

Oracle

1Z0-819 Exam

Java SE 11 Developer

Version : 5.0
[Total Questions : 215]

Question: 1

Given the code fragment:

```
public class FizzBuzz {  
    public static String convert(int x) {  
        if (x % 15 == 0) return "FizzBuzz";  
        else if (x % 3 == 0) return "Fizz";  
        else if (x % 5 == 0) return "Buzz";  
        else return Integer.toString(x);  
    }  
  
    public static void main(String[] args) {  
        for (int i = 1; i < 101; i++) {  
            System.out.println(convert(i));  
        }  
    }  
}
```

Which code fragment replaces the for statement?

- A. IntStream.rangeClosed(1, 100).map(FizzBuzz::convert).forEach(System.out::println);
- B. IntStream.ranged(1, 100).map(FizzBuzz::convert).forEach(System.out::println);
- C. intstream.rangeclosed(1, 100).mapToObj{FizzBuzz::convert}.forEach(System.out::println);
- D. IntStream.range(1, 100).mapToObj(FizzBuzz::convert).forEach(System.out::println);

Answer: A

Question: 2

Given:

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

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

- A . Abacus aba = (int m, int n) -> { m * n };
- B . Abacus aba = (int e, int f) -> { return e * f; };
- C . Abacus aba = (a, b) -> a * b;
- D . Abacus aba = v, w -> x * y;

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

Answer: C, E

Question: 3

Given:

```
public interface Copier {  
    public default void print(String msg){  
        System.out.println("Message from Copier: "+msg);  
    }  
}
```

and

```
public abstract class AbstractCopier {  
    protected void print(String load){  
        System.out.println("Message from Abstract Copier: "+load);  
    }  
}
```

and

```
public class TestImpl extends AbstractCopier implements Copier {  
    public static void main(String[] args){  
        TestImpl test = new TestImpl();  
        test.print("Attempt00");  
    }  
}
```

What is the output?

- A . A compilation error is thrown.
- B . Message from Copier: Attempt00
- C . Message from Abstract Copier: Attempt00
- D . A runtime error is thrown.

Answer: A

Question: 4

Given the code fragment:

Which two code snippets inserted independently inside print method print Mondial: domainmodal?

- A . prefix + name
- B . prefix + getName
- C . new Main () .prefix + new Main().name
- D . prefix + Main, name

- E . Main.prefix + Main.name
- F . Main.prefix + Main.getName()

Answer: C, D

Question: 5

Your organization makes mlib.jar available to your cloud customers. While working on a code cleanup project for mlib.jar, you see this method by customers:

```
public void enableService(String hostName, String portNumber) throws IOException {  
    this.transportSocket = new Socket(hostName, portNumber);  
}
```

What security measures should be added to this method so that it meets the requirements for a customer accessible method?

A.

Insert this code before the call to new Socket:
hostName = new String(hostName);
portNumber = new String(portNumber);

- B . Create a method that validates the hostName and portNumber parameters before opening the socket.
- C . Make enableService private.
- D . Enclose the call to new Socket In an AccessController.doPrivileged block.

Answer: D

Question: 6

Given

:

```
public class Point {  
    @JsonField(type=JsonField.Type.STRING, name="name")  
    private String _name;  
  
    @JsonField(type=JsonField.Type.INT)  
    private int x;  
  
    @JsonField(type=JsonField.Type.INT)  
    private int y;  
}
```

What is the correct definition of the JsonField annotation that makes the Point class compile?

A)

```
@Target(ElementType.FIELD)
@interface JsonField {
    String name() default "";
    enum Type {
        INT, STRING, BOOLEAN
    };
    Type type();
}
```

B)

```
@interface JsonField {
    String name();
    enum Type {
        INT, STRING, BOOLEAN
    };
    Type type();
}
```

C)

```
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)
@interface JsonField {
    String name() default "";
    enum Type {
        INT, STRING, BOOLEAN
    };
    Type type();
}
```

- A . Option A
- B . Option B
- C . Option C

Answer: A

Question: 7

Given the code fragment:

```
9. Integer[] ints = {1,2,3,4,5,6,7};
10. var list = Arrays.asList(ints);
11. UnaryOperator<Integer> uo = x -> x * 3;
12. list.replaceAll(uo);
```

Which can replace line 11?

- A . UnaryOperator<Integer> uo = (var x) -> (x * 3);
- B . UnaryOperator<Integer> uo = var x -> { return x * 3;};

C . UnaryOperator<Integer>uo = x -> { return x * 3; };

D . UnaryOperator<Integer>uo = (int x) -> x * 3;

Answer: A

Question: 8

Given the code fragment:

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

You want to examine whether path is a directory.

Which code inserted on line 1 will accomplish this?

A . BasicFileAttributes attributes = Files.isDirectory (path);

B . BasicFileAttributes attributes =Files.getAttribute (path, "isDirectory");

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

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

Answer: D

Question: 9

Given the code fragment:

```
public class Test {
    class L extends Exception { }
    class M extends L { }
    class N extends RuntimeException { }
    public void p() throws L { throw new M(); }
    public void q() throws N { throw new N(); }
    public static void main(String[] args) {
        try {
            Test t = new Test();
            t.p();
            t.q();
        } /* line 1 */
        System.out.println("Exception caught");
    }
}
```

What change on line 1 will make this code compile?

- A . Add catch (L | N e).
- B . Add catch (L | M N e).
- C . Add catch (L e).
- D . Add catch (N | L | M e).
- E . Add catch (M | L e).

Answer: C

Question: 10

Given the code fragment:

```
char[][] arrays = {{'g', 'j'}, {'h', 'k'}, {'i', 'l'}};
for (char[] xx : arrays) {
    for (char yy : xx) {
        System.out.print(yy);
    }
    System.out.print(" ");
}
```

What is the result?

- A . An ArrayIndexOutOfBoundsException is thrown at runtime.
- B . The compilation fails.
- C . gh ij kl
- D . gj hk il**
- E . ghi jkl

Answer: A

Question: 11

And the code fragment:

```
Resource resource = new Resource();
Worker worker = new Worker();
Thread t1 = new Thread(() -> resource.processWork(worker));
Thread t2 = new Thread(() -> worker.consumeResource(resource));

t1.start();
t2.start();
```

Which situation will occur on code fragment execution?

- A . Livelock
- B . Deadlock
- C . Race Condition
- D . Starvation

Answer: D

Question: 12

Given:

```
public interface AdaptorFirst {  
    void showFirst();  
}
```

Which three classes successfully override showFirst ()?

A)

```
public abstract class MainClass implements AdaptorFirst {  
    public String showFirst(){  
        return "first";  
    }  
}
```

B)

```
public abstract class MainClass implements AdaptorFirst {  
    public void showFirst(){  
        System.out.println("first");  
    }  
}
```

C)

```
public class MainClass implements AdaptorFirst {  
    void showFirst();  
}
```

D)

```
public class MainClass implements AdaptorFirst {  
    private void showFirst(){  
        System.out.println("first");  
    }  
}
```

E)

```
public abstract class MainClass implements AdaptorFirst {  
    public abstract void showFirst();  
}
```

F)

```
public class MainClass implements AdaptorFirst {  
    public void showFirst(){  
        System.out.println("first");  
    }  
}
```

A . Option A

B . Option B

C . Option C

D . Option D

E . Option E

F . Option F

Answer: C

Question: 13

Given:

```
public class Menu {  
    enum Machine{  
        AUTO("Truck"), MEDICAL("Scanner");  
        private String type;  
        private Machine(String type) {  
            this.type = type;  
        }  
        private void setType(String type) {  
            this.type = type; // line 1  
        }  
        private String getType() {  
            return type;  
        }  
    }  
    public static void main(String[] args) {  
        Machine.AUTO.setType("Sedan"); // line 2  
        for (Machine p : Machine.values()) {  
            System.out.println(p + ": " + p.getType()); // line 3  
        }  
    }  
}
```

A) An exception is thrown at run time.

B)

```
AUTO: Sedan  
MEDICAL: Scanner
```

C) The compilation fails due to an error on line 2.

D) The compilation fails due to an error on line 1.

E)

```
AUTO: Truck  
MEDICAL: Scanner
```

F)

The compilation fails due to an error on line 3.

- A . Option A
- B . Option B
- C . Option C
- D . Option D
- E . Option E
- F . Option F

Answer: A

Question: 14

Given the code fragment:

```
StringBuilder txt1 = new StringBuilder("PPQRRRSSTT");  
int i = 0;  
a:  
while (i < txt1.length()) {  
    char x = txt1.charAt(i);  
    int j = 0;  
    i++;  
    b:  
    while (j < txt1.length()) {  
        char y = txt1.charAt(j);  
        if (i != j && y == x) {  
            txt1.deleteCharAt(j);  
            // line 1  
        }  
        j++;  
    }  
}  
System.out.println(txt1);
```

Which two statement inserted independently at line 1 enable this code to print PRRT?

- A . i--;
- B . continue b;
- C . break b;
- D . j--;
- E . continue a;

F . break a ;

Answer: F

Question: 15

Given:

```
import java.util.*;  
  
public class Main {  
    static Map<String, String> map = new HashMap<>();  
    static List<String> keys =  
        new ArrayList<>(List.of("S", "P", "Q", "R"));  
    static String[] values =  
        {"senate", "people", "of", "rome" };  
  
    static {  
        for(var i = 0; i < keys.size(); i++) {  
            map.put(keys.get(i), values[i]);  
        }  
    }  
  
    public static void main(String[] args) {  
        keys.clear();  
        values = new String[0];  
        System.out.println("Keys: " + keys.size() +  
            " Values: " + values.length +  
            " Map: " + map.size());  
    }  
}
```

What is the result?

- A . Keys: 4 Values: 4 Map: 0
- B . Keys: 4 Values: 4 Map: 4
- C . The compilation fails.
- D . Keys: 0 Values: 0 Map:
- E . Keys: 0 Values: 0 Map: 0

Answer: B

Question: 16

Given the code fragment:

```
Consumer<String> c1 = arg -> System.out.println(arg);  
c1.accept("c1 accepted");  
Consumer<String> c2 = arg -> System.out.println(arg);  
c2.accept("c2 accepted");  
c2.andThen(c1).accept("after then");  
c2.accept("c2 accepted again");
```

What is the result?

A)

```
c1 accepted  
c2 accepted  
and followed by an exception
```

B)

```
c1 accepted  
c2 accepted  
after then  
c1 accepted  
c2 accepted again
```

C)

```
c1 accepted  
c2 accepted  
after then  
c2 accepted again
```

D)

```
c1 accepted  
c2 accepted  
after then  
after then  
c2 accepted again
```

- A . Option A
- B . Option B
- C . Option C
- D . Option D

Answer: C

Question: 17

Given the code fragment:

```
public class City {  
    public static void main(String[] args) {  
        String[] towns = {"boston", "paris", "bangkok", "oman"};  
        Comparator<String> ms = (a, b) -> b.compareTo(a);  
        Arrays.sort(towns, ms);  
        System.out.println(Arrays.binarySearch(towns, "oman", ms));  
    }  
}
```

What is the result?

