

1z0-809.examcollection.premium.exam.207q

Number: 1z0-809

Passing Score: 800

Time Limit: 120 min

File Version: 10.0



1z0-809

Java SE 8 Programmer II

Version 10.0

Exam A

QUESTION 1

Given the definition of the Vehicle class:

```
Class Vehicle {  
    int distance;  
    Vehicle (int x) {  
        this distance = x;  
    }  
    public void increSpeed(int time) {  
        int timeTravel = time; //line n1  
        /line n3  
        class Car {  
            int value = 0;  
            public void speed () {  
                value = distance /timeTravel; //line n2  
                System.out.println ("Velocity with new speed"+value+"kmph");  
            }  
        }  
        speed(); //line n3  
    }  
}
```

and this code fragment:

```
Vehicle v = new Vehicle (100);  
v.increSpeed(60);
```

What is the result?

- A. Velocity with new speed 1 kmph
- B. A compilation error occurs at line n1.
- C. A compilation error occurs at line n2.
- D. A compilation error occurs at line n3.

Correct Answer: A => D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 2

Given:

```
IntStream stream = IntStream.of (1,2,3);  
IntFunction<Integer> inFu= x -> y -> x*y; //line n1  
IntStream newStream = stream.map(inFu.apply(10)); //line n2  
newStream.forEach(System.out::print);
```

Which modification enables the code fragment to compile?

- A. Replace line n1 with:
`IntFunction<UnaryOperator> inFu = x -> y -> x*y;`
- B. Replace line n1 with:
`IntFunction<IntUnaryOperator> inFu = x -> y -> x*y;`
- C. Replace line n1 with:
`BiFunction<IntUnaryOperator> inFu = x -> y -> x*y;`

D. Replace line n2 with:

```
IntStream newStream = stream.map(inFu.applyAsInt (10));
```

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 3

Given the code fragment:

```
List<Integer> values = Arrays.asList (1, 2, 3);
values.stream ()
    .map(n -> n*2)          //line n1
    .peek(System.out::print)   //line n2
    .count();
```

What is the result?

A. 246

- B. The code produces no output.
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 4

Given the code fragment:

```
public class Foo {
    public static void main (String [ ] args) {
        Map<Integer, String> unsortMap = new HashMap< > ( );
        unsortMap.put (10, "z");
        unsortMap.put (5, "b");
        unsortMap.put (1, "d");
        unsortMap.put (7, "e");
        unsortMap.put (50, "j");

        Map<Integer, String> treeMap = new TreeMap <Integer, String> (new
            Comparator<Integer> ( ) {
                @Override public int compare (Integer o1, Integer o2) {return
                    o2.compareTo
                    (o1); } } );

        treeMap.putAll (unsortMap);

        for (Map.Entry<Integer, String> entry : treeMap.entrySet () ) {
            System.out.print (entry.getValue () + " ");
        }
    }
}
```

What is the result?

- A. A compilation error occurs.
- B. d b e z j
- C. j z e b d**
- D. z b d e j

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 5

Which two reasons should you use interfaces instead of abstract classes? (Choose two.)

- A. You expect that classes that implement your interfaces have many common methods or fields, or require access modifiers other than public.
- B. You expect that unrelated classes would implement your interfaces.
- C. You want to share code among several closely related classes.
- D. You want to declare non-static or non-final fields.
- E. You want to take advantage of multiple inheritance of type.

Correct Answer: BE

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://books.google.com.br/books?id=nS2tBQAAQBAJ&pg=PT235&lpg=PT235&dq>You+want+to+share+code+among+several+closely+related+classes.&source=bl&ots=3oY0u2XXN-&sig=uVFS0KB15BqyEgghXnnjJSUdcrE&hl=pt-BR&sa=X&ved=0ahUKEwjlsKe-n6baAhVEhZAKHeiEDTqQ6AEIMDAB#v=onepage&q=You%20want%20to%20share%20code%20among%20several%20closely%20related%20classes.&f=false>

QUESTION 6

Given:

```
public class Counter {  
    public static void main (String[ ] args) {  
        int a = 10;  
        int b = -1;  
        assert (b >=1) : "Invalid Denominator";  
        int c = a / b;  
        System.out.println (c);  
    }  
}
```

What is the result of running the code with the `-ea` option?

- A. -10
- B. 0
- C. An `AssertionError` is thrown.
- D. A compilation error occurs.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 7

Given:

```
class Bird {  
    public void fly () { System.out.print("Can fly"); }  
}  
class Penguin extends Bird {  
    public void fly () { System.out.print("Cannot fly"); }  
}
```

and the code fragment:

```
class Birdie {  
    public static void main (String [ ] args) {  
        fly( () -> new Bird ( ));  
        fly (Penguin : : new);  
    }  
    /* line n1 */  
}
```

Which code fragment, when inserted at line n1, enables the `Birdie` class to compile?

- A. static void fly (Consumer<Bird> bird) {
 bird :: fly ();
}
- B. static void fly (Consumer<? extends Bird> bird) {
 bird.accept() fly ();
}
- C. static void fly (Supplier<Bird> bird) {
 bird.get() fly ();
}
- D. static void fly (Supplier<? extends Bird> bird) {
 bird::fly();
}

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 8

Given:

```
1. abstract class Shape { System.out.println ("Shape"); }  
2.  
3.     protected void area ( ) { System.out.println ("Shape"); }  
4. }  
5.  
6. class Square extends Shape {  
7.     int side;  
8.     Square int side {
```

```

9.         /* insert code here */
10.        this.side = side;
11.    }
12.    public void area ( ) { System.out.println ("Square");      }
13. }
14. class Rectangle extends Square {
15.     int len, br;
16.     Rectangle (int x, int y)   {
17.         /* insert code here */
18.         len = x, br = y;
19.     }
20. void area ( ) { System.out.println ("Rectangle");      }
21. }

```

Which two modifications enable the code to compile? (Choose two.)

