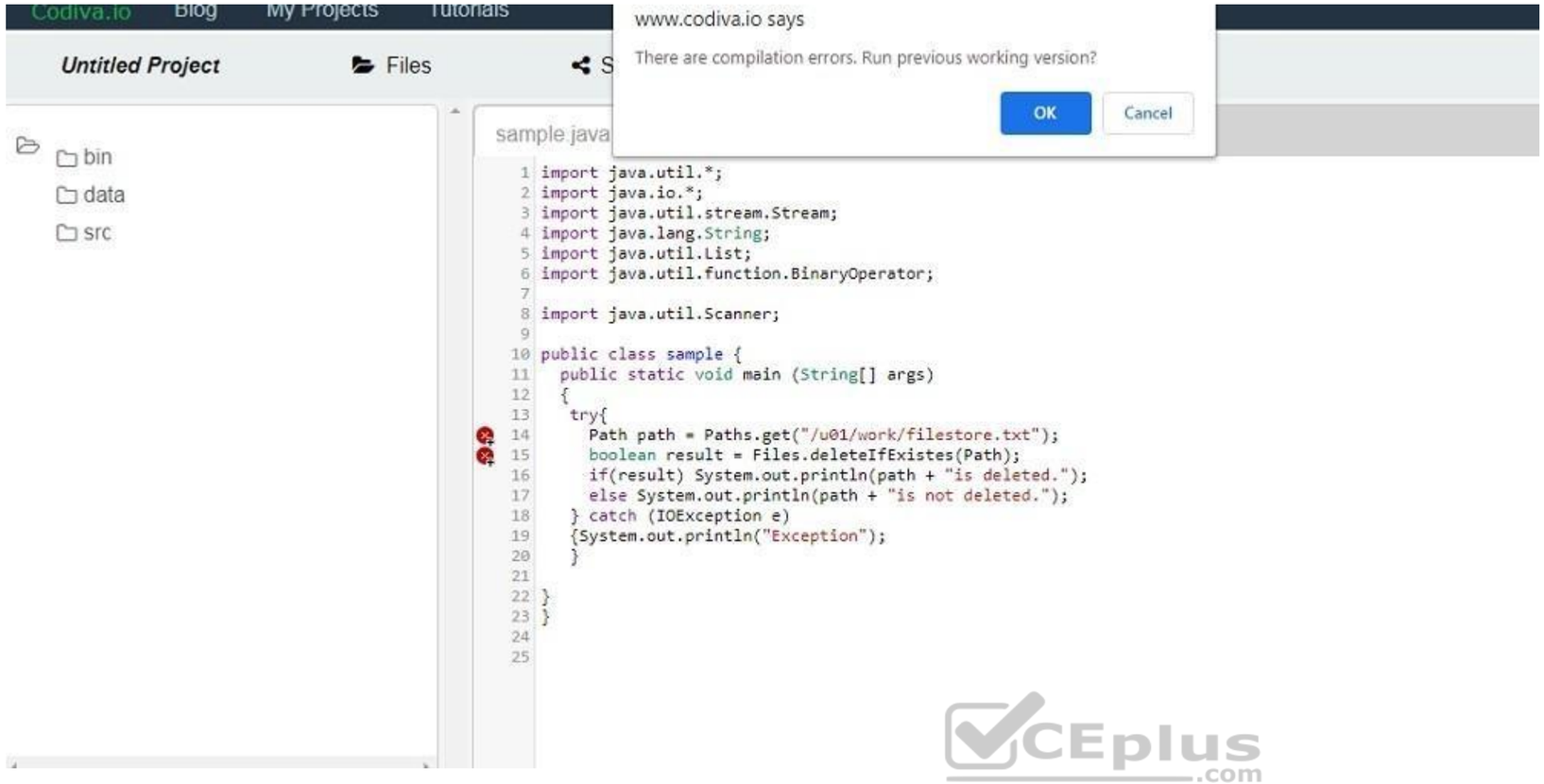
Assume the file on `path` does not exist.