- A . 2
- B . -1
- C . 1
- D . -3

Answer: A

Question: 18

Given:

```
public class Plant { }  
and  
public class Tulip extends Plant { }  
and  
public class Garden {  
    private static Plant plant;  
    public static void main(String[] args) {  
        plant = new Tulip();  
        feed(plant);  
        feed(plant);  
    }  
    public static void feed(Plant p) {  
        if (p instanceof Tulip) {  
            System.out.println("Take extra care");  
        }  
        p = null;  
    }  
}
```

What is the result?

- A . Take extra care
- B . The program prints nothing.
- C . Take extra care Take extra care
- D . An exception is thrown at runtime

Answer: D

Question: 19

Given:

```

class Item {
    public String name; public int count;
    public Item(String name, int count) {
        this.name = name; this.count = count;
    }
}

```

and the code fragment:

```

public class Test {
    public static void main(String[] args) {
        var items = List.of(new Item("A", 10), new Item("B", 2),
                           new Item("C", 12), new Item("D", 5), new Item("E", 6));
        // line 1
        System.out.println("There is an item for which the variable count is below zero.");
    }
}

```

You want to examine the items list it contains an item for which the variable count is below zero.

Which code fragment at line 1 accomplish this?

- A . If (items.stream () .filter (i -> count < 0) . findFirst ()) {
- B . If (items.stream () .filter (i -> count < 0) . findAny ()) {
- C . If (items.stream () .allmatch (i -> count < 0) < 0)) {
- D . If (items.stream () .anymatch (i -> count < 0) < 0) {

Answer: A

Question: 20

Given:

```

class ConSuper {
    protected ConSuper(){
        this(2);
        System.out.print("3");
    }
    protected ConSuper(int a){
        System.out.print(a);
    }
}

```

and

```

public class ConSub extends ConSuper{
    ConSub(){
        this(4);
        System.out.print("1");
    }
    ConSub(int a) {
        System.out.print(a);
    }
    public static void main (String[] args){
        new ConSub(4);
    }
}

```

What is the result?

- A . 2134
- B . 234
- C . 2341
- D . 214

Answer: A

Question: 21

Given:

```
class MyPersistenceData {  
    String str;  
    private void methodA() {  
        System.out.println("methodA");  
    }  
}
```

You want to implement the jav

- a. Io, serializable interface to the MypersisteneData class.

Which method should be overriden?

- A . The readExternal and writeExternal method
- B . The readExternal method
- C . The writeExternal method
- D . nothing

Answer: A

Question: 22

Given the Customer table structure:

* ID Number Primary Key

* NAME Text Nullable

Given code fragment:

```
12. PreparedStatement stmt = con.prepareStatement("INSERT INTO CUSTOMER VALUES (?,?)");
13. stmt.setInt(1, 42);
14. /* Insert code here */
15. int n = stmt.executeUpdate();
```

Which statement inserted on line 14 sets NAME column to a NULL value?

- A . Stmt.setNull(2, java.sql.Types, VARCHAR);
- B . Stmt.setNull(2 string, class);
- C . Stmt.setNull(2, null);
- D . Stmt.setNull(2, java.lang, string);

Answer: A

Question: 23

How many Thing objects are eligible for garbage collection in line 1?

- A . 3
- B . 2
- C . 0
- D . 1
- E . 4

Answer: D

Question: 24

A company has an existing Java app that includes two Java 8 jar files, sales-3.10. jar and clients-10.2.jar.

The jar file ,sales -8, 10, jar reference packages in clients -10.2 jar, but clients-10.2 jar does not reference packages in sales -8.10, jar.

They have decided to modularize clients-10.2.jar.

Which module-info. Java file would work for the new library version clients-10.3 jar?

A)

```
module com.company.clients{
    uses com.company.clients;
}
```

B)

```
module com.company.clients{
    requires com.company.clients;
}
```

C)

```
module com.company.clients {
    exports com.company.clients.Client;
}
```

D)

```
module com.company.clients {
    exports com.company.clients;
}
```

- A . Option A
- B . Option B
- C . Option C
- D . Option D

Answer: C

Question: 25

Given the code fragment:

```
int i = 0;
for( ; i<10; i++){
    System.out.print(++i + " ");
}
```

What is the result?

- A . 13 5 7 9
- B . 1 3 5 7 9 11
- C . 2 4 6 B 10
- D . 2 4 6 8

Answer: B

Question: 26

Given:

```
public class GameObject {  
    public Object[] move(int x, int y) {  
        System.out.println("Move GameObject");  
        return new Integer[] { x + 10, y + 10};  
    }  
}  
  
and  
  
public class Avatar extends GameObject {  
    public Object[] move(Number x, Number y) {  
        System.out.println("Move Character");  
        return super.move(x.intValue(), y.intValue());  
    }  
    public static void main(String... args) {  
        var character = new Avatar();  
        character.move(10.0, 10.0);  
        character.move(10, 10);  
    }  
}
```

What is the result?

A)

```
Move GameObject  
Move GameObject
```

B)

```
Move Character  
Move GameObject  
Move GameObject
```

C)

```
Move GameObject
```

D)

```
Move GameObject  
Move Character  
Move GameObject
```

- A . Option A
- B . Option B
- C . Option C
- D . Option D

Answer: A

Question: 27

Given:

```
public class Tester {  
    public static void main(String[] args) {  
        String s = "hat at store";  
        int x = s.indexOf("at");  
        s.substring(x + 3);  
        x = s.indexOf("at");  
        System.out.println(s + " " + x);  
    }  
}
```

What is the result?

- A . An indexOutOfBoundsException is thrown at runtime.
- B . At once 0
- C . Hat at store 4
- D . At once 1
- E . Hat at store 1

Answer: E

Question: 28

Given:

```
List<Integer> numbers = List.of(2, 3, 0, 8, 1, 9, 5, 7, 6, 4);  
int sum = numbers.stream().reduce(0, (n, m) -> n + m); // line 1
```

You wants to make the reduction operation parallelized.

Which two modifications will accomplish this?

- A . Replace line 1 with int sum = numbers. Stream (). Interate (0, a -> a+1. Reduce (0, (n m) -> n+m);
- B . Replace line 1with int sum = numbers. ParallelStream (). Reduce (0, (n, m) -> n + m);
- C . Replace line 1 with int sum = numbers. Parallel (). Stream (). Reduce (0, (n, m) -> n + m);
- D . Replace line 1with int sum = number. Stream () . flatMap (a -> a) .reduce (0, (n, m) -> n +m);
- E . Replace line 1with int sum = number.stream. parallel (). Reduce (0, (n, m) -> n + m);

Answer: A, D

Question: 29

There is a copyServiceAPI that has the org.copyservice.spi.Copy interface

To use this service in a module, which module- info.java would be correct?

A)

```
module CopyConsumer {  
    requires CopyServiceAPI;  
    uses org.copyservice.spi.Copy;  
}
```

B)

```
module CopyConsumer {  
    requires transitive org.copyservice.spi.Copy;  
}
```

C)

```
module CopyConsumer {  
    requires org.copyservice.spi.Copy;  
}
```

D)

```
module CopyConsumer {  
    uses CopyServiceAPI;  
}
```

- A . Option A
- B . Option B
- C . Option C
- D . Option D

Answer: C

Question: 30

Which method throws an exception for not-a-number and infinite input values?

A)

```
static float validate1(String s) throws IllegalArgumentException {  
    return Float.parseFloat(s);  
}
```

B)

```
static float validate3(String s, float min, float max) throws IllegalArgumentException {
    float f = Float.parseFloat(s);
    if (!Float.isFinite(f) || f < min || f > max) {
        throw new IllegalArgumentException();
    }
    return f;
}
```

C)

```
static float validate2(String s, float min, float max) throws IllegalArgumentException {
    float f = Float.parseFloat(s);
    if (f < min || f > max) {
        throw new IllegalArgumentException();
    }
    return f;
}
```

D)

```
static float validate4(String s, float min, float max) throws IllegalArgumentException {
    float f = Float.parseFloat(s);
    if (Float.isFinite(f) && f < min && f > max) {
        throw new IllegalArgumentException();
    }
    return f;
}
```

- A . Option A
- B . Option B
- C . Option C
- D . Option D

Answer: A

Question: 31

Why does this compilation fail?

- A . The method Y. print (object) does not call the method super.print (object)
- B . The method x. print (object) is not accessible to Y.
- C . In method x. print (Collection), system. Out :: prints is an invalid Java identifier.
- D . The method print (object) and the method print (object...) are duplicates of each other.
- E . The method Y. print (object...) cannot override the final method x.print (object....).

Answer: D

Question: 32

Given:

```
public class Main {  
    private String[] strings = {"ABCDEFGHIJKLMNPQRSTUVWXYZ",  
        "abcdefghijklmnopqrstuvwxyz", "0123456789"};  
    public void write(String filename){  
        // line 1  
        for (String str: strings) {  
            ByteBuffer buffer = ByteBuffer.wrap(str.getBytes());  
            fileChannel.write(buffer);  
        }  
    } catch(IOException e){  
        e.printStackTrace();  
    }  
}  
public static void main(String[] args) {  
    Main test = new Main();  
    test.write("file_to_path");  
}
```

You want to obtain the Filechannel object on line 1.

Which code fragment will accomplish this?

A)

```
try (FileChannel fileChannel = Channels.newChannel(new FileOutputStream(filename));) {
```

B)

```
try(FileChannel fileChannel = new FileOutputStream(filename).getChannel();) {
```

C)

```
try (FileChannel fileChannel = new FileOutputStream(new FileChannel(filename));) {
```

D)

```
try(FileChannel fileChannel = new FileChannel(new FileOutputStream(filename));) {
```

A . Option A

B . Option B

C . Option C

D . Option D

Answer: A

Question: 33

What change will cause the code to compile successfully?

- A . Insert PriceChecker (?) prod on line 1.
- B . Insert PriceChecker <> prod on line 1.
- C . Insert PriceChecker <Electronics> prod on line 1.
- D . Insert PriceChecker <Plushy extends Products> prod on line 1.

Answer: A

Question: 34

Given the code fragment:

```
8. public class Test {  
9.     private final int x = 1;  
10.    static final int y;  
11.    public Test() {  
12.        System.out.print(x);  
13.        System.out.print(y);  
14.    }  
15.    public static void main(String args[]) {  
16.        new Test();  
17.    }  
18. }
```

What is the result?