- A. At line 1, remove `abstract`
- B. At line 9, insert `super ();`
- C. At line 12, remove `public`
- D. At line 17, insert `super (x);`
- E. At line 17, insert `super (); super.side = x;`
- F. At line 20, use `public void area () {`

Correct Answer: DF

Section:(none)

Explanation

Explanation/Reference:

QUESTION 9

Given:

```

class Sum extends RecursiveAction           {                               //line n1
    static final int THRESHOLD_SIZE    = 3;
    int stIndex, lstIndex;
    int [ ] data;
    public Sum (int [ ]data, int start, int end)      {
        this.data = data;
        this.stIndex = start;
        this.lstIndex = end;
    }
    protected void compute ( )      {
        int sum = 0;
        if (lstIndex - stIndex <= THRESHOLD_SIZE)  {
            for (int i = stIndex; i < lstIndex; i++)  {
                sum += data [i];
            }
            System.out.println(sum);
        } else {
            new Sum (data, stIndex + THRESHOLD_SIZE, lstIndex).fork();
            new Sum (data, stIndex,
                    Math.min (lstIndex, stIndex + THRESHOLD_SIZE)
                    ).compute ();
        }
    }
}

```

and the code fragment:

```
ForkJoinPool fjPool = new ForkJoinPool ( );
int data [ ] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
fjPool.invoke (new Sum (data, 0, data.length));
```

and given that the sum of all integers from 1 to 10 is 55.

Which statement is true?

- A. The program prints several values that total 55.
- B. The program prints 55.
- C. A compilation error occurs at line n1.
- D. The program prints nothing.

Correct Answer: A

Section:(none)

Explanation

Explanation/Reference:

QUESTION 10

Given the content of Operator.java, EngineOperator.java, and Engine.java files:

```
Operator.java:
public abstract class Operator {
    protected void turnON();
    protected void turnOFF();
}

EngineOperator.java:
public class EngineOperator extends Operator{
    public final void turnON() { System.out.print("ON "); }
    public final void turnOFF() { System.out.println("OFF"); }
}

Engine.java:
public class Engine{
    Operator m = new EngineOperator();
    public void operate() {
        m.turnON();
        m.turnOFF();
    }
}
```

and the code fragment:

```
Engine carEngine = new Engine();
carEngine.operate();
```

What is the result?

- A. The Engine.java file fails to compile.
- B. The EngineOperator.java file fails to compile.

C. The Operator.java file fails to compile.

D. ON OFF

Correct Answer: A

=> C Section:

(none)

Explanation

Explanation/Reference:

QUESTION 11

Given the code fragment:

```
Stream<List<String>> iStr= Stream.of (
    Arrays.asList ("1", "John"),
    Arrays.asList ("2", null));
Stream<String> nInSt = iStr.flatMapToInt ((x) -> x.stream ());
nInSt.forEach (System.out :: print);
```

What is the result?

- A. 1John2null
- B. 12
- C. A NullPointerException is thrown at run time.
- D. A compilation error occurs.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 12

Given the code fragment:

```
Path file = Paths.get ("courses.txt");
// line n1
```

Assume the courses.txt is accessible.

Which code fragment can be inserted at line n1 to enable the code to print the content of the courses.txt file?

- A. List<String> fc = Files.list(file);
 fc.stream().forEach (s -> System.out.println(s));
- B. Stream<String> fc = Files.readAllLines (file);
 fc.forEach (s -> System.out.println(s));
- C. List<String> fc = readAllLines(file);
 fc.stream().forEach (s -> System.out.println(s));
- D. Stream<String> fc = Files.lines (file);
 fc.forEach (s -> System.out.println(s));

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 13

Given the code fragment:

```
public void recDelete (String dirName) throws IOException {
    File [ ] listOfFiles = new File (dirName) .listFiles();
    if (listOfFiles ! = null && listOfFiles.length >0)      {
        for (File aFile : listOfFiles)      {
            if (aFile.isDirectory ())      {
                recDelete (aFile.getAbsolutePath ());
            } else {
                if (aFile.getName ().endsWith (".class"))
                    aFile.delete ();
            }
        }
    }
}
```

Assume that `Projects` contains subdirectories that contain `.class` files and is passed as an argument to the `recDelete ()` method when it is invoked.

What is the result?

- A. The method deletes all the `.class` files in the `Projects` directory and its subdirectories.
- B. The method deletes the `.class` files of the `Projects` directory only.
- C. The method executes and does not make any changes to the `Projects` directory.
- D. The method throws an `IOException`.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 14

Given the code fragments:

```
4. void doStuff() throws ArithmeticException, NumberFormatException, Exception
{
5.     if (Math.random() >-1 throw new Exception ("Try again");
6. }
```

and

```
24. try {
25.     doStuff ( );
26. } catch (ArithmaticException | NumberFormatException | Exception e)  {
27.     System.out.println (e.getMessage()); }
28. catch (Exception e)      {
29.     System.out.println (e.getMessage()); }
30. }
```

Which modification enables the code to print Try again?