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:



QUESTION 36
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 37
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:

```
Console 4
Error: Could not find or load main class CustomType
Caused by: java.lang.ClassNotFoundException: CustomType
```

QUESTION 38 Which statement about a functional interface is true?

A. It must be defined with the `public` access modifier.

B. It must be annotated with `@FunctionalInterface`.

C. It is declared with a single `abstract` method.

D. It is declared with a single `default` method.

E. It cannot have any `private` methods and `static` methods.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Reference: https://www.geeksforgeeks.org/functional-interfaces-java/

QUESTION 39
Given:

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

And the command: `java`

`Main Helloworld`

What is the result ?

A. `Input: Echo:`
B. `Input: Helloworld Echo: Helloworld`
C. `Input:`
   Then block until any input comes from `System.in`.
D. `Input:`
   `Echo: Helloworld`
E. A `NullPointerException` is thrown at run time.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Explanation:

QUESTION 40
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:

```
13
14 public class Main {
15   public static void main (String[] args) {
16     public class X {
17
18     }
19
20   public final class Y extends X {
21
22   }
23   }
24
```

QUESTION 41 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 42
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
```

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:
Explanation:

```
1  import java.util.*;
2  import java.io.*;
3  import java.lang.Thread;
4  import java.util.ArrayList;
5  import java.util.LinkedList;
6  import java.util.List;
7  import java.util.function.Consumer;
8
9  public class FunctionalInterfaceTest {
10  public static void main (String[] args) {
11         List fruits = Arrays.asList("apple", "orange", "banana");
12         Consumer<String> c = System.out::print;
13         Consumer<String> output = c.andThen(x -> System.out.println(":" + x.toUpperCase()));
14
15    fruits.forEach(output);
16
17   }
18 }
```

QUESTION 43
Given:

```
public class Test {
   public static void doThings() throws GeneralException {
      try {
         throw new RuntimeException("Someting happened");
      }  catch (Exception e) {
         throw new SpecificException(e.getMessage());
      }
   }
   public static void main(String args[]) {
      try{
         Test.doThings();
      }  catch (Exception e) {
         System.out.println(e.getMessage());
      }
   }
}
class GeneralException /* line 1 */ {
   public GeneralException(String s) { super(s); }
}
class SpecificException /* line 2 */ {
   public SpecificException(String s) { super(s); }
}
```

Which option should you choose to enable the code to print `Something happened`?

A. Add `extends GeneralException` on line 1. Add `extends Exception` on line 2.
B. Add `extends SpecificException` on line 1. Add `extends GeneralException` on line 2.
C. Add `extends Exception` on line 1.
   Add `extends Exception` on line 2.
D. Add `extends Exception` on line 1.
   Add `extends GeneralException` on line 2.

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation:

```
1  import java.util.*;
2  import java.io.*;
3  import java.lang.Thread;
4  import java.util.ArrayList;
5  import java.util.LinkedList;
6  import java.util.List;
7
8  public class Test {
9
10   public static void doThings() throws GeneralException {
11     try{
12       throw new RuntimeException("Something happened");
13     } catch (Exception e) {
14       throw new SpecificException (e.getMessage());
15
16     }
17   }
18
19   public static void main(String args[]) {
20     try{
21       Test.doThings();
22     }catch (Exception e) {
23         System.out.println(e.getMessage());
24       }
25     }
26     class GeneralException extends Exception {
27       public GeneralException(String s) { super(s); }
28     }
29   class SpecificException extends GeneralException {
30     public SpecificException(String s) { super(s);}
31   }
32 }
```

QUESTION 44
Given:

```
List<Reader> dataFiles = new ArrayList<>();
File indexFile = new File("MyIndex.idx");
try (BufferedReader indexReader =
      new BufferedReader(new FileReader(indexFile))) {
   for(String file = indexReader.readbine(); file != null;
      file = indexReader.readLine()) {
    BufferedReader dataReader = new BufferedReader (
      new FileReader(new File(file))); // Line 1
    dataFiles.add(dataReader); // Line 2
    processData(dataReader); // Line 3
   }
}  catch (IOException ex) {
   ...
}  finally {
   for(Reader r : dataFiles) {
      try {
         r.close();
      } catch (IOException ex) (
        ...
      } // Line 4
   }
}
```

What will secure this code from a potential Denial of Service condition?

A. After Line 4, add `indexReader.close()`.
B. On Line 3, enclose `processData(dataReader)` with `try with resources`.
C. After Line 3, add `dataReader.close()`.
D. On Line 1, use `try with resources` when opening each `dataReader`.
E. Before Line 1, check the size of `dataFiles` to make sure it does not exceed a threshold.

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 45 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?

```
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;
}
```

A.

B. C.

D.

Correct Answer: C
Section: (none)
Explanation

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

QUESTION 46
Given:

```
String originalPath = "data\\projects\\a-project\\..\\..\\another-project";
Path path = Paths.get(originalPath);
System.out.print(path.normalize());
```

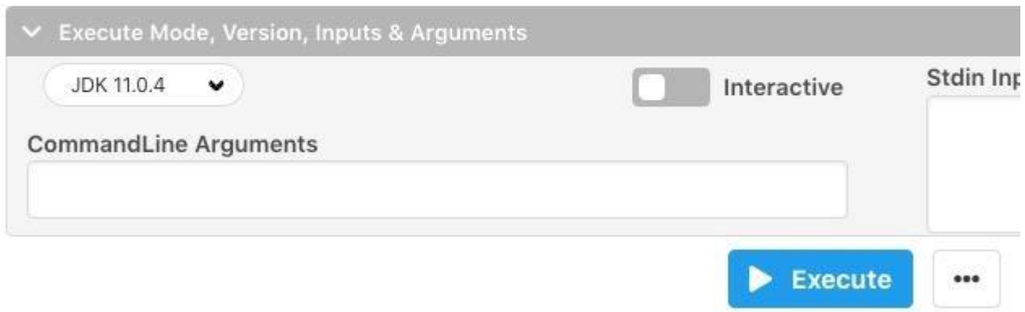
What is the result?

A. `data\another-project`
B. `data\projects\a-project\another-project`
C. `data\\projects\\a-project\\..\\..\\another-project`
D. `data\projects\a-project\..\..\another-project`

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation:

```
1  import java.util.*;
2  import java.io.*;
3  import java.nio.file.*;
4
5  public class Test {
6
7    public static void main(String[] args) {
8      String originalPath = "data\\projects\\a-project\\..\\..\\another-project";
9  Path path = Paths.get(originalPath);
10 System.out.print(path.normalize());
11   }
12 }
```



QUESTION 47
Given:

```
public class Main {
   public static void main(String[] args) {
      Consumer consumer = msg -> System.out::print; // line 1
      consumer.accept("Hello Lambda !");
   }
}
```

This code results in a compilation error.

Which code should be inserted on line 1 for a successful compilation?

A. `Consumer consumer = msg -> { return System.out.print(msg); };`
B. `Consumer consumer = var arg -> {System.out.print(arg);};`
C. `Consumer consumer = (String args) -> System.out.print(args);`
D. `Consumer consumer = System.out::print;`

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation:

```java
import java.util.*;
import java.io.*;
import java.nio.file.*;
import java.util.List;
import java.util.function.Consumer;

public class Main {

  public static void main(String[] args) {
    Consumer consumer = System.out::print;
    consumer.accept("Hello Lambda !");
  }
}
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.16 sec(s), Memory: 32896 kilobyte(s)

```
Hello Lambda !
```

QUESTION 48

Given:

```
int arr[][] = {{5,10},{8,12},{9,3}};
long count = Stream.of(arr)
                   .flatMapToInt(IntStream::of)
                   .map(n -> n + 1)
                   .filter(n -> (n % 2 == 0))
                   .peek(System.out::print)
                   .count();
System.out.println(" " + count);
```

What is the result?