- A . 1
- B . The compilation fails at line
- C . 10
- D . The compilation fails at line 16.
- E . The compilation fails at line 13.

Answer: C

Question: 35

A)

```
Map<String, List<Employee>> e3 =  
    roster.stream()  
        .collect(Collectors.groupingBy(  
            e -> Optional.ofNullable(e.getNeighborhood())  
                .get()  
        ));
```

B)

```
Map<String, List<Employee>> e3 =  
    roster.stream()  
        .collect(Collectors.groupingBy(  
            e -> Optional.ofNullable(e.getNeighborhood())  
            .get()  
        ));
```

C)

```
Map<String, List<Employee>> e1 =  
    roster.stream()  
        .collect(Collectors.groupingBy(  
            e -> Optional.ofNullable(e.getNeighborhood())  
        ));
```

D)

```
Map<Object, List<Employee>> e2 =  
    roster.stream()  
        .collect(Collectors.groupingBy(  
            e -> Optional.ofNullable(e.getNeighborhood())  
        ));
```

- A . Option A
- B . Option B
- C . Option C
- D . Option D

Answer: D

Question: 36

Given:

```
public final class X {  
    private String name;  
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
    public String toString() { return getName(); }  
}
```

and

```
public class Y extends X{  
    public Y(String name) {  
        super();  
        setName(name);  
    }  
    public static void main (String... args) {  
        Y y = new Y("HH");  
        System.out.println(y);  
    }  
}
```

What is the result?

- A . The compilation fails.
- B . Y@<< hashCode >>
- C . Null
- D . HH

Answer: B

Question: 37

Given the code fragment:

```
public class Main {  
    public static void main(String... args) {  
        List<String> list1 = new ArrayList<>(  
            List.of("Plane", "Automobile", "Motorcycle"));  
        List<String> list2 = new ArrayList<>(List.copyOf(list1));  
  
        list1.sort((String item1, String item2) -> item1.compareTo(item2));  
        list2.sort((String item1, String item2) -> item1.compareTo(item2));  
        System.out.println(list1.equals(list2));  
    }  
}
```

What is the result?

- A . A java.lang.UnsupportedOperationException is thrown.
- B . True
- C . False
- D . A java.lang.NullPointerException is thrown.

Answer: A

Question: 38

Given the content:

```
MessagesBundle.properties file:  
username = Username  
password = Password
```

and

```
MessagesBundle_fr_FR.properties file:  
username = Utilisateur  
password = Le passe
```

and

```
MessagesBundle_ru.properties file:  
username = Пользователь  
password = Пароль
```

and the code fragment:

```
public class Test {  
    public static void main(String[] args) {  
        Locale.setDefault(Locale.FRANCE);  
        ResourceBundle msg = ResourceBundle.getBundle("MessageBundle", new Locale("ru"));  
        System.out.println("User " + msg.getString("username"));  
        System.out.println("Pass " + msg.getString("password"));  
    }  
}
```

What Is the result?

A)

```
User = Пользователь  
Pass = Пароль
```

B)

| The compilation fails.

C)

A MissingResourceException is thrown at runtime.

D)

```
User = Utilisateur  
Pass = Le passe
```

E)

User Username

- A . Option A
- B . Option B
- C . Option C
- D . Option D
- E . Option

Answer: D, E

Question: 39

Given TripleThis.java:

```
6. import java.util.function.*;
7. public class TripleThis {
8.     public static void main(String[] args) {
9.         Function tripler = x -> { return (Integer) x * 3; };
10.        TripleThis.printValue(tripler, 4);
11.    }
12.    public static <T> void printValue(Function f, T num) {
13.        System.out.println(f.apply(num));
14.    }
15. }
```

Compiling TripleThis.java gives this compiler warning:

Note: TripleThis.java uses unchecked or unsafe operations.

Which two replacements done together remove this compiler warning?

- A . Replace line 9 with function<Integer> tripler = x-> - { return (Integer) X * 3 ; }.
- B . Replace line 12 with public static void printValue function<Integer> f, int num) {}.
- C . Replace line 12 with public static int printValue function<Integer, Integer>, f, T num {}.
- D . Replace line 12 with public static <T> void printValue (Function<T, T> f, T num) {}.
- E . Replace line 9 with function<Integer>, Integer> = X -> { return (integer) x * 3; }.

Answer: A, C

Question: 40

Given:

```
public class Employee {
    private String name;
    private String neighborhood;
    private LocalDate birthday;
    private int salary;
}

and

List<Employee> roster = new ArrayList<>(...);
Map<String, Optional<Employee>> m = roster.stream()
// Line 1
```

Which code fragment on line 1 makes the m map contain the employee with the highest salary for each neighborhood?

A)

```
.collect(Collectors.maxBy(Employee::getSalary,  
    Collectors.groupingBy(Comparator.comparing(e -> e.getNeighborhood()))));
```

B)

```
.collect(Collectors.groupingBy(Employee::getNeighborhood,  
    Collectors.maxBy(Comparator.comparing(Employee::getSalary))));
```

C)

```
.collect(Collectors.groupingBy(e -> e.getNeighborhood(),  
    Collectors.maxBy((x, y) -> y.getSalary() - x.getSalary())));
```

D)

```
.collect(Collectors.maxBy((x, y) -> y.getSalary() - x.getSalary(),  
    Collectors.groupingBy(Employee::getNeighborhood)));
```

- A . Option A
- B . Option B
- C . Option C
- D . Option D

Answer: D

Question: 41

Given:

```
public class Main {  
    public static void main(String[] args) {  
        List l = new ArrayList();  
        l.add("hello");  
        l.add("world");  
        print(l);  
    }  
    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

Answer: B

Question: 42

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.

Answer: A

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: 43

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));
- B. longList.parallelStream()
 .filter(w -> w.indexOf('e') != -1)
 .forEach(w -> shortList.add(w));
- C. shortList = longList.stream()
 .filter(w -> w.indexOf('e') != -1)
 .parallel()
 .collect(Collectors.toList());
- D. longList.stream()
 .filter(w -> w.indexOf('e') != -1)
 .parallel()
 .collect(shortlist);

- A . Option A
- B . Option B
- C . Option C
- D . Option D

Answer: C

Question: 44

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.

Answer: C

Question: 45

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"; }

Answer: C

Question: 46

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>();

Answer: A, C

Question: 47

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"

Answer: B

Question: 48

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**

Answer: B

Question: 49

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

- A . ToDoubleFunction**
- B . Supplier
- C . BiFunction
- D . LongConsumer

Answer: A

Question: 50

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.

Answer: C

Question: 51

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.

Answer: A, C

Question: 52

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();

Answer: A, C, F

Question: 53

Consider this method declaration:

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

- A) "SET SESSION AUTHORIZATION " + user
- B) "SET SESSION AUTHORIZATION " + stmt.getIdentifier(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 en quoting 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.

Answer: A

Question: 54

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.

Answer: A, B

Explanation:

/validating-input-using-java-util-scanner

Question: 55

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()));
```
- B.

```
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()));
```
- C.

```
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()));
```
- D.

```
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 . Option A
- B . Option B
- C . Option C
- D . Option D

Answer: C

Question: 56

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.

Answer: A, C

Question: 57

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);

Answer: B, D

Question: 58

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

Answer: D

Question: 59

Given:

```
public static void main(String[] args) {  
    try (Reader reader1 = new FileReader("File1.txt");  
         Reader reader2 = new FileReader("File2.txt");  
         Reader reader3 = new FileReader("File3_txt")) {  
  
    } catch (IOException ex) {  
        Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);  
    }  
    // Line 1  
    System.out.println("Done");  
}
```

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

- A . All three readers are still open.
- B . All three readers have been closed.
- C . The compilation fails.
- D . Only reader1 has been closed.

Answer: C

Question: 60

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.

Answer: A, D, F

Question: 61

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

Answer: C

Question: 62

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();

Answer: D

Question: 63

Given:

```
public class Main {  
    class Student {  
        String classname; // line 1  
        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 {.

Answer: B, D

Question: 64

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

Answer: A

Question: 65

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.

Answer: D

Question: 66

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

Answer: C

Question: 67

Given:

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

Which two are correct? (Choose two.)

- A . The output will be exactly 2 1 3 4 5.
- B . The program prints 1 4 2 3, but the order is unpredictable.
- C . Replacing forEach() with forEachOrdered(), the program prints 2 1 3 4 5, but the order is unpredictable.
- D . Replacing forEach() with forEachOrdered(), the program prints 1 2 3 4 5.
- E . Replacing forEach() with forEachOrdered(), the program prints 2 1 3 4 5.

Answer: B, D

Question: 68

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;.

Answer: E, F, G

Question: 69

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]]

Answer: B

Question: 70

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

Answer: B

Question: 71

Given:

```

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

```

When is the `readObject` method called?

- A . before this object is deserialized
- B . after this object is deserialized**
- C . before this object is serialized

- D . The method is never called.
E . after this object is serialized

Answer: B

Question: 72

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

Answer: D

Question: 73

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

Answer: D

Question: 74

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;

Answer: D

Question: 75

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

Answer: D

Question: 76

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;  
}
```

B)

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

C)

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

D)

```
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 . Option A
- B . Option B
- C . Option C
- D . Option D

Answer: C

Question: 77

Given:

```
List<Reader> dataFiles = new ArrayList<>();  
File indexFile = new File("MyIndex.idx");  
try (BufferedReader indexReader =  
        new BufferedReader(new FileReader(indexFile))) {  
    for(String file = indexReader.readLine(); 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.

Answer: B

Question: 78

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.

Answer: D

Question: 79

Given:

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

What is the output?

- A . :APPLE:ORANGE:BANANA appleorangebanana
- B . :APPLE:ORANGE:BANANA
- C . APPLE:apple ORANGE:orange BANANA:banana
- D . appleorangebanana :APPLE:ORANGE:BANANA
- E . apple:APPLE orange:ORANGE banana:BANANA

Answer: E

Question: 80

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")};

Answer: C

Question: 81

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.

Answer: B

Question: 82

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.

Answer: C

Question: 83

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.

Answer: C

Question: 84

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.println(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.

Question: 85

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.compareTo(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.compareTo(p2.name); } on line 2.
- D . Insert Comparator<Person> on line 1. Insert public int compare(Person person) { return person.name.compareTo(this.name); } on line 2.

Question: 86

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.

Answer: A

Question: 87

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

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

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

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

```

- A . Option A
- B . Option B
- C . Option C
- D . Option D

Answer: C, D

Question: 88

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.

Answer: D

Question: 89

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

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

Answer: D

Question: 90

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);

Question: 91

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))));`

- A . Option A
- B . Option B**
- C . Option C
- D . Option D

Question: 92

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.