- A. Comment the lines 28, 29 and 30.
- B. Replace line 26 with:
 `} catch (Exception | ArithmaticException | NumberFormatException e) {`
- C. Replace line 26 with:

```
    } catch (ArithmeticException | NumberFormatException e) {  
D. Replace line 27 with:  
    throw e;
```

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 15

Given the definition of the Country class:

```
public class country {  
    public enum Continent {ASIA, EUROPE}  
    String name;  
    Continent region;  
  
    public Country (String na, Continent reg) {  
        name = na, region = reg;  
    }  
    public String getName () {return name;}  
    public Continent getRegion () {return region;}  
}
```

and the code fragment:

```
List<Country> couList = Arrays.asList (  
    new Country ("Japan", Country.Continent.ASIA),  
    new Country ("Italy", Country.Continent.EUROPE),  
    new Country ("Germany", Country.Continent.EUROPE));  
Map<Country.Continent, List<String>> regionNames = couList.stream ()  
.collect(Collectors.groupingBy (Country ::getRegion,  
    Collectors.mapping(Country::getName, Collectors.toList())));  
System.out.println(regionNames);
```

- A. {EUROPE = [Italy, Germany], ASIA = [Japan]}
- B. {ASIA = [Japan], EUROPE = [Italy, Germany]}
- C. {EUROPE = [Germany, Italy], ASIA = [Japan]}
- D. {EUROPE = [Germany], EUROPE = [Italy], ASIA = [Japan]}

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 16

Given the code fragment:

```
Map<Integer, String> books = new TreeMap<>();  
books.put (1007, "A");  
books.put (1002, "C");  
books.put (1001, "B");  
books.put (1003, "B");  
System.out.println (books);
```

What is the result?

- A. {1007 = A, 1002 = C, 1001 = B, 1003 = B}
- B. {1001 = B, 1002 = C, 1003 = B, 1007 = A}
- C. {1002 = C, 1003 = B, 1007 = A}
- D. {1007 = A, 1001 = B, 1003 = B, 1002 = C}

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

Reference:

TreeMap inherits SortedMap and automatically sorts the element's key

QUESTION 17

Given:

```
class Book {
    int id;
    String name;
    public Book (int id, String name)  {
        this.id = id;
        this.name = name;
    }
    public boolean equals (Object obj) {           //line n1
        boolean output = false;
        Book b = (Book) obj;
        if (this.name.equals(b.name)) }
            output = true;
        }
        return output;
    }
}
```

and the code fragment:

```
Book b1 = new Book (101, "Java Programming");
Book b2 = new Book (102, "Java Programming");
System.out.println (b1.equals(b2));           //line n2
```

Which statement is true?

- A. The program prints true.
- B. The program prints false.
- C. A compilation error occurs. To ensure successful compilation, replace line n1 with:
`boolean equals (Book obj) {`
- D. A compilation error occurs. To ensure successful compilation, replace line n2 with:
`System.out.println (b1.equals((Object) b2));`

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 18

Given the content of /resources/Message.properties:

```
welcome1="Good day!"
```

and given the code fragment:

```
Properties prop = new Properties ();
FileInputStream fis = new FileInputStream ("resources/Message.properties");
prop.load(fis);
System.out.println(prop.getProperty("welcome1"));
System.out.println(prop.getProperty("welcome2", "Test")); //line n1
System.out.println(prop.getProperty("welcome3"));
```

What is the result?

- A. Good day!
Test
followed by an Exception stack trace
- B. Good day!
followed by an Exception stack trace
- C. Good
day!
Test
null
- D. A compilation error occurs at line n1.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 19

Which action can be used to load a database driver by using JDBC3.0?

- A. Add the driver class to the META-INF/services folder of the JAR file.
- B. Include the JDBC driver class in a `jdbc.properties` file.
- C. Use the `java.lang.Class.forName` method to load the driver class.
- D. Use the `DriverManager.getDriver` method to load the driver class.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 20

Given the code fragment:

```
Path p1 = Paths.get("/Pics/MyPic.jpeg");
System.out.println (p1.getNameCount() +
    ":" + p1.getName(1) +
    ":" + p1.getFileName());
```

Assume that the `Pics` directory does NOT exist.

What is the result?

- A. An exception is thrown at runtime.

- B. 2:MyPic.jpeg: MyPic.jpeg
- C. 3::MyPic.jpeg
- D. 2:Pics: MyPic.jpeg

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 21

Given the code fragments:

```
class MyThread implements Runnable {  
    private static AtomicInteger count = new AtomicInteger (0);  
    public void run () {  
        int x = count.incrementAndGet();  
        System.out.print (x+" ");  
    }  
}
```

and

```
Thread thread1 = new Thread(new MyThread());  
Thread thread2 = new Thread(new MyThread());  
Thread thread3 = new Thread(new MyThread());  
  
Thread [] ta = {thread1, thread2, thread3};  
for (int x= 0; x < 3; x++) {  
    ta[x].start();  
}
```

Which statement is true?

- A. The program prints 1 2 3 and the order is unpredictable.
- B. The program prints 1 2 3.
- C. The program prints 1 1 1.
- D. A compilation error occurs.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 22

Given the code fragment:

```
public static void main (String [ ] args) throws IOException {  
    BufferedReader br = new BufferedReader (new InputStreamReader (System.in));  
    System.out.print ("Enter GDP: ");  
    //line 1  
}
```

Which code fragment, when inserted at line 1, enables the code to read the GDP from the user?

- A. int GDP = Integer.parseInt (br.readLine());
- B. int GDP = br.read();
- C. int GDP = br.nextInt();
- D. int GDP = Integer.parseInt (br.next());

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 23

Assuming that the file /data/december/log.txt does not exist and given the code fragment:

```
Path source = Paths.get ("/data/december/log.txt");
Path destination = Paths.get("/data");
Files.copy (source, destination);
```

What is the result?

- A. A file with the name log.txt is created in the /data directory and the content of the /data/december/log.txt file is copied to it.
- B. The program executes successfully and does NOT change the file system.
- C. A FileNotFoundException is thrown at run time.
- D. A NoSuchElementException is thrown at run time.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 24

Given:

```
class Student {
    String course, name, city;
    public Student (String name, String course, String city) {
        this.course = course; this.name = name; this.city = city;
    }
    public String toString() {
        return course + ":" + name + ":" + city;
    }
}
```

and the code fragment:

```
List<Student> stds = Arrays.asList(
    new Student ("Jessy", "Java ME", "Chicago"),
    new Student ("Helen", "Java EE", "Houston"),
    new Student ("Mark", "Java ME", "Chicago"));
stds.stream()
    .collect(Collectors.groupingBy(Student::getCourse))
    .forEach(src, res) -> System.out.println(src));
```

What is the result?