A. `6910 3`
B. `10126 3`
C. `3`
D. `6104 3`

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
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;
import java.util.stream.Stream;
import java.util.stream.IntStream;


public class Main {

  public static void main(String[] args) {
    int arr[][] = {{5,10}, {8,12}, {9,3}};
    long count = Stream.of(arr)
        .flatMapToInt(IntStream::of)
        .map (n -> n + 1)
        .filter(n -> (n % 2 == 0))
        .peek(System.out::print)
        .count();
    System.out.println(" " + count);
  }
}
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.32 sec(s), Memory: 34220 kilobyte(s)

```
6104 3
```

QUESTION 49
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 50** 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 deserializedC. 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 51
Given:

```
1.  void insertionSort(int values[]) {
2.       int n = values.length;
3.       for (int j = 1; j < n; j++) {
4.            int tmp = values[j];
5.            int i = j - 1;
6.            while ( (i > -1) && (values[i] > tmp) ) {
7.               values[i + 1] = values[i];
8.                 i--;
9.            }
10.           values[i + 1] = tmp;
11.      }
12.  }
```

After which line can we insert `assert i < 0 || values[i] <= values[i + 1];` to verify that the `values` array is partially sorted?

A. after line 8 B.
after line 6
C. after line 5
D. after line 10

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
Explanation:

```
1  import java.util.*;
2  import java.io.*;
3  import java.lang.Thread;
4  import java.util.ArrayList;
5  import java.util.LinkedList;
6  import java.util.List;
7  import java.util.function.Consumer;
8  import java.util.stream.Stream;
9  import java.util.stream.IntStream;
10
11
12 public class Main {
13
14
15     void insertionSort (int values[]) {
16       int n = values.length;
17         for (int j = 1; j < n; j++) {
18           int tmp = values[j];
19
20           int i = j - 1;
21           assert i < 0 || values[i] <= values[i + 1];
22           while ((i > 1) && (values[i] > tmp) ) {
23             values[i + 1] = values[i];
24             i--;
25
26           }
27           values[i + 1] = tmp;
28
29         }
30     }
31 }
```

QUESTION 52
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);
```

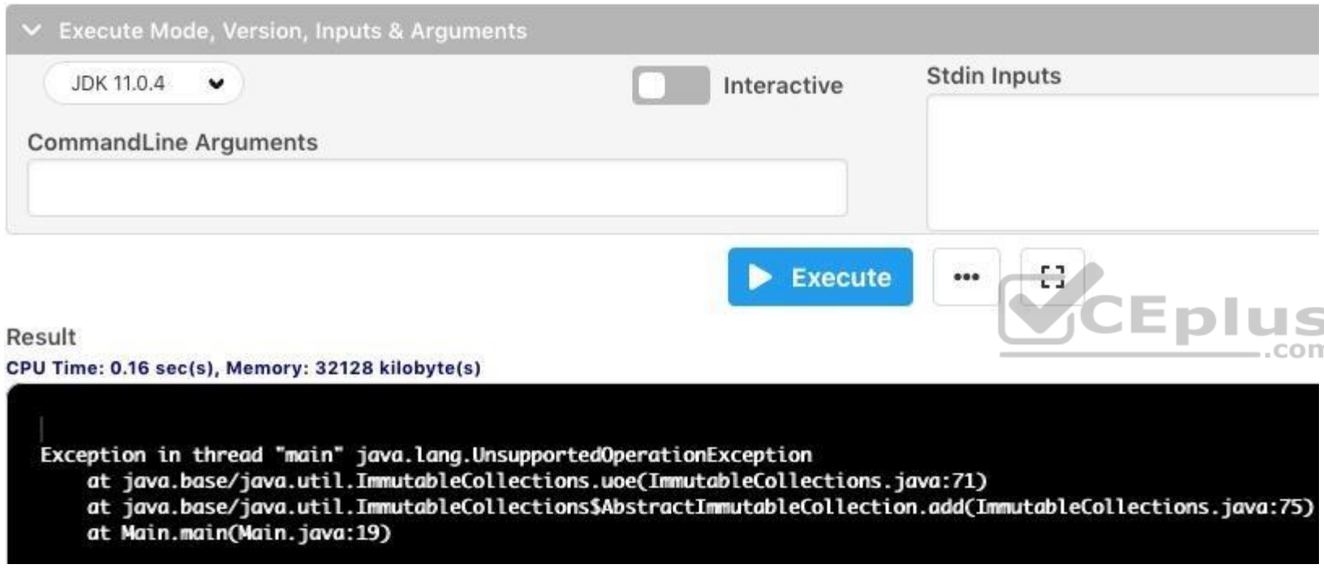
What is the result?

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:

```
12 public class Main {
13   public static void main(String[] args) {
14
15     List<String> list1 = new ArrayList<>();
16     list1.add("A");
17     list1.add("B");
18     List list2 = List.copyOf(list1);
19     list2.add("C");
20     List<List<String>> list3 = List.of(list1, list2);
21     System.out.println(list3);
22   }
23
24   }
25
```



QUESTION 53
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 54
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:

```
 8 public class Secret {
 9     public static void main(String[] args) {
10     Integer[] intArray = {1, 2, 3, 4, 5};
11     List<Integer> list =
12     new ArrayList<> (Arrays.asList (intArray));
13     list.parallelStream()
14     .forEachOrdered(e -> System.out.print(e + " "));
15     }
16 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

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

```
1 2 3 4 5
```

QUESTION 55

Given the contents:

`MessageBundle.properties` file:

```
message=Hello
```

`MessageBundle_en.properties` file:

```
message=Hello (en)
```

`MessageBundle_US.properties` file:

```
message=Hello (US)
```

`MessageBundle_en_US.properties` file:

```
message=Hello (en_US)
```

`MessageBundle_fr_FR.properties` file:

```
message=Bonjour
```

and the code fragment:

```
Locale.setDefault(Locale.FRANCE);
Locale currentLocale = new Locale.Builder().setLanguage("en").build();

ResourceBundle messages = ResourceBundle.getBundle("MessageBundle", currentLocale);
System.out. println(messages.getString("message"));
```

Which file will display the content on executing the code fragment?

A. `MessageBundle_en_US.properties`

B. `MessageBundle_en.properties`

C. `MessageBundle_fr_FR.properties`
D. `MessageBundle_US.properties`
E. `MessageBundle.properties`

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Reference: https://www.javatpoint.com/ResourceBundle-class

QUESTION 56
Given:

```
public class Main {
   public static void main(String[] args) {
      var numbers = List.of(1,2,3,4,5,6,7,8,9,10);
      Optional<Integer> result = numbers.stream().filter(x -> x % 3 != 0).reduce((i, j)
-> i + j);
      result.ifPresent(System.out::print); // line 1
```

Which is true about line 1?

A. If the value is not present, a `NoSuchElementException` is thrown at run time.
B. It always executes the `System.out::print` statement.
C. If the value is not present, a `NullPointerException` is thrown at run time.
D. If the value is not present, nothing is done.

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
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;
import java.util.stream.Stream;
import java.util.stream.IntStream;
import java.util.Optional;


public class Main {
    public static void main(String[] args) {
        var numbers = List.of(1,2,3,4,5,6,7,8,9,10);
        Optional<Integer> result = numbers.stream().filter (x -> x % 3 != 0).reduce( (i, j) -> i + j);

    }
}
```

Result

CPU Time: 0.18 sec(s), Memory: 33380 kilobyte(s)

```
JDoodle in Action.... Running the program...
```

QUESTION 57
Given:

```
List<String> list1 = new LinkedList<String>();
Set<String> hs1 = new HashSet<String>();
String[] v = {"a", "b", "c", "b", "a"};
for (String s: v) {
     list1.add(s);
     hs1.add(s);
}
System.out.print(hs1.size() + " " + list1.size() + " ");
HashSet hs2 = new HashSet(list1);
LinkedList list2 = new LinkedList(hs1);
System.out.print(hs2.size() + " " + list2.size());
```

What is the result?

A. 3 5 3 3 B.
3 3 3 3 C. 3
5 3 5
D. 5 5 3 3

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
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;
import java.util.stream.Stream;
import java.util.stream.IntStream;
import java.util.Optional;


public class Main {
    public static void main(String[] args) {
        List<String> list1 = new LinkedList<String>();
        Set<String> hs1 = new HashSet<String>();
        String[] v = {"a", "b", "c", "b", "a"};
        for (String s: v) {
            list1.add(s);
            hs1.add(s);
        }
        System.out.println(hs1.size() + "" + list1.size() + "");
        HashSet hs2 = new HashSet(list1);
        LinkedList list2 = new LinkedList(hs1);
        System.out.print(hs2.size() + "" + list2.size());

    }
}
```

Result

CPU Time: 0.28 sec(s), Memory: 36204 kilobyte(s)

```
35
33
```

QUESTION 58

Given:

```java
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:

```
1  import java.util.*;
2  import java.io.*;
3  import java.lang.Thread;
4  import java.util.ArrayList;
5  import java.util.LinkedList;
6  import java.util.List;
7  import java.util.function.Consumer;
8  import java.util.stream.Stream;
9  import java.util.stream.IntStream;
10 import java.util.Optional;
11
12
13 public class Main {
14     class Student {
15         String classname;
16         public Student (String classname) {
17             this.classname = classname;
18         }
19
20             }
21         public static void main (String[] args) {
22             var student = new Student ("Biology");
23         }
24 }
```

QUESTION 59
Given:

```
public class Employee {
   private String name;
   private String locality;
   /* the constructor, getter and setter methods code goes here */
}
```

and:

```
8. List<Employee> roster = new ArrayList<>();
9. long empCount = roster.stream()
10. /* insert code here */
11. System.out.print(empCount);
```

Which code, when inserted on line 10, prints the number of unique localities from the `roster` list?

A.
```
.map(Employee::getLocality)
.distinct()
.count();
```

B. `map(e -> e.getLocality()) .count();`

C. `.map(e -> e.getLocality())`

```
    .collect(Collectors.toSet())
    .count();
```

D.
```
.filter(Employee::getLocality)
.distinct()
.count();
```

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Reference: https://developer.android.com/reference/android/location/Address

QUESTION 60
Given the `Person` class with `age` and `name` along with getter and setter methods, and this code fragment:

```
List<Person> persons = new ArrayList(List.of(new Person(44,"Tom"),
                                              new Person(40,"Aman"),
                                              new Person(40,"Peter")));
persons.sort(Comparator.comparing((Person::getAge))
                                        .thenComparing(Person::getName)
                                        .reversed());
persons.forEach(p1->System.out.print(" "+p1.getName()));
```

What will be the result?

A. `Aman Tom Peter`
B. `Tom Aman Peter`
C. `Aman Peter Tom`
D. `Tom Peter Aman`

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 61 Which three guidelines are used to protect confidential information?
(Choose three.)

A. Limit access to objects holding confidential information.
B. Clearly identify and label confidential information.
C. Manage confidential and other information uniformly.
D. Transparently handle information to improve diagnostics.
E. Treat user input as normal information.
F. Validate input before storing confidential information.
G. Encapsulate confidential information.

Correct Answer: ADF
Section: (none)
Explanation

Explanation/Reference:
Reference: https://danielkvist.net/code/java-secure-coding-guidelines

QUESTION 62
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.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 63 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 64 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 65
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 66 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?