Answer: D

Question: 93

Given:

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

Answer: C

Question: 94

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();

Answer: C

Question: 95

```
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));

Answer: B, D

Question: 96

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.

Answer: A

Question: 97

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

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

Which two are true? (Choose two.)

- A . A module providing an implementation of country.CountryDetails can be compiled and added without recompiling the main module.
- B . A module providing an implementation of country.CountryDetails must have a requires main; directive in its module-info.java file.
- C . An implementation of country.countryDetails can be added to the main module.
- D . To compile without an error, the application must have at least one module in the module source path that provides an implementation of country.CountryDetails.
- E . To run without an error, the application must have at least one module in the module path that provides an implementation of country.CountryDetails.

Answer: B, D

Explanation:

Question: 98

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.

Answer: D

Question: 99

Which code fragment prints 100 random numbers?

- A. var r= new Random();
new DoubleStream(r::nextDouble).limit(100).forEach(System.out::print);
- B. DoubleStream.generate(Random::nextDouble)
.limit (100).forFach(System.out::print);
- C. Doublestream.generate(Random.nextDouble).limit(100).forEach(System.out.print);
- D. var r = new Random(); DoubleStream.generate(r::nextDouble).limit(100).forEach(System.out::print);

- A . Option A
- B . Option B
- C . Option C
- D . Option D

Answer: D

Question: 100

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 ();

Answer: B

Question: 101

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;

Answer: A, B

Question: 102

Which code fragment compiles?

```

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

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

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

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

```

- A . Option A
 B . Option B
 C . Option C
D . Option D

Answer: D

Question: 103

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.

Answer: D

Question: 104

Given:

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

What is required to make the Foo class thread safe?

- A . No change is required.
- B . Make the declaration of lock static.
- C . Replace the lock constructor call with new ReentrantLock (true).
- D . Move the declaration of lock inside the foo method.

Answer: C

Explanation:

/how-to-make-java-class-thread-safe

Question: 105

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"));

Answer: B

Question: 106

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;

Answer: A

Question: 107

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();`.

Answer: A, D

Question: 108

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`

Answer: C

Question: 109

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.

Answer: C

Question: 110

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.

Answer: C

Question: 111

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.

Answer: A, C

Question: 112

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();

Answer: B

Question: 113

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.

Answer: A

Question: 114

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?

- A. private default void probeProcedure(){
 System.out.println("Launch Probe");
 System.out.println("Land on Asteroid");
}
- B. static void probeProcedure(){
 System.out.println("Launch Probe");
 System.out.println("Land on Asteroid");
}
- C. private void probeProcedure(){
 System.out.println("Launch Probe");
 System.out.println("Land on Asteroid");
}
- D. default void probeProcedure(){
 System.out.println("Launch Probe");
 System.out.println("Land on Asteroid");
}

A . Option A

B . Option B

C . Option C

D . Option D

Answer: B

Question: 115

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.

Answer: D

Question: 116

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(); }

Answer: C

Question: 117

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

Answer: C, D

Question: 118

Given the declaration:

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

Examine this code fragment:

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

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

- A . @Resource(priority=100)
- B . @Resource(priority=0)
- C . @Resource(name="Customer1", priority=100)
- D . @Resource(name="Customer1")
- E . @Resource

Answer: A, B

Question: 119

Which two are functional interfaces? (Choose two.)

- A. @FunctionalInterface
interface MyRunnable {
 public void run();
}
- B. @FunctionalInterface
interface MyRunnable {
 public void run();
 public void call();
}
- C. interface MyRunnable {
 public default void run() {}
 public void run(String s);
}
- D. @FunctionalInterface
interface MyRunnable {
}
- E. interface MyRunnable {
 @FunctionalInterface
 public void run();
}

- A . Option A
- B . Option B
- C . Option C

D . Option D

E . Option E

Answer: C, E

Question: 120

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 FileAlreadyExistsException.

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.

Answer: C

Question: 121

What makes Java dynamic?

A . At runtime, classes are loaded as needed, and new code modules can be loaded on demand.

- B . The runtime can process machine language sources as well as executables from different language compilers.
- C . The Java compiler uses reflection to test if class methods are supported by resources of a target platform.
- D . The Java compiler preprocesses classes to run on specific target platforms.

Answer: A

Question: 122

Given:

```
for(var i = 0; i < 10; i++) {  
    switch(i%5) {  
        case 2:  
            i *= i;  
            break;  
        case 3:  
            i++;  
            break;  
        case 1:  
        case 4:  
            i++;  
            continue;  
        default:  
            break;  
    }  
    System.out.print(i + " ");  
    i++;  
}
```

What is the result?

- A . nothing
- B . 0
- C . 10
- D . 0 4 9

Answer: A

Question: 123

Given:

```
1. public class Main {  
2.     public static void greet(String... args) {  
3.         System.out.print("Hello ");  
4.         for (String arg : args) {  
5.             System.out.println(arg);  
6.         }  
7.     }  
8.     public static void main(String[] args) {  
9.         Main c = null;  
10.        c.greet();  
11.    }  
12. }
```

What is the result?

- A . NullPointerException is thrown at line 4.
- B . NullPointerException is thrown at line 10.
- C . A compilation error occurs.

D . Hello

Answer: D

Question: 124

Given:

```
class Super {  
    static String greeting() { return "Good Night"; }  
    String name() { return "Harry"; }  
}
```

and

```
class Sub extends Super {  
    static String greeting() { return "Good Morning"; }  
    String name() { return "Potter"; }  
}
```

and

```
class Test {  
    public static void main(String[] args) {  
        Super s = new Sub();  
        System.out.println(s.greeting() + ", " + s.name());  
    }  
}
```

What is the result?

- A . Good Morning, Potter
- B . Good Night, Potter
- C . Good Morning, Harry
- D . Good Night, Harry

Answer: B

Question: 125

Given:

```
public class Person {  
    private String name;  
    public Person(String name) {  
        this.name = name;  
    }  
    public String toString() {  
        return name;  
    }  
}
```

and

```
public class Tester {  
    public static void main(String[] args) {  
        Person p = null;  
        checkPerson(p);  
        System.out.println(p);  
        p = new Person("Mary");  
        checkPerson(p);  
        System.out.println(p);  
    }  
    public static Person checkPerson(Person p) {  
        if (p == null) {  
            p = new Person("Joe");  
        } else {  
            p = null;  
        }  
        return p;  
    }  
}
```

What is the result?

- A . JoeMarry
- B . Joenull
- C . nullnull
- D . nullMary

Answer: D

Question: 126

Examine this excerpt from the declaration of the java.se module:

```
module java.se {  
    ...  
    requires transitive java.sql;  
    ...  
}
```

What does the transitive modifier mean?

- A . Only a module that requires the java.se module is permitted to require the java.sql module.
- B . Any module that requires the java.se module does not need to require the java.sql module.
- C . Any module that attempts to require the java.se module actually requires the java.sql module instead.
- D . Any module that requires the java.sql module does not need to require the java.se module.

Answer: A

Question: 127

Given:

/code/a/Test.java

containing:

```
package a;  
import b.Best;  
public class Test {  
    public static void main(String[] args) {  
        Best b = new Best();  
    }  
}
```

and

/code/b/Best.java

containing:

package b;

public class Best { }

Which is the valid way to generate bytecode for all classes?

- A . java /code/a/Test.java
- B . javac --d /code /code/a/Test

- C . java /code/a/Test.java /code/b/Best.java
- D . java --cp /code a.Test
- E . javac --d /code /code/a/Test.java /code/b/Best.java
- F . javac --d /code /code/a/Test.java

Answer: E

Question: 128

Given:

Automobile.java

```
public abstract class Automobile { //line 1
    abstract void wheels();
}
```

Car.java

```
public class Car extends Automobile {
    void wheels(int i) {           // line 2
        System.out.print(4);
    }
    public static void main(String[] args) {
        Automobile ob = new Car(); // line 4
        ob.wheels();
    }
}
```

What must you do so that the code prints 4?

- A . Remove the parameter from wheels method in line 3.
- B . Add @Override annotation in line 2.
- C . Replace the code in line 2 with Car ob = new Car();
- D . Remove abstract keyword in line 1.

Answer: B

Question: 129

Given:

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

and

```
public class Bar extends Foo {  
    public void foo(Collection arg) {  
        System.out.println("Hello world!");  
    }  
    public void foo(List arg) {  
        System.out.println("Hola Mundo!");  
    }  
}
```

and

```
Foo f1 = new Foo();  
Foo f2 = new Bar();  
Bar b1 = new Bar();  
List<String> li = new ArrayList<>();
```

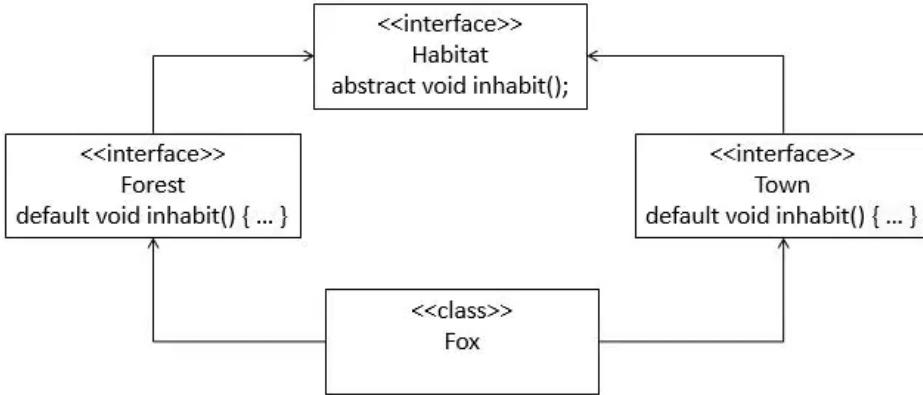
Which three are correct? (Choose three.)

- A . b1.foo(li) prints Hello world!
- B . f1.foo(li) prints Bonjour le monde!
- C . f1.foo(li) prints Hello world!
- D . f1.foo(li) prints Hola Mundo!
- E . b1.foo(li) prints Bonjour le monde!
- F . f2.foo(li) prints Hola Mundo!
- G . f2.foo(li) prints Bonjour le monde!
- H . b1.foo(li) prints Hola Mundo!
- I . f2.foo(li) prints Hello world!

Answer: A, B, H

Question: 130

Given:



Which statement is true about the Fox class?

- A . Fox class does not have to override inhabit method, so long as it does not try to call it.
- B . Fox class does not have to override the inhabit method if Forest and Town provide compatible implementations.
- C . Fox class must implement either Forest or Town interfaces, but not both.
- D . The inhabit method implementation from the first interface that Fox implements will take precedence.
- E . Fox class must provide implementation for the inhabit method.

Answer: B

Question: 131

Given:

```

public class Main {

    public static void checkConfiguration(String filename) {
        File file = new File(filename);
        if(!file.exists()) {
            throw new Error("Fatal Error: Configuration File, "
                + filename + ", is missing.");
        }
    }

    public static void main(String[] args) {
        checkConfiguration("App.config");
        System.out.println("Configuration is OK");
    }
}

```

If file 'App.config' is not found, what is the result?

- A . Configuration is OK
- B . The compilation fails.

C . Exception in thread 'main' java.lang.Error:Fatal Error: Configuration File, App.config, is missing.

D . nothing

Answer: B

Question: 132

Given:

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

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

- A . getTCount
- B . getACount
- C . getTotalCount
- D . getCCount
- E . getGCount

Answer: C, D

Question: 133

Given:

```
public class Person {  
    private String name = "Joe Bloggs";  
    public Person(String name) {  
        this.name = name;  
    }  
    public String toString() {  
        return name;  
    }  
}
```

and

```
public class Tester {  
    public static void main(String[] args) {  
        Person p1 = new Person(); // line 1  
        System.out.println(p1);  
    }  
}
```

What is the result?

- A . null
- B . Joe Bloggs
- C . The compilation fails due to an error in line 1.
- D . p1

Answer: C

Question: 134

Given:

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

and

```
public class Bar extends Foo {  
    public void foo(Collection arg) {  
        System.out.println("Hello world!");  
    }  
    public void foo(List arg) {  
        System.out.println("Olá Mundo!");  
    }  
}
```

and

```
Foo f1 = new Foo();  
Foo f2 = new Bar();  
Bar b1 = new Bar();  
Collection<String> c = new ArrayList<>();
```

Which three are true? (Choose three.)

- A . b1.foo(c) prints Bonjour le monde!
- B . f1.foo(c) prints Hello world!
- C . f1.foo(c) prints Ol Mundo!
- D . b1.foo(c) prints Hello world!
- E . f2.foo(c) prints Ol Mundo!
- F . b1.foo(c) prints Ol Mundo!
- G . f2.foo(c) prints Bonjour le monde!
- H . f2.foo(c) prints Hello world!
- I . f1.foo(c) prints Bonjour le monde!

Answer: B, F, G

Question: 135

Given the code fragment:

```
String s = "";
if (Double.parseDouble("11.00f") > 11) {
    s += 1;
}
if (1_7 == Integer.valueOf("17")) {
    s += 2;
}
if (1024 > 1023L) {
    s += 3;
}
System.out.print(s);
```

What is the result?

- A . 23
- B . 12
- C . 123
- D . 13

Answer: A

Question: 136

Given:

```

import java.io.FileNotFoundException;
import java.io.IOException;

public class Tester {
    public static void main(String[] args) {
        try {
            doA();
        } //line 1
    }
    private static void doA() throws IOException, IndexOutOfBoundsException {
        if (false) {
            throw new FileNotFoundException();
        } else {
            throw new IndexOutOfBoundsException();
        }
    }
}

```

What must be added in line 1 to compile this class?

- A . catch(IOException e) { }
- B . catch(FileNotFoundException | IndexOutOfBoundsException e) { }
- C . catch(FileNotFoundException | IOException e) { }
- D . catch(IndexOutOfBoundsException e) { }catch(FileNotFoundException e) { }
- E . catch(FileNotFoundException e) { }catch(IndexOutOfBoundsException e) { }

Answer: A

Question: 137

Which three initialization statements are correct? (Choose three.)

- A . int x = 12_34;
- B . short sh = (short)'A';
- C . String contact# = "(+2) (999) (232)";
- D . boolean true = (4 == 4);
- E . float x = 1.99;
- F . int[][] e = {{1,1},{2,2}};
- G . byte b = 10;char c = b;

Answer: A, B, F

Question: 138

Given:

```
public class Main {  
    public static void main(String[] args) {  
        int i = 1;  
        for(String s : args) {  
            System.out.println((i++) + " " + s);  
        }  
    }  
}
```

executed with this command:

```
java Main one two three
```

What is the output of this class?

- A . The compilation fails.
- B . 1) one2) two3) three
- C . A java.lang.ArrayIndexOutOfBoundsException is thrown.
- D . 1) one
- E . nothing

Answer: B

Question: 139

Given:

```
public class Tester {  
    public static void main(String[] args) {  
        byte x = 7, y = 6;  
        // line 1  
        System.out.println(z);  
    }  
}
```

Which expression when added at line 1 will produce the output of 1.17?

- A . float z = (float)(Math.round((float)x/y*100)/100);
- B . float z = Math.round((int)(x/y),2);
- C . float z = Math.round((float)x/y,2);
- D . float z = Math.round((float)x/y*100)/(float)100;

Answer: D

Question: 140

Given:

```
public class Tester {  
    public static void main(String[] args) {  
        int x = 4;  
        int y = 2;  
        System.out.println(x+y+"=(x+y)="+x+y);  
    }  
}
```

What is the result?

- A . An exception is thrown at runtime.
- B . $42=(x+y)=42$
- C . $42=(x+y)=6$
- D . $6=(x+y)=42$**
- E . $6=(x+y)=6$

Answer: D

Question: 141

Given:

```
package a;  
public abstract class Animal {  
    protected abstract void walk();  
}  
package b;  
public abstract class Human extends Animal {  
    // line 1  
}
```

Which two lines inserted in line 1 will allow this code to compile? (Choose two.)

- A . `protected void walk(){};`**
- B . `void walk(){};`
- C . `abstract void walk();`

- D . private void walk(){}
E . public abstract void walk();

Answer: A, E

Question: 142

Given:

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

and

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

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

- A . map.forEach((k, v) -> set(v));
B . set(map.values());
C . super.set(List<String> map)
D . super.set(map.values());
E . set(map)

Answer: B, D

Question: 143

Given:

```

public class Test {
    private int sum;
    public int compute() {
        int x = 0;
        while(x < 3) {
            sum += x++;
        }
        return sum;
    }
    public static void main(String[] args) {
        Test t = new Test();
        int sum = t.compute();
        sum = t.compute();
        t.compute();
        System.out.println(sum);
    }
}

```

What is the result?

- A . 9
- B . An exception is thrown at runtime.
- C . 3
- D . 6

Answer: D

Question: 144

Given:

```

public class Person {
    private String name;
    public void setName(String name) {
        String title = "Dr. ";
        name = title+name;
    }
    public String toString() {
        return name;
    }
}

```

and

```

public class Test {
    public static void main(String args[]) {
        Person p = new Person();
        p.setName("Who");
        System.out.println(p);
    }
}

```

What is the result?

- A . Dr. Who
- B . Dr. Null
- C . An exception is thrown at runtime.
- D . null

Answer: D

Question: 145

Given:

```
import java.util.*;

public class Main {
    static Map<String, String> map = new HashMap<>();
    static List<String> keys =
        new ArrayList<>(List.of("A", "B", "C", "D"));
    static String[] values =
        {"one", "two", "three", "four" };

    static {
        for(var i = 0; i < keys.size(); i++) {
            map.put(keys.get(i), values[i]);
        }
    }

    public static void main(String[] args) {
        keys.clear();
        values = new String[0];
        System.out.println("Map: " + map.size() +
            " Keys: " + keys.size() +
            " Values: " + values.length);
    }
}
```

What is the result?

- A . Map: 0 Keys: 0 Values: 0
- B . The compilation fails.
- C . Map: 4 Keys: 4 Values: 4
- D . Map: 4 Keys: 0 Values: 0
- E . Map: 0 Keys: 4 Values: 4

Answer: D

Question: 146

Which command line runs the main class com.acme.Main from the module com.example?

- A . java --module-path mods com.example/com.acme.Main
- B . java --classpath com.example.jar com.acme.Main
- C . java --module-path mods -m com.example/com.acme.Main
- D . java -classpath com.example.jar --m com.example/com.acme.Main

Answer: D

Question: 147

Given the code fragment:

```
public static void main(String[] args) {  
    List<Integer> even = List.of();  
    even.add(0, -1);  
    even.add(0, -2);  
    even.add(0, -3);  
    System.out.println(even);  
}
```

What is the output?

- A . The compilation fails.
- B . [-1, -2, -3]
- C . [-3, -2, -1]
- D . A runtime exception is thrown.

Answer: D

Question: 148

Which statement about access modifiers is correct?

- A . An instance variable can be declared with the static modifier.
- B . A local variable can be declared with the final modifier.
- C . An abstract method can be declared with the private modifier.
- D . An inner class cannot be declared with the public modifier.

E . An interface can be declared with the protected modifier.

Answer: B

Question: 149

Given:

```
List<String> list = ... ;  
list.forEach( x -> { System.out.println(x); } );
```

What is the type of x?

- A . char
- B . List<Character>
- C . String
- D . List<String>

Answer: C

Question: 150

Given:

```
package A;  
class Test {  
    String name;  
    public Test(String name) {  
        this.name = name;  
    }  
    public String toString() {  
        return name;  
    }  
}
```

and

```
package B;  
import A.Test;  
public class Main {  
    public static void main(String[] args) {  
        Test test = new Test("Student");  
        System.out.println(test);  
    }  
}
```

What is the result?

- A . null
- B . nothing
- C . It fails to compile.
- D . java.lang.IllegalAccessException is thrown.
- E . Student

Answer: C

Question: 151

Given:

```
package test;
import java.time.*;
public class Diary {
    private LocalDate now = LocalDate.now();
    public LocalDate getDate() {
        return now;
    }
}
```

and

```
package test;
public class Tester {
    public static void main(String[] args) {
        Diary d = new Diary();
        System.out.println(d.getDate());
    }
}
```

Which statement is true?

- A . Class Tester does not need to import java.time.LocalDate because it is already visible to members of the package test.
- B . All classes from the package java.time. are loaded for the class Diary.
- C . Only LocalDate class from java.time package is loaded.
- D . Tester must import java.time.LocalDate in order to compile.

Answer: A

Question: 152

Given:

```
public interface A {  
    public Iterable a();  
}  
public interface B extends A {  
    public Collection a();  
}  
public interface C extends A {  
    public Path a();  
}  
public interface D extends B, C {  
}
```

Why does D cause a compilation error?

- A . D inherits a() only from C.
- B . D inherits a() from B and C but the return types are incompatible.**
- C . D extends more than one interface.
- D . D does not define any method.

Answer: B

Question: 153

Given:

```
public class Sportscar extends Automobile{  
    private float turbo;  
    ....  
    public void setTurbo (float turbo){  
        this.turbo = turbo;  
    }  
}
```

What is known about the Sportscar class?

- A . The Sportscar class is a subclass of Automobile and inherits its methods.**
- B . The Sportscar subclass cannot override setTurbo method from the superclass Automobile.
- C . The Sportscar class is a superclass that has more functionality than the Automobile class.
- D . The Sportscar class inherits the setTurbo method from the superclass Automobile.

Answer: A

Question: 154

Given:

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

and

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

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

- A . B cannot be abstract.
- B . B is a subtype of A.
- C . A cannot be abstract.
- D . A cannot be final.
- E . B cannot be final.
- F . A is a subtype of B.

Answer: A, B, D

Question: 155

Given:

```
public class Main {  
    public static void main(String[] args) {  
        for(int i = 0; i < args.length; i++) {  
            System.out.println(i + " . " + args[i]);  
            switch(args[i]) {  
                case "one":  
                    continue;  
                case "two":  
                    i--;  
                    continue;  
                default:  
                    break;  
            }  
        }  
    }  
}
```

executed with this command:

java Main one two three

What is the result?

- A . 0). one
- B . 0). one1). two2). three
- C . The compilation fails.
- D . It creates an infinite loop printing:0). one1). two1). two...
- E . A java.lang.NullPointerException is thrown.

Answer: D

Question: 156

Given:

```
public interface ExampleInterface{ }
```

Which two statements are valid to be written in this interface? (Choose two.)

- A . public abstract void methodB();
- B . final void methodG(){System.out.println("G");}
- C . private abstract void methodC();
- D . public String methodD();
- E . public int x;
- F . final void methodE();
- G . public void methodF(){System.out.println("F");}

Answer: A, D

Question: 157

Given this requirement:

Module vehicle depends on module part and makes its com.vehicle package available for all other modules.

Which module-info.java declaration meets the requirement?

A

```
module vehicle{
    requires part;
    exports com.vehicle;
}
```

B

```
module vehicle {
    requires part;
    uses com.vehicle;
}
```

C

```
module vehicle{
    requires part;
    exports com.vehicle to part;
}
```

D

```
module vehicle {
    requires com.vehicle;
    exports part;
}
```

A . Option A

B . Option B

C . Option C

D . Option D

Answer: A

Question: 158

Which is the correct order of possible statements in the structure of a Java class file?

A . class, package, import

B . package, import, class

C . import, package, class

D . package, class, import

E . import, class, package

Answer: B

Question: 159

Given the formula to calculate a monthly mortgage payment:

$$M = P \frac{r(1+r)^n}{(1+r)^n - 1}$$

and these declarations:

```
double m;           //monthly payment
double r = 0.05/12; //monthly interest rate
int p = 100_000;    //principal
int n = 180;        //number of payments
```

How can you code the formula?

- A . `m = p * (r * Math.pow(1 + r, n) / (Math.pow(1 + r, n) - 1));`
- B . `m = p * ((r * Math.pow(1 + r, n) / (Math.pow(1 + r, n)) - 1));`
- C . `m = p * r * Math.pow(1 + r, n) / Math.pow(1 + r, n) - 1;`
- D . `m = p * (r * Math.pow(1 + r, n) / Math.pow(1 + r, n) - 1);`

Answer: A

Question: 160

Given:

```
public class MethodTest {
    // line 1
}
```

Which two method implementations are correct, when inserted independently in line 1? (Choose two.)

```
A.  
public boolean methodD(int x) {  
    return x > 0;  
}  
  
B.  
public String methodB() {  
    System.out.println("methodB");  
}  
  
C.  
public char methodE (String msg) {  
    return msg;  
}  
  
D.  
public void methodC(int x) {  
    return ++x;  
}  
  
E.  
public void methodA() {  
    System.out.println("methodA");  
}
```

- A . Option A
- B . Option B
- C . Option C
- D . Option D
- E . Option E

Answer: A, E

Question: 161

Which describes an aspect of Java that contributes to high performance?

- A . Java prioritizes garbage collection.
- B . Java has a library of built-in functions that can be used to enable pipeline burst execution.
- C . Java monitors and optimizes code that is frequently executed.
- D . Java automatically parallelizes code execution.

Answer: C

Question: 162

Given:

```
public method foo() throws FooException {  
    ...  
}
```

and omitting the throws FooException clause results in a compilation error.

Which statement is true about FooException?

- A . FooException is a subclass of RuntimeException.
- B . FooException is unchecked.
- C . The body of foo can only throw FooException.
- D . The body of foo can throw FooException or one of its subclasses.

Answer: D

Question: 163

Given:

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

What is the output?

- A . Hello world!Bonjour le monde!
- B . Hello world!Hello world!

C . Bonjour le monde!Hello world!

D . Bonjour le monde!Bonjour le monde!

Answer: C

Question: 164

Given:

```
import java.util.*;
public class Foo {
    public List<Number> foo(Set<CharSequence> m) { ... }
}
```

and

```
import java.util.*;
public class Bar extends Foo {
    //line 1
}
```

Which two statements can be added at line 1 in Bar to successfully compile it? (Choose two.)

- A . public List<Integer> foo(Set<CharSequence> m) { ... }
- B . public ArrayList<Number> foo(Set<CharSequence> m) { ... }
- C . public List<Integer> foo(TreeSet<String> m) { ... }
- D . public List<Integer> foo(Set<String> m) { ... }
- E . public List<Object> foo(Set<CharSequence> m) { ... }
- F . public ArrayList<Integer> foo(Set<String> m) { ... }

Answer: B, C

Question: 165

Analyze the code:

```

public class Test {
    static String prefix = "Global:";
    private String name = "namescope";
    public static String getName() {
        return new Test().name;
    }
    public static void main(String[] args) {
        Test t = new Test();
        System.out.println(/* Insert code here */);
    }
}

```

Which two options can you insert inside println method to produce Global:namescope? (Choose two.)

- A . Test.prefix+Test.name
- B . new Test().prefix+new Test().name
- C . Test.prefix+Test.getName()
- D . Test.getName+prefix
- E . prefix+Test.name
- F . prefix+name

Answer: B, C

Question: 166

Given:

```

public class Foo {
    public static void main(String... args) {
        for (var x : args) {
            System.out.println(x);
        }
    }
}

```

What is the type of the local variable x?

- A . Character
- B . char
- C . String[]
- D . String

Answer: D

Question: 167

Given the code fragment:

```
int x = 0;
do {
    x++;
    if (x == 1) {
        continue;
    }
    System.out.println(x);
} while(x < 1);
```

What is the result?

- A . 01
- B . 0
- C . 1
- D . The program prints nothing.
- E . It prints 1 in the infinite loop.

Answer: D

Question: 168

Given:

```
public class Tester {  
    public static void main(String[] args) {  
        char letter = 'b';  
        int i = 0;  
        switch(letter) {  
            case 'a':  
                i++;  
                break;  
            case 'b':  
                i++;  
            case 'c' | 'd': // line 1  
                i++;  
            case 'e':  
                i++;  
                break;  
            case 'f':  
                i++;  
                break;  
            default:  
                System.out.print(letter);  
        }  
        System.out.println(i);  
    }  
}
```

What is the result?

- A . b1
- B . 2
- C . b2
- D . 1
- E . b3
- F . 3
- G . The compilation fails due to an error in line 1.

Answer: F

Question: 169

Given:

```

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

    void setACount(int cCount){
        cCount = cCount;
    }
    void setTCount(){
        this.tCount = tCount;
    }
    int setCCount(){
        return cCount;
    }
    int setGCount(int g){
        gCount = g;
        return gCount;
    }
    void setAllCounts(int x){
        aCount = tCount = this.cCount = setGCount(x);
    }
}

```

Which two methods modify field values? (Choose two.)

- A . setAllCounts
- B . setACount
- C . setGCount
- D . setCCount
- E . setTCount

Answer: A, C

Question: 170

Given:

```

public class Test {
    public static void main(String[] args) {
        int x;
        int y = 5;
        if (y > 2) {
            x = ++y;
            y = x + 7;
        } else {
            y++;
        }
        System.out.print(x + " " + y);
    }
}

```

What is the result?

- A . compilation error
- B . 0 5
- C . 6 13
- D . 5 12

Answer: A

Question: 171

Which two describe reasons to modularize the JDK? (Choose two.)

- A . easier to understand the Java language
- B . improves security and maintainability
- C . easier to expose implementation details
- D . improves application robustness
- E . easier to build a custom runtime linking application modules and JDK modules

Answer: B, D

Question: 172

Which two statements are correct about modules in Java? (Choose two.)

- A . java.base exports all of the Java platforms core packages.
- B . module-info.java can be placed in any folder inside module-path.
- C . A module must be declared in module-info.java file.
- D . module-info.java cannot be empty.
- E . By default, modules can access each other as long as they run in the same folder.

Answer: A, C

Question: 173

Given:

```
void myLambda() {  
    int i = 25;  
    Supplier<Integer> foo = () -> i;  
    i++;  
    System.out.println(foo.get());  
}
```

Which is true?

- A . The code compiles but does not print any result.
- B . The code prints 25.**
- C . The code does not compile.
- D . The code throws an exception at runtime.

Answer: C

Question: 174

Given:

```
public interface A {  
    abstract void x();  
}
```

and

```
public abstract class B /* position 1 */ {  
    /* position 2 */  
    public void x() { }  
    public abstract void z();  
}
```

and

```
public class C extends B implements A {  
/* position 3 */  
}
```

Which code, when inserted at one or more marked positions, would allow classes B and C to compile?

- A . @Override // position 3void x() {} // position 3@Override // position 3public void z() {} // position 3
- B . @Override // position 2public void z() {} // position 3**
- C . implements A // position 1@Override // position 2
- D . public void z() {} // position 3

Answer: B

Question: 175

Given:

```
import java.time.LocalDate;
import static java.time.DayOfWeek.*;
public class Main {
    public static void main(String[] args) {
        var today = LocalDate.now().with(TUESDAY).getDayOfWeek();
        switch(today) {
            case SUNDAY:
            case SATURDAY:
                System.out.println("Weekend");
                break;
            case MONDAY:  FRIDAY:
                System.out.println("Working");
            default:
                System.out.println("Unknown");
        }
    }
}
```

What is the result?

- A . Working
- B . Unknown**
- C . Tuesday
- D . The compilation fails.
- E . Tuesday
- F . Working

Answer: B

Question: 176

Given:

```

String[][] arr = {
    {"Red", "White"},
    {"Black"},
    {"Blue", "Yellow", "Green", "Violet"}
};
for(int row = 0; row < arr.length; row++) {
    int column = 0;
    for(; column < arr[row].length; column++) {
        System.out.println("[" + row + "," + column + "] = " + arr[row][column]);
    }
}

```

What is the result?

- A . [0,0] = Red[0,1] = White[1,0] = Black[1,1] = Blue[2,0] = Yellow[2,1] = Green[3,0] = Violet
- B . [0,0] = Red[1,0] = Black[2,0] = Blue
- C . java.lang.ArrayIndexOutOfBoundsException thrown
- D . [0,0] = Red[0,1] = White[1,0] = Black[2,0] = Blue[2,1] = Yellow[2,2] = Green[2,3] = Violet

Answer: D

Question: 177

Which two commands are used to identify class and module dependencies? (Choose two.)

- A . jmod describe
- B . java Hello.java
- C . jdeps --list-deps
- D . jar --show-module-resolution
- E . java --show-module-resolution

Answer: C, E

Question: 178

Given:

```

public class Tester {
    public static void main(String[] args) {
        String s = "this is it";
        int x = s.indexOf("is");
        s.substring(x+3);
        x = s.indexOf("is");
        System.out.println(s+" "+x);
    }
}

```

What is the result?

- A . is it 1
- B . An IndexOutOfBoundsException is thrown at runtime.
- C . is it 0
- D . this is it 2**
- E . this is it 3

Answer: D

Question: 179

Given:

```

class ConSuper {
    protected ConSuper(){
        this(2);
        System.out.print("1");
    }
    protected ConSuper(int a){
        System.out.print(a);
    }
}

```

and

```

public class ConSub extends ConSuper{
    ConSub(){
        this(4);
        System.out.print("3");
    }
    ConSub(int a) {
        System.out.print(a);
    }
    public static void main (String[] args){
        new ConSub(4);
    }
}

```

What is the result?

- A . 2134
- B . 2143
- C . 214
- D . 234

Answer: C

Question: 180

Given:

```
public class Price {  
    private final double value;  
    public Price(String value) {  
        this(Double.parseDouble(value));  
    }  
    public Price(double value) {  
        this.value = value;  
    }  
    public Price () {}  
    public double getValue() { return value; }  
    public static void main(String[] args) {  
        Price p1 = new Price("1.99");  
        Price p2 = new Price(2.99);  
        Price p3 = new Price();  
        System.out.println(p1.getValue()+" "+p2.getValue()+" "+p3.getValue());  
    }  
}
```

What is the result?

- A . The compilation fails.
- B . 1.99,2.99,0
- C . 1.99,2.99,0.0
- D . 1.99,2.99

Answer: A

Question: 181

Given:

```
public class Over {  
    public void analyze(Object[] o){  
        System.out.println("I am an object array");  
    }  
    public void analyze(long[] l){  
        System.out.println("I am an array");  
    }  
    public void analyze(Object o){  
        System.out.println("I am an object");  
    }  
    public static void main(String[] args) {  
        int[] nums = new int[10];  
        new Over().analyze(nums); // line 1  
    }  
}
```

What is the output?

- A . I am an object array
- B . The compilation fails due to an error in line 1.
- C . I am an array
- D . I am an object

Answer: D

Question: 182

Which two statements are correct about try blocks? (Choose two.)

- A . A try block can have more than one catch block.
- B . A finally block in a try-with-resources statement executes before the resources declared are closed.
- C . A finally block must be immediately placed after the try or catch blocks.
- D . A try block must have a catch block and a finally block.
- E . catch blocks must be ordered from generic to specific exception types.

Answer: A, C

Question: 183

Given:

```

public class Test {
    public static void main(String[] args) {
        AnotherClass ac = new AnotherClass();
        SomeClass sc = new AnotherClass();
        ac = sc;
        sc.methodA();
        ac.methodA();
    }
}
class SomeClass {
    public void methodA() {
        System.out.println("SomeClass#methodA()");
    }
}
class AnotherClass extends SomeClass {
    public void methodA() {
        System.out.println("AnotherClass#methodA()");
    }
}

```

What is the result?

- A . A ClassCastException is thrown at runtime.
- B . AnotherClass#methodA()AnotherClass#methodA()
- C . The compilation fails.
- D . SomeClass#methodA()AnotherClass#methodA()
- E . AnotherClass#methodA()SomeClass#methodA()
- F . SomeClass#methodA()SomeClass#methodA()

Answer: C

Question: 184

Given:

```

public interface InterfaceOne {
    void printOne();
}

```

Which three classes successfully override printOne()? (Choose three.)

```

A.
public abstract class TestClass implements InterfaceOne {
    public abstract void printOne();
}

B.
public class TestClass implements InterfaceOne {
    private void printOne(){
        System.out.println("one");
    }
}

C.
public class TestClass implements InterfaceOne {
    public void printOne(){
        System.out.println("one");
    }
}

D.
public abstract class TestClass implements InterfaceOne {
    public void printOne(){
        System.out.println("one");
    }
}

E.
public abstract class TestClass implements InterfaceOne {
    public String printOne(){
        return "one";
    }
}

F.
public class TestClass{
    public void printOne(){
        System.out.println("one");
    }
}

```

- A . Option A
- B . Option B
- C . Option C
- D . Option D
- E . Option E
- F . Option F

Answer: A, C, D

Question: 185

Given the code fragment:

```
int x = 0;
while(x < 10){
    System.out.print(x++);
}
```

Which "for" loop produces the same output?

A.

```
int b = 0;
for( ; b < 10; ){
    System.out.print(++b);
}
```

B.

```
for(a; a < 10; a++){
    System.out.print(a);
}
```

C.

```
for(int d = 0; d < 10; ){
    System.out.print(d);
    ++d;
}
```

D.

```
for(int c = 0; ; c++){
    System.out.print(c);
    if(c == 10){
        break;
    }
}
```

A . Option A

B . Option B

C . Option C

D . Option D

Answer: C

Question: 186

Given:

```
import java.util.ArrayList;
import java.util.Arrays;
public class NewMain {
    public static void main(String[] args) {
        String[] fruitNames = { "apple", "orange",
            "grape", "lemon", "apricot", "watermelon" };
        var fruits = new ArrayList<>(Arrays.asList(fruitNames));
        fruits.sort((var a, var b) -> -a.compareTo(b));
        fruits.forEach(System.out::println);
    }
}
```

What is the result?

- A . watermelonorangelemongrapeapricotapple
- B . nothing
- C . appleapricotgrapelemonorangerawatermelon
- D . appleorangegrapelemonapricotwatermelon

Answer: A

Question: 187

Given the code fragment:

```
String s1 = new String("ORACLE");
String s2 = "ORACLE";
String s3 = s1.intern();

System.out.print((s1==s2) + " ");
System.out.print((s2==s3) + " ");
System.out.println(s1==s3);
```

What is the result?

- A . false true true
- B . true false false
- C . false false true
- D . false true false

Answer: D

Question: 188

Given:

```
public class Test {  
    private String[] strings;  
}
```

Which two constructors will compile and set the class field strings? (Choose two.)

A.

```
public Test(List<String> strings) {  
    this.strings = strings;  
}
```

B.

```
public Test(String... strings) {  
    strings = strings;  
}
```

C.

```
public Test(String... strings) {  
    this.strings = strings;  
}
```

D.

```
public Test(String strings) {  
    strings = strings;  
}
```

E.

```
public Test(String[] strings) {  
    this.strings = strings;  
}
```

- A . Option A
- B . Option B
- C . Option C
- D . Option D
- E . Option E

Answer: C, E

Question: 189

Given:

```
public class Hello {  
    public static void main(String[] args) {  
        System.out.println(args[0]+args[1]+args[2]);  
    }  
}
```

executed using command:

```
java Hello "Hello World" Hello World
```

What is the output?

- A . An exception is thrown at runtime.
- B . Hello WorldHello World
- C . Hello World Hello World
- D . Hello WorldHelloWorld
- E . HelloHello WorldHelloWorld

Answer: C

Question: 190

Given the code fragment:

```
char[][] arrays = {{'a', 'd'}, {'b', 'e'}, {'c', 'f'}};  
for (char[] xx : arrays) {  
    for (char yy : xx) {  
        System.out.print(yy);  
    }  
    System.out.print(" ");  
}
```

What is the result?

- A . ab cd ef
- B . An ArrayIndexOutOfBoundsException is thrown at runtime.
- C . The compilation fails.
- D . abc def
- E . ad be cf

Answer: E

Question: 191

Given:

```
public class Foo {  
    public <T> Collection<T> foo(Collection<T> arg) { ... }  
}
```

and

```
public class Bar extends Foo { ... }
```

Which two statements are true if the method is added to Bar? (Choose two.)

- A . public Collection<String> foo(Collection<String> arg) { ... } overrides Foo.foo.
- B . public <T> Collection<T> foo(Stream<T> arg) { ... } overloads Foo.foo.
- C . public <T> List<T> foo(Collection<T> arg) { ... } overrides Foo.foo.
- D . public <T> Collection<T> foo(Collection<T> arg) { ... } overloads Foo.foo.
- E . public <T> Collection<T> bar(Collection<T> arg) { ... } overloads Foo.foo.
- F . public <T> Iterable<T> foo(Collection<T> arg) { ... } overrides Foo.foo.

Answer: C, F

Question: 192

Given:

```
package test.t1;
public class A {
    public int x = 42;
    protected A() {} // line 1
}
```

and

```
package test.t2;
import test.t1.*;
public class B extends A {
    int x = 17; // line 2
    public B() { super(); } // line 3
}
```

and

```
package test;
import test.t1.*;
import test.t2.*;
public class Tester {
    public static void main(String[] args) {
        A obj = new B(); // line 4
        System.out.println(obj.x); // line 5
    }
}
```

What is the result?

A . 42

B . The compilation fails due to an error in line 4.

C . 17

D . The compilation fails due to an error in line 3.

E . The compilation fails due to an error in line 2.

F . The compilation fails due to an error in line 1.

G . The compilation fails due to an error in line 5.

Answer: A

Question: 193

Which describes a characteristic of setting up the Java development environment?

A . Setting up the Java development environment requires that you also install the JRE.

B . The Java development environment is set up for all operating systems by default.

C . You set up the Java development environment for a specific operating system when you install the JDK.

D . Setting up the Java development environment occurs when you install an IDE before the JDK.

Answer: D

Question: 194

Given:

```
class Employee {  
    String office;  
}
```

and the code fragment:

```
5. public class HRApp {  
6.     var employee = new ArrayList<Employee>();  
7.     public var display() {  
8.         var employee = new Employee();  
9.         var offices = new ArrayList<>();  
10.        offices.add("Chicago");  
11.        offices.add("Bangalore");  
12.        for (var office : offices) {  
13.            System.out.print("Employee Location"+ office);  
14.        }  
15.    }  
16. }
```

Which two lines cause compilation errors? (Choose two.)

- A . line 12
- B . line 6**
- C . line 9
- D . line 8
- E . line 7**

Answer: B, E

Question: 195

Given:

```
StringBuilder s = new StringBuilder("ABCD");
```

Which would cause s to be AQCD?

- A . s.replace(s.indexOf("A"), s.indexOf("C"), "Q");
- B . s.replace(s.indexOf("B"), s.indexOf("C"), "Q");**
- C . s.replace(s.indexOf("B"), s.indexOf("B"), "Q");
- D . s.replace(s.indexOf("A"), s.indexOf("B"), "Q");

Answer: B

Question: 196

Given:

```
1. interface Pastry {  
2.     void getIngredients();  
3. }  
4. abstract class Cookie implements Pastry {}  
5.  
6. class ChocolateCookie implements Cookie {  
7.     public void getIngredients() {}  
8. }  
9. class CoconutChocolateCookie extends ChocolateCookie {  
10.    void getIngredients(int x) {}  
11. }
```

Which is true?

- A . The compilation fails due to an error in line 6.**
- B . The compilation succeeds.
- C . The compilation fails due to an error in line 4.
- D . The compilation fails due to an error in line 10.
- E . The compilation fails due to an error in line 7.
- F . The compilation fails due to an error in line 9.
- G . The compilation fails due to an error in line 2.

Answer: A

Question: 197

Given:

```
class Mycar {  
}  
  
and  
  
javac C:\workspace4\Mycar.java
```

What is the expected result of javac?

- A . javac fails to compile the class and prints the error message, C:\workspace4\Mycar.java:1:error: package java does not exist
- B . javac compiles Mycar.java without errors or warnings.**
- C . javac fails to compile the class and prints the error message, C:\workspace4\Mycar.java:1:error: expected import java.lang
- D . javac fails to compile the class and prints the error message, Error: Could not find or load main class Mycar.class

Answer: B

Question: 198

Given:

```
public class DNASynth {  
    int aCount;  
    int tCount;  
    int cCount;  
    int gCount;  
  
    DNASynth(int a, int tCount, int c, int g){  
        // line 1  
    }  
    int setCCount(int c){  
        return c;  
    }  
    void setGCount(int gCount){  
        this.gCount = gCount;  
    }  
}
```

Which two lines of code when inserted in line 1 correctly modifies instance variables? (Choose two.)

- A . setCCount(c) = cCount;
- B . tCount = tCount;
- C . setGCount(g);
- D . cCount = setCCount(c);**

E . aCount = a;

Answer: B, E

Question: 199

Which two statements are true about Java modules? (Choose two.)

- A . Modular jars loaded from --module-path are automatic modules.
- B . Any named module can directly access all classes in an automatic module.
- C . Classes found in --classpath are part of an unnamed module.
- D . Modular jars loaded from --classpath are automatic modules.
- E . If a package is defined in both the named module and the unnamed module, then the package in the unnamed module is ignored.

Answer: A, C

Question: 200

Given:

```
1. {
2.     Iterator iter = List.of(1,2,3).iterator();
3.     while (iter.hasNext()) {
4.         foo(iter.next());
5.     }
6.     Iterator iter2 = List.of(1,2,3).iterator();
7.     while (iter.hasNext()) {
8.         bar(iter2.next());
9.     }
10. }
11. for (Iterator iter = List.of(1,2,3).iterator(); iter.hasNext(); ) {
12.     foo(iter.next());
13. }
14. for (Iterator iter2 = List.of(1,2,3).iterator(); iter.hasNext(); ) {
15.     bar(iter2.next());
16. }
```

Which loop incurs a compile time error?

- A . the loop starting line 11
- B . the loop starting line 7
- C . the loop starting line 14

D . the loop starting line 3

Answer: C

Question: 201

Given:

```
package b;
public class Person {
    protected Person() {           //line 1
    }
}
```

and

```
package a;
import b.Person;
public class Main {             //line 2
    public static void main(String[] args) {
        Person person = new Person(); //line 3
    }
}
```

Which two allow a.Main to allocate a new Person? (Choose two.)

- A . In Line 1, change the access modifier to privateprivate Person() {
- B . In Line 1, change the access modifier to publicpublic Person() {
- C . In Line 2, add extends Person to the Main classpublic class Main extends Person {and change Line 3 to create a new Main objectPerson person = new Main();
- D . In Line 2, change the access modifier to protectedprotected class Main {
- E . In Line 1, remove the access modifierPerson() {

Answer: B, C

Question: 202

Given:

```
class Myclass {
public static void main(String [] args) {
    System.out.println(arg[1] + "--" + arg[3] + "--" + arg[0]);
}
}
```

executed using this command:

java Myclass My Car is red

What is the output of this class?

- A . Car--red--My
- B . My--Car--is
- C . My--is--java
- D . java--Myclass--My
- E . Myclass--Car--red

Answer: A

Question: 203

Given:

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

What is the result?

- A . It throws a runtime exception.
- B . Value of Euler = 2.71828
- C . The code does not compile.
- D . Value of Euler = "2.71828"

Answer: C

Question: 204

Given:

```

public class Tester {
    private int x;
    private static int y;
    public static void main(String[] args) {
        Tester t1 = new Tester();
        t1.x = 2;
        Tester.y = 3;
        Tester t2 = new Tester();
        t2.x = 4;
        t2.y = 5;
        System.out.println(t1.x+" "+t1.y);
        System.out.println(t2.x+" "+Tester.y);
        System.out.println(t2.x+" "+t1.y);
    }
}

```

What is the result?

- A . 2,34,34,5
- B . 2,34,54,5
- C . 2,54,54,5**
- D . 2,34,54,3

Answer: C

Question: 205

Given:

```

var i = 10;
var j = 5;
i += (j * 5 + j) / i - 2;
System.out.println(i);

```

What is the result?

- A . 5
- B . 3
- C . 23
- D . 25
- E . 11**

Answer: E

Question: 206

Given:

```
import java.util.function.BiFunction;
public class Pair<T> {
    final BiFunction<T, T, Boolean> validator;
    T left = null;
    T right = null;
    private Pair() {
        validator=null;
    }
    Pair(BiFunction<T, T, Boolean> v, T x, T y) {
        validator = v;
        set(x, y);
    }
    void set(T x, T y) {
        if (!validator.apply(x, y)) throw new IllegalArgumentException();
        setLeft(x);
        setRight(y);
    }
    void setLeft(T x) {
        left = x;
    }
    void setRight(T y) {
        right = y;
    }
    final boolean isValid() {
        return validator.apply(left, right);
    }
}
```

It is required that if p instanceof Pair then p.isValid() returns true.

Which is the smallest set of visibility changes to insure this requirement is met?

- A . setLeft and setRight must be protected.
- B . left and right must be private.**
- C . isValid must be public.
- D . left, right, setLeft, and setRight must be private.

Answer: B

Question: 207

Given:

```
public class Tester {  
    public static void main(String[] args) {  
        StringBuilder sb = new StringBuilder(5);  
        sb.append("HOWDY");  
        sb.insert(0, ' ');  
        sb.replace(3, 5, "LL");  
        sb.insert(6, "COW");  
        sb.delete(2, 7);  
        System.out.println(sb.length());  
    }  
}
```

What is the result?

- A . 4
- B . 3
- C . An exception is thrown at runtime.
- D . 5

Answer: D

Question: 208

Which set of commands is necessary to create and run a custom runtime image from Java source files?

- A . java, jdeps
- B . javac, jlink
- C . jar, jlink
- D . javac, jar

Answer: B

Question: 209

Given:

```

import java.io.*;
public class Tester {
    public static void main(String[] args) {
        try {
            doA();
            doB();
        } catch(IOException e) {
            System.out.print("c");
            return;
        } finally{
            System.out.print("d");
        }
        System.out.print("f");
    }
    private static void doA() {
        System.out.print("a");
        if (false) {
            throw new IndexOutOfBoundsException();
        }
    }
    private static void doB() throws FileNotFoundException {
        System.out.print("b");
        if (true) {
            throw new FileNotFoundException();
        }
    }
}

```

What is the result?

- A . The compilation fails.
- B . abdf
- C . abd
- D . adf
- E . abcd

Answer: E

Question: 210

Given the code fragment:

```

int[] secA = { 2, 4, 6, 8, 10 };
int[] secB = { 2, 4, 8, 6, 10 };
int res1 = Arrays.mismatch(secA, secB);
int res2 = Arrays.compare(secA, secB);
System.out.print(res1 + " : " + res2);

```

What is the result?

A . -1 : 2

B . 2 : -1

C . 2 : 3

D . 3 : 0

Answer: B

Question: 211

Which two statements are true about the modular JDK? (Choose two.)

A . The foundational APIs of the Java SE Platform are found in the java.base module.

B . An application must be structured as modules in order to run on the modular JDK.

C . It is possible but undesirable to configure modules' exports from the command line.

D . APIs are deprecated more aggressively because the JDK has been modularized.

Answer: A, C

Question: 212

Given:

```
public class Test{
    private int num = 1;
    private int div = 0;

    public void divide() {
        try {
            num = num / div;
            System.out.print("Exception");
        }
        catch(ArithmaticException ae) { num = 100; }
        catch(Exception e) { num = 200; }
        finally { num = 300; }
        System.out.print(num);
    }
    public static void main(String args[])
    {
        Test test = new Test();
        test.divide();
    }
}
```

What is the output?

A . 300

B . Exception

C . 200

D . 100

Answer: A

Question: 213

Which two modules include APIs in the Java SE Specification? (Choose two.)

A . java.logging

B . java.desktop

C . javafx

D . jdk.httpserver

E . jdk.jartool

Answer: A, D

Question: 214

Given:

```
public interface API {    //line 1
    public void checkValue(Object value)
        throws IllegalArgumentException; //line 2
    public boolean isValueANumber(Object val) {
        if(val instanceof Number) {
            return true;
        }else {
            try {
                Double.parseDouble(val.toString());
                return true;
            }catch (NumberFormatException ex) {
                return false;
            }
        }
    }
}
```

Which two changes need to be made to make this class compile? (Choose two.)

A . Change Line 1 to an abstract class:public abstract class API {

- B . Change Line 2 access modifier to protected:protected void checkValue(Object value)throws
IllegalArgumentException;
- C . Change Line 1 to a class:public class API {
- D . Change Line 1 to extend java.lang.AutoCloseable:public interface API extends AutoCloseable {
- E . Change Line 2 to an abstract method:public abstract void checkValue(Object value)throws
IllegalArgumentException;

Answer: C, E

Question: 215

Given:

```
public class A {  
    private boolean checkValue(int val) {  
        return true;  
    }  
}
```

and

```
public class B extends A {  
    public int modifyVal(int val) {  
        if(checkValue(val)) {  
            return val;  
        } else {  
            return 0;  
        }  
    }  
    public static void Main(String[] args) {  
        B b = new B();  
        System.out.println(b.modifyVal(10));  
    }  
}
```

What is the result?

- A . nothing
- B . It fails to compile.
- C . 0
- D . A java.lang.IllegalArgumentException is thrown.
- E . 10

Answer: B