- A. [Java EE: Helen:Houston]
[Java ME: Jessy:Chicago, Java ME: Mark:Chicago]
- B. Java EE
Java ME
- C. [Java ME: Jessy:Chicago, Java ME: Mark:Chicago]
[Java EE: Helen:Houston]
- D. A compilation error occurs.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 25

Given the code fragments:

```
interface CourseFilter extends Predicate<String> {
    public default boolean test (String str) {
        return str.equals ("Java");
    }
}
```

and

```
List<String> strs = Arrays.asList("Java", "Java EE", "Java ME");
Predicate<String> cf1 = s -> s.length() > 3;
Predicate cf2 = new CourseFilter() { //line n1
    public boolean test (String s) {
        return s.contains ("Java");
    }
};
long c = strs.stream()
    .filter(cf1)
    .filter(cf2) //line n2
.count();
System.out.println(c);
```

What is the result?

- A. 2
- B. 3
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

trinhduc.giang
2021-07-10 16:56:12

pending

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 26

Given:

```

public class Emp {
    String fName;
    String lName;
    public Emp (String fn, String ln) {
        fName = fn;
        lName = ln;
    }
    public String getfName() { return fName; }
    public String getlName() { return lName; }
}

```

and the code fragment:

```

List<Emp> emp = Arrays.asList (
    new Emp ("John", "Smith"),
    new Emp ("Peter", "Sam"),
    new Emp ("Thomas", "Wale"));
emp.stream()
    //line n1
    .collect(Collectors.toList());

```

Which code fragment, when inserted at line n1, sorts the employees list in descending order of fName and then ascending order of lName?

- A. .sorted (Comparator.comparing(Emp::getfName).reserved().thenComparing(Emp::getlName))
- B. .sorted (Comparator.comparing(Emp::getfName).thenComparing(Emp::getlName))
- C. .map(Emp::getfName).sorted(Comparator.reverseOrder())
- D. .map(Emp::getfName).sorted(Comparator.reverseOrder()).map(Emp::getlName).reserved

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 27

Given:

```

public enum USCurrency {
    PENNY (1),
    NICKLE(5),
    DIME (10),
    QUARTER(25);

    private int value;

    public USCurrency(int value) {
        this.value = value;
    }
    public int getValue() {return value;}
}

public class Coin {
    public static void main (String[] args) {
        USCurrency usCoin =new USCurrency.DIME;
        System.out.println(usCoin.getValue());
    }
}

```

Which two modifications enable the given code to compile? (Choose two.)

- A. Nest the `USCurrency` enumeration declaration within the `Coin` class.
- B. Make the `USCurrency` enumeration constructor `public`.
- C. Remove the `new` keyword from the instantiation of `usCoin`.
- D. Make the `getValue()` method `public`.
- E. Add the `final` keyword in the declaration of `value`.

Correct Answer: BC

Section: (none)

Explanation

Explanation/Reference:

QUESTION 28

Given:

```
class ImageScanner implements AutoCloseable {  
    public void close () throws Exception {  
        System.out.print ("Scanner closed.");  
    }  
    public void scanImage () throws Exception {  
        System.out.print ("Scan.");  
        throw new Exception("Unable to scan.");  
    }  
}  
class ImagePrinter implements AutoCloseable {  
    public void close () throws Exception {  
        System.out.print ("Printer closed.");  
    }  
    public void printImage () {System.out.print("Print.");}  
}
```

and this code fragment:

```
try (ImageScanner ir = new ImageScanner();  
     ImagePrinter iw = new ImagePrinter()) {  
    ir.scanImage();  
    iw.printImage();  
} catch (Exception e) {  
    System.out.print(e.getMessage());  
}
```

What is the result?

- A. Scan.Printer closed. Scanner closed. Unable to scan.
- B. Scan.Scanner closed. Printer closed.Unable to scan.
- C. Scan. Unable to scan.
- D. Scan. Unable to scan. Scanner closed.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 29

Given the structure of the STUDENT table:

```
Student (id INTEGER, name VARCHAR)
```

Given:

```
public class Test {
    static Connection newConnection = null;
    public static Connection get DBConnection () throws SQLException {
        try (Connection con = DriverManager.getConnection(URL, username,
password)) {
            newConnection = con;
        }
        return newConnection;
    }
    public static void main (String [] args) throws SQLException {
        get DBConnection ();
        Statement st = newConnection.createStatement();
        st.executeUpdate("INSERT INTO student VALUES (102, 'Kelvin')");
    }
}
```

Assume that:

The required database driver is configured in the classpath.

The appropriate database is accessible with the URL, userName, and passWord exists.

The SQL query is valid.

What is the result?

- A. The program executes successfully and the STUDENT table is updated with one record.
- B. The program executes successfully and the STUDENT table is NOT updated with any record.
- C. A SQLException is thrown as runtime.
- D. A NullPointerException is thrown as runtime.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 30

Given the code fragments:

```
class Employee {
    Optional<Address> address;
    Employee (Optional<Address> address) {
        this.address = address;
    }
    public Optional<Address> getAddress() { return address; }
}

class Address {
    String city = "New York";
    public String getCity { return city; }
    public String toString() {
        return city;
    }
}
```

and

```
Address address = null;
Optional<Address> addrs1 = Optional.ofNullable (address);
Employee e1 = new Employee (addrs1);
String eAddress = (addrs1.isPresent()) ? addrs1.get().getCity() : "City Not
available";
```

What is the result?