```
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()));
```

A.

B. C.

D.

Correct Answer: C

Section: (none)
Explanation

Explanation/Reference:

QUESTION 67 Which two safely validate inputs? (Choose two.)

A. Delegate numeric range checking of values to the database.
B. Accept only valid characters and input values.
C. Use trusted domain-specific libraries to validate inputs.
D. Assume inputs have already been validated.
E. Modify the input values, as needed, to pass validation.

Correct Answer: AB
Section: (none)
Explanation

Explanation/Reference:
Reference: https://stackoverflow.com/questions/3059333/validating-input-using-java-util-scanner

QUESTION 68
Consider this method declaration:

```
void setSessionUser(Connection conn, String user) throws SQLException {
     Statement stmt = conn.createStatement();
     String sql = <EXPRESSION>;
     stmt .execute();
}
```

A) `"SET SESSION AUTHORIZATION " + user`
B) `"SET SESSION AUTHORIZATION " + stmt.enquoteIdentifier(user)` Is A or B the correct replacement for `<EXPRESSION>` and why?

A. A, because it sends exactly the value of `user` provided by the calling code.
B. B, because enquoting values provided by the calling code prevents SQL injection.
C. A and B are functionally equivalent.
D. A, because it is unnecessary to enclose identifiers in quotes.
E. B, because all values provided by the calling code should be enquoted.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
Reference:
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&ved=2ahUKEwj7ycO80fLoAhVHPcAKHcoLC9cQFjADegQIAxAB&url=ftp%3A%2F%2Fftp.software.ibm.com%2Fps%2Fproducts%2Fdb2%2Finfo%2Fvr9%2Fpdf%2Fletter%2Fen_US%2Fdb2s2e90.pdf&usg=AOvVaw2VqpeEh5HpbeXfa0OB5Lec

QUESTION 69 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 70
Which two statements correctly describe capabilities of interfaces and abstract classes? (Choose two.)

A. Interfaces cannot have `protected` methods but abstract classes can.
B. Both interfaces and abstract classes can have `final` methods.
C. Interfaces cannot have instance fields but abstract classes can.
D. Interfaces cannot have `static` methods but abstract classes can.
E. Interfaces cannot have methods with bodies but abstract classes can.

Correct Answer: AC
Section: (none)
Explanation

Explanation/Reference:
Reference: https://www.guru99.com/interface-vs-abstract-class-java.html

QUESTION 71
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:

```
12 public class Person {
13     public static void main (String[] args) {
14         final List<String> fruits =
15         List.of("Orange", "Apple", "Lemmon", "raspberry");
16         final List<String> types =
17         List.of("Juice", "Pie", "Ice", "Tart");
18         final var stream =
19         IntStream.range(0, Math.min(fruits.size(), types.size()))
20         .mapToObj ((i) -> fruits.get(i) + " " + types.get(i) );
21         stream. forEach(System.out::println);
22     }
23
24 }
```

Result

compiled and executed in 1.227 sec(s)

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

QUESTION 72
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 73
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 74
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=java%20pattern%20formats%20eeee%20d%2Bth%20of%2B%20MMMM%20yyyy&f=false

QUESTION 75 Which two statements independently compile? (Choose two.)

A. `List<? super Short> list = new ArrayList<Number>();`
B. `List<? super Number> list = new ArrayList<Integer>();`
C. `List<? extends Number> list = new ArrayList<Byte>();`
D. `List<? extends Number> list = new ArrayList<Object>();`
E. `List<? super Float> list = new ArrayList<Double>();`

Correct Answer: AC
Section: (none)
Explanation

Explanation/Reference:
Explanation:

```
1  import java.util.*;
2  import java.text.*;
3  import java.io.*;
4  import java.lang.Thread;
5  import java.util.ArrayList;
6  import java.util.LinkedList;
7  import java.util.List;
8  import java.util.function.Consumer;
9  import java.util.stream.Stream;
10 import java.util.stream.IntStream;
11 import java.util.Optional;
12
13 public class Intel {
14     public static void main (String[] args) {
15 List<? extends Number> list = new ArrayList<Byte>()
16 }
17 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

Result

compiled and executed in 1.173 sec(s)

QUESTION 76
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:

```
13 public class Main {
14 enum Letter {
15     ALPHA(100), BETA(200), GAMMA(300);
16     int v;
17     Letter(int v) { this.v = v; }
18    public String toString() { return String.valueOf(v); }
19
20
21
22 }
23 public static void main (String[] args) {
24  System.out.println(Letter.values() [1]);
25 }
26 }
27
28
```

Result

compiled and executed in 1.099 sec(s)

```
200
```

QUESTION 77 Given the
code fragment:

```
Path source = Paths.get("/repo/a/a.txt");
Path destination = Paths.get("/repo");
Files.move(source, destination);     // line 1
Files.delete (source);             // line 2
```

Assuming the `source` file and `destination` folder exist, what Is the result?

A. A `java.nio.file.FileAlreadyExistsException` is thrown on `line 1`.
B. A `java.nio.file.NoSuchFileException` is thrown on `line 2.`
C. A copy of `/repo/a/a.txt` is moved to the `/repo` directory and `/repo/a/a.txt` is deleted.
D. `a.txt` is renamed repo.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 78
Given:

```
List<String> longlist = List.of("Hello","World","Beat");
List<String> shortlist = new ArrayList<>();
```

Which code fragment correctly forms a short list of words containing the letter "e"? A.

```
longList.stream()
    .filter(w -> w.indexOf('e') != -1)
    .parallel()
    .forEach(w -> shortList.add(w));
```

```
longList.parallelStream()
    .filter(w -> w.indexOf('e') != -1)
    .forEach(w -> shortList.add(w));
```

```
shortList = longList.stream()
    .filter(w -> w.indexOf('e') != -1)
    .parallel()
    .collect(Collectors.toList());
```

```
longList.stream()
    .filter(w -> w.indexOf('e') != -1)
    .parallel()
    .collect(shortlist);
```

B.

C.

D.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 79
Given:

```
jdeps -jdkinternals C:\workspace4\SimpleSecurity\jar\classes.jar
```

Which describes the expected output?

A. `jdeps` lists the module dependencies and the package names of all referenced JDK internal APIs. If any are found, the suggested replacements are output in the console.
B. `jdeps` outputs an error message that the `-jdkinternals` option requires either the `-summary` or the `-verbose` options to output to the console.
C. The `-jdkinternals` option analyzes all classes in the `.jar` and prints all class-level dependencies.

D. The `-jdkinternals` option analyzes all classes in the `.jar` for class-level dependencies on JDK internal APIs. If any are found, the results with suggested replacements are output in the console.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
Explanation:
`-jdkinternals` option analyzes all classes in the `.jar` for class-level dependencies on JDK internal APIs. If any are found, the results with suggested replacements are output in the console.

QUESTION 80
Given:

```
public class Main {
   public static void main(String[] args) {
      List 1 = new ArrayList();
      l.add("hello");
      l.add("world");
      print(1);
   }
   private static void print(List<String>... args) {
        for (List<String> str : args) {
          System.out.println (str);
        }
   }
}
```

Which annotation should be used to remove warnings from compilation?

A. `@SuppressWarnings` on the `main` and `print` methods
B. `@SuppressWarnings("unchecked")` on `main` and `@SafeVarargs` on the `print` method
C. `@SuppressWarnings("rawtypes")` on `main` and `@SafeVarargs` on the `print` method
D. `@SuppressWarnings("all")` on the `main` and `print` methods

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
Explanation:

```
13 @SuppressWarnings("unchecked")
14 public class Main {
15
16     public static void main(String[] args) {
17
18         List l = new ArrayList();
19         l.add("Hello");
20         l.add("world");
21         print(1);
22
23     }
24
25     private static void print(List<String>... args) {
26         for (List<String> str : args) {
27             System.out.println (str);
28
29         }
30     }
31     @SafeVarargs
32 }
```