- A. New York
- B. City Not available
- C. null
- D. A NoSuchElementException is thrown at run time.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 31

Given the code fragment:

```
Stream<Path> files = Files.walk(Paths.get(System.getProperty("user.home")));
files.forEach (fName -> {
    try {
        Path aPath = fName.toAbsolutePath();           //line n1
        System.out.println(fName + ":"
                           + Files.readAttributes(aPath,
Basic.File.Attributes.class).creationTime
());
    } catch (IOException ex) {
        ex.printStackTrace();
    });
});
```

What is the result?

- A. All files and directories under the home directory are listed along with their attributes.
- B. A compilation error occurs at line n1.
- C. The files in the home directory are listed along with their attributes.
- D. A compilation error occurs at line n2.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 32

Given:

```
class Vehicle {
    int vno;
```

```

String name;

public Vehicle (int vno, String name) {
    this.vno = vno;
    this.name = name;
}
public String toString () {
    return vno + ":" + name;
}
}

```

and this code fragment:

```

Set<Vehicle> vehicles = new TreeSet <> ();
vehicles.add(new Vehicle (10123, "Ford"));
vehicles.add(new Vehicle (10124, "BMW"));
System.out.println(vehicles);

```

What is the result?

- A. 10123 Ford
10124 BMW
- B. 10124 BMW
10123 Ford
- C. A compilation error occurs.
- D. A ClassCastException is thrown at run time.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 33

Given that course.txt is accessible and contains:

```
Course : : Java
```

and given the code fragment:

```

public static void main (String[ ] args) {
    int i;
    char c;
    try (FileInputStream fis = new FileInputStream ("course.txt");
        InputStreamReader isr = new InputStreamReader(fis);) {
        while (isr.ready()) { //line n1
            isr.skip(2);
            i = isr.read ();
            c = (char) i;
            System.out.print(c);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

What is the result?

A. ur :: va

- B. ueJa
- C. The program prints nothing.
- D. A compilation error occurs at line n1.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 34

Given:

```
public class Test<T>      {  
    private T t;  
    public T get ()  {  
        return t;  
    }  
    public void set (T t)      {  
        this.t = t;  
    }  
    public static void main (String args [ ] )      {  
        Test<String> type = new Test<>();  
        Test type1 = new Test ();           //line n1  
        type.set("Java");  
        type1.set(100);                  //line n2  
        System.out.print(type.get() + " " + type1.get());  
    }  
}
```

What is the result?

- A. Java 100
- B. java.lang.String@<hashcode>java.lang.Integer@<hashcode>
- C. A compilation error occurs. To rectify it, replace line n1 with:
`Test<Integer> type1 = new Test<>();`
- D. A compilation error occurs. To rectify it, replace line n2 with:
`type1.set (Integer(100));`

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 35

Given the definition of the Vehicle class:

```
class Vehicle  {  
    String name;  
    void setName (String name)      {  
        this.name = name;  
    }  
    String getName()      {  
        return name;  
    }  
}
```

Which action encapsulates the Vehicle class?

- A. Make the Vehicle class public.
- B. Make the name variable public.
- C. Make the getName method public.
- D. Make the name variable private.
- E. Make the setName method private.
- F. Make the getName method private.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 36

Given:

```
public class Product {
    int id; int price;
    public Product (int id, int price) {
        this.id = id;
        this.price = price;
    }
    public String toString() { return id + ":" + price; }
}
```

and the code fragment:

```
List<Product> products = Arrays.asList(new Product(1, 10),
    new Product(2, 30),
    new Product(3, 20));
Product p = products.stream().reduce(new Product(4, 0), (p1, p2) -> {
    p1.price+=p2.price;
    return new Product(p1.id, p1.price);});
products.add(p);
products.stream().parallel()
    .reduce((p1, p2) -> p1.price > p2.price ? p1 : p2)
    .ifPresent(System.out::println);
```

What is the result?

- A. 2 : 30
- B. 4 : 0
- C. 4 : 60**
- D. 4 : 60
2 : 30
3 : 20
1 : 10
- E. The program prints nothing.

Correct Answer: C => sửa đè

Section: (none) Explanation

Explanation/Reference:**QUESTION 37**

Given the code fragments:

```
public class Book implements Comparator<Book> {
    String name;
    double price;
    public Book () {}
    public Book(String name, double price) {
        this.name = name;
        this.price = price;
    }
    public int compare(Book b1, Book b2) {
        return b1.name.compareTo(b2.name);
    }
    public String toString() {
        return name + ":" + price;
    }
}
```

and

```
List<Book>books = Arrays.asList (
    new Book ("Beginning with Java", 2),
    new book ("A Guide to Java Tour", 3)
);
Collections.sort(books, new Book());
System.out.print(books);
```

What is the result?

- A. [A Guide to Java Tour:3.0, Beginning with Java:2.0]
- B. [Beginning with Java:2.0, A Guide to Java Tour:3.0]
- C. A compilation error occurs because the Book class does not override the abstract method compareTo().
- D. An Exception is thrown at run time.

Correct Answer: D

=> A

Section: (none)

Explanation

Explanation/Reference:

Explanation:

If asList is changed to List, the output would be B, Beginning with java:2.0, A guide to Java Tour:3.0.

QUESTION 38

Given the code fragment:

```
List<String> listVal = Arrays.asList("Joe", "Paul", "Alice", "Tom");
System.out.println (
    // line n1
);
```

Which code fragment, when inserted at line n1, enables the code to print the count of string elements whose length is greater than three?

- A. listVal.stream().filter(x -> x.length()>3).count()
- B. listVal.stream().map(x -> x.length()>3).count()

- C. `listVal.stream().peek(x -> x.length()>3).count().get()`
- D. `listVal.stream().filter(x -> x.length()>3).mapToInt(x -> x).count()`

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 39

Given the code fragments:

```
class Caller implements Callable<String>      {
    String str;
    public Caller (String s) {this.str=s;}
    public String call() throws Exception { return str.concat ("Caller");}
}
class Runner implements Runnable      {
String str;
    public Runner (String s) {this.str=s;}
    public void run () { System.out.println (str.concat ("Runner")); }
}
```

and

```
public static void main (String[] args) InterruptedException, ExecutionException
{
    ExecutorService es = Executors.newFixedThreadPool(2);
    Future f1 = es.submit (new Caller ("Call"));
    Future f2 = es.submit (new Runner ("Run"));
    String str1 = (String) f1.get();
    String str2 = (String) f2.get();           //line n1
    System.out.println(str1+ ":" + str2);
}
```

What is the result?

- A. The program prints:
Run Runner
Call Caller : null
And the program does not terminate.
- B. The program terminates after printing:
Run Runner
Call Caller : Run
- C. A compilation error occurs at line n1.
- D. An Execution is thrown at run time.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 40

Given:

```
public class Canvas implements Drawable {
```

```

        public void draw ()      {   }
}

public abstract class Board extends Canvas {    }

public class Paper extends Canvas {
    protected void draw (int color)      {      }
}

public class Frame extends Canvas implements Drawable {
    public void resize ()      {   }
}

public interface Drawable {
    public abstract void draw ();
}

```

Which statement is true?

- A. Board does not compile.
- B. Paper does not compile.
- C. Frame does not compile.
- D. Drawable does not compile.
- E. All classes compile successfully.

Correct Answer: E

Section: (none)

Explanation

Explanation/Reference:

QUESTION 41

Given the code fragment:

```

List<String> str = Arrays.asList ("my", "pen", "is", "your", "pen");
Predicate<String> test = s -> {
    int i = 0;
    boolean result = s.contains ("pen");
    System.out.print(i++) + ":";
    return result;
};
str.stream()
    .filter(test)
    .findFirst()
    .ifPresent(System.out ::print);

```

What is the result?

- A. 0 : 0 : pen
- B. 0 : 1 : pen
- C. 0 : 0 : 0 : 0 : pen
- D. 0 : 1 : 2 : 3 : 4 :
- E. A compilation error occurs.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 42

Given the code fragment:

```
List<String> empDetails = Arrays.asList("100, Robin, HR",
                                         "200, Mary, AdminServices",
                                         "101, Peter, HR");
empDetails.stream()
    .filter(s-> s.contains("1"))
    .sorted()
    .forEach(System.out::println); //line n1
```

What is the result?

- A. 100, Robin, HR
101, Peter, HR
- B. A compilation error occurs at line n1.
- C. 100, Robin, HR
101, Peter, HR
200, Mary, AdminServices
- D. 100, Robin, HR
200, Mary, AdminServices
101, Peter, HR

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 43

Given:

```
interface Rideable {Car getCar (String name); }
class Car {
    private String name;
    public Car (String name) {
        this.name = name;
    }
}
```

Which code fragment creates an instance of Car?

- A. Car auto = Car ("MyCar") : : new;
- B. Car auto = Car : : new;
 Car vehicle = auto : : getCar("MyCar");
- C. Rideable rider = Car : : new;
 Car vehicle = rider.getCar("MyCar");
- D. Car vehicle = Rideable : : new : : getCar("MyCar");

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 44

Which statement is true about the single abstract method of the `java.util.function.Function` interface?

- A. It accepts one argument and returns `void`.
- B. It accepts one argument and returns `boolean`.
- C. It accepts one argument and always produces a result of the same type as the argument.
- D. It accepts an argument and produces a result of any data type.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 45

Which statement is true about the `DriverManager` class?

- A. It returns an instance of database.
- B. It executes SQL statements against the database.
- C. It loads the database driver class mentioned in the `jdbc.drivers` property
- D. it is written by different vendors for their specific database.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://javaconceptoftheday.com/drivermanager-class-in-java/>

QUESTION 46

Given the code fragment:

```
List<Integer> nums = Arrays.asList (10, 20, 8);
System.out.println (
    //line n1
);
```

Which code fragment must be inserted at `line n1` to enable the code to print the maximum number in the `nums` list?

- A. `nums.stream().max(Comparator.comparing(a -> a)).get()`
- B. `nums.stream().max(Integer :: max).get()`
- C. `nums.stream().max()`
- D. `nums.stream().map(a -> a).max()`

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 47

Given:

```
public final class Cream {  
    public void prepare() {}  
}  
public class Cake extends Cream{  
    public void bake(int min, int temp) {}  
    public void mix() {}  
}  
public class Shop {  
    private Cake c = new Cake();  
    private final double discount = 0.25;  
    public void makeReady () { c.bake(10, 120); }  
}  
public class Bread extends Cake {  
    public void bake(int minutes, int temperature) {}  
    public void addToppings() {}  
}
```

Which statement is true?

- A. A compilation error occurs in Cream.
- B. A compilation error occurs in Cake.
- C. A compilation error occurs in Shop.
- D. A compilation error occurs in Bread.
- E. All classes compile successfully.

Correct Answer: E

=> B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 48

Which two statements are true about localizing an application? (Choose two.)

- A. Support for new regional languages does not require recompilation of the code.
- B. Textual elements (messages and GUI labels) are hard-coded in the code.
- C. Language and region-specific programs are created using localized data.
- D. Resource bundle files include data and currency information.
- E. Language codes use lowercase letters and region codes use uppercase letters.

Correct Answer: AE

Section: (none)

Explanation

Explanation/Reference:

Reference: <http://docs.oracle.com/javase/7/docs/technotes/guides/intl/>

QUESTION 49

Which statement is true about `java.util.stream.Stream`?

- A. A stream cannot be consumed more than once.
- B. The execution mode of streams can be changed during processing.
- C. Streams are intended to modify the source data.
- D. A parallel stream is always faster than an equivalent sequential stream.
- E. Stream operation accepts lambda expressions as its parameters.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 50

The `data.doc`, `data.txt` and `data.xml` files are accessible and contain text.

Given the code fragment:

```
Stream<Path> paths = Stream.of (Paths. get("data.doc"),
    Paths. get("data.txt"),
    Paths. get("data.xml"));
paths.filter(s-> s.toString().contains("data")).forEach(
    s -> {
        try {
            Files.readAllLines(s)
                .stream()
                .forEach(System.out::println); //line n1
        } catch (IOException e) {
            System.out.println("Exception");
        }
    }
);
```

What is the result?

- A. The program prints the content of `data.txt` file.
- B. The program prints:
 Exception
 <<The content of the `data.txt` file>>
 <<The content of the `data.xml` file>>
- D. The program prints the content of the three files.**

Correct Answer: A

=> D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 51

Given:

```
final class Folder {           //line n1
    //line n2
```

```

        public void open () {
            System.out.print("Open");
        }
    }
public class Test {
    public static void main (String [] args) throws Exception {
        try (Folder f = new Folder()) {
            f.open();
        }
    }
}

```

Which two modifications enable the code to print Open Close? (Choose two.)

- A. Replace line n1 with:

```
class Folder implements AutoCloseable {
```

- B. Replace line n1 with:

```
class Folder extends Closeable {
```

- C. Replace line n1 with:

```
class Folder extends Exception {
```

- D. At line n2, insert:

```
final void close () {
    System.out.print("Close");
}
```

- E. At line n2, insert:

```
public void close () throws IOException {
    System.out.print("Close");
}
```

Correct Answer: AE

Section: (none)

Explanation

Explanation/Reference:

QUESTION 52

You want to create a singleton class by using the Singleton design pattern.

Which two statements enforce the singleton nature of the design? (Choose two.)

- A. Make the class static.

- B. Make the constructor private.

- C. Override equals() and hashCode() methods of the java.lang.Object class.

- D. Use a public reference to point to the single instance.

- E. Implement the Serializable interface.

- F. Make the single instance created static and final.

Correct Answer: BF

Section: (none)

Explanation

Explanation/Reference:

QUESTION 53

Given the code fragment:

```

9. Connection conn = DriverManager.getConnection(dbURL, userName, passWord);
10. String query = "SELECT id FROM Employee";
11. try (Statement stmt = conn.createStatement()) {
12.     ResultSet rs = stmt.executeQuery(query);
13.     stmt.executeQuery("SELECT id FROM Customer");
14.     while (rs.next()) {
15.         //process the results
16.         System.out.println("Employee ID: " + rs.getInt("id"));
17.     }
18. } catch (Exception e) {
19.     System.out.println ("Error");
20. }

```

Assume that:

The required database driver is configured in the classpath.

The appropriate database is accessible with the dbURL, userName, and passWord exists.

The Employee and Customer tables are available and each table has id column with a few records and the SQL queries are valid.

What is the result of compiling and executing this code fragment?

- A. The program prints employee IDs.
- B. The program prints customer IDs.
- C. The program prints Error.
- D. compilation fails on line 13.

Correct Answer: C

=> A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 54

Given the code fragment:

```

List<Integer> codes = Arrays.asList (10, 20);
UnaryOperator<Double> uo = s -> s +10.0;
codes.replaceAll(uo);
codes.forEach(c -> System.out.println(c));

```

What is the result?

- A. 20.0
30.0
- B. 10.0
20.0
- C. A compilation error occurs.
- D. A NumberFormatException is thrown at run time.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 55

Given:

```

public class Customer {
    private String fName;
    private String lName;
    private static int count;
    public customer (String first, String last) {fName = first, lName = last;
    ++count;}
    static { count = 0; }
    public static int getCount() {return count; }
}

public class App {
    public static void main (String [] args) {
        Customer c1 = new Customer("Larry", "Smith");
        Customer c2 = new Customer("Pedro", "Gonzales");
        Customer c3 = new Customer("Penny", "Jones");
        Customer c4 = new Customer("Lars", "Svenson");
        c4 = null;
        c3 = c2;
        System.out.println (Customer.getCount());
    }
}

```

What is the result?

- A. 0
- B. 2
- C. 3
- D. 4
- E. 5

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 56

Given:

Item table

- ID, INTEGER: PK
- DESCRIPT, VARCHAR(100)
- PRICE, REAL
- QUANTITY< INTEGER

And given the code fragment:

```

9. try {
10.     ConnecGheryn = "Select * FROM Item WHERE ID = 110";username, password);
11.     Statement stmt = conn.createStatement();
12.     ResultSet rs = stmt.executeQuery(query);
13.     while(rs.next()) {
14.         System.out.println("ID:      " + rs.getInt("Id"));
15.         System.out.println("Description:  " + rs.getString("Descrip"));
16.         System.out.println("Price:      " + rs.getDouble("Price"));
17.         System.out.println("Quantity:   " + rs.getInt("Quantity"));
18.

```

```
19.    }
20. } catch (SQLException se) {
21.     System.out.println("Error");
22. }
```

Assume that:

- The required database driver is configured in the classpath.
- The appropriate database is accessible with the dbURL, userName, and passWord exists.
- The SQL query is valid.

What is the result?

- A. An exception is thrown at runtime.
- B. Compilation fails.
- C. The code prints Error.
- D. The code prints information about Item 110.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 57

Given:

```
class Worker extends Thread {
    CyclicBarrier cb;
    public Worker(CyclicBarrier cb) { this.cb = cb; }
    public void run () {
        try {
            cb.await();
            System.out.println("Worker...");
        } catch (Exception ex) { }
    }
}
class Master implements Runnable { //line n1
    public void run () {
        System.out.println("Master...");
    }
}
```

and the code fragment:

```
Master master = new Master();
//line n2
Worker worker = new Worker(cb);
worker.start();
```

You have been asked to ensure that the run methods of both the Worker and Master classes are executed. Which modification meets the requirement?

- A. At line n2, insert CyclicBarrier cb = new CyclicBarrier(2, master);
- B. At line n2, insert CyclicBarrier cb = new CyclicBarrier(1);
- C. At line n2, insert CyclicBarrier cb = new CyclicBarrier(1, master);
- D. At line n2, insert CyclicBarrier cb = new CyclicBarrier(master);

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 58

Given the code fragment:

```
String str = "Java is a programming language";
ToIntFunction<String> indexVal = str::indexOf; //line n1
int x = indexVal.applyAsInt("Java");           //line n2
System.out.println(x);
```

What is the result?

- A. 0
- B. 1
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 59

Given the code fragment:

```
List<String> codes = Arrays.asList ("DOC", "MPEG", "JPEG");
codes.forEach (c -> System.out.print(c + " "));
String fmt = codes.stream()
    .filter (s-> s.contains ("PEG"))
    .reduce((s, t) -> s + t).get();
System.out.println("\n" + fmt);
```

What is the result?

- A. DOC MPEG JPEG
 MPEGJPEG
- B. DOC MPEG MPEGJPEG
 MPEGMPEGJPEG
- C. MPEGJPEG
 MPEGJPEG
- D. java.util.NoSuchElementException is thrown.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 60

Given the code fragment:

```
List<String> nL = Arrays.asList("Jim", "John", "Jeff");
Function<String, String> funVal = s -> "Hello : ".contact(s);
nL.stream()
    .map(funVal)
    .peek(System.out::print);
```

What is the result?

- A. Hello : Jim Hello : John Hello : Jeff
- B. Jim John Jeff
- C. The program prints nothing.
- D. A compilation error occurs.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 61

Given:

```
public interface Moveable<Integer> {
    public default void walk (Integer distance) {System.out.println("Walking");}
    public void run(Integer distance);
}
```

Which statement is true?

- A. Moveable can be used as below:

```
Moveable<Integer> animal = n -> System.out.println("Running" + n);
animal.run(100);
animal.walk(20);
```

- B. Moveable can be used as below:

```
Moveable<Integer> animal = n -> n + 10;
animal.run(100);
animal.walk(20);
```

- C. Moveable can be used as below:

```
Moveable animal = (Integer n1, Integer n2) -> n+ n2;
animal.run(100);
animal.walk(20);
```

- D. Movable cannot be used in a lambda expression.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 62

Which two code blocks correctly initialize a Locale variable? (Choose two.)

- A. Locale loc1 = "UK";
- B. Locale loc2 = Locale.getInstance("ru");

- C. Locale loc3 = Locale.getLocaleFactory("RU");
- D. Locale loc4 = Locale.UK;
- E. Locale loc5 = new Locale ("ru", "RU");

Correct Answer: DE

Section: (none)

Explanation

Explanation/Reference:

QUESTION 63

Given:

```
class FuelNotAvailException extends Exception { }
class Vehicle {
    void ride() throws FuelNotAvailException { //line n1
        System.out.println("Happy Journey!");
    }
}
class SolarVehicle extends Vehicle {
    public void ride () throws Exception { //line n2
        super ride ();
    }
}
```

and the code fragment:

```
public static void main (String[] args) throws FuelNotAvailException, Exception
{
    Vehicle v = new SolarVehicle ();
    v.ride();
}
```

Which modification enables the code fragment to print Happy Journey!?

- A. Replace line n1 with `public void ride() throws FuelNotAvailException {`
- B. Replace line n1 with `protected void ride() throws Exception {`
- C. Replace line n2 with `void ride() throws Exception {`
- D. Replace line n2 with `private void ride() throws FuelNotAvailException {`

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 64

Given the definition of the Emp class:

```
public class Emp
    private String eName;
    private Integer eAge;

    Emp(String eN, Integer eA) {
        this.eName = eN;
        this.eAge = eA;
```

```

    }
    public Integer getEAge () {return eAge;}
    public String getEName () {return eName;}
}

```

and code fragment:

```

List<Emp>li = Arrays.asList(new Emp("Sam", 20), New Emp("John", 60), New Emp
("Jim", 51));
Predicate<Emp> agVal = s -> s.getEAge() > 50; //line n1
li = li.stream().filter(agVal).collect(Collectors.toList());
Stream<String> names = li.stream().map.(Emp::getEName); //line n2
names.forEach(n -> System.out.print(n + " "));

```

What is the result?

- A. Sam John Jim
- B. John Jim
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 65

For which three objects must a vendor provide implementations in its JDBC driver? (Choose three.)

- A. Time
- B. Date
- C. Statement
- D. ResultSet
- E. Connection
- F. SQLException
- G. DriverManager

Correct Answer: CDE

Section: (none)

Explanation

Explanation/Reference:

Explanation:

Database vendors support JDBC through the JDBC driver interface or through the ODBC connection. Each driver must provide implementations of java.sql.Connection, java.sql.Statement, java.sql.PreparedStatement, java.sql.CallableStatement, and java.sql.ResultSet. They must also implement the java.sql.Driver interface for use by the generic java.sql.DriverManager interface.

QUESTION 66

Given the code fragment:

```

LocalDate valentinesDay =LocalDate.of(2015, Month.FEBRUARY, 14);
LocalDate nextYear = valentinesDay.plusYears(1);
nextYear.plusDays(15); //line n1
System.out.println(nextYear);

```

What is the result?

- A. 2016-02-14
- B. A `DateTimeException` is thrown.
- C. 2016-02-29
- D. A compilation error occurs at line n1.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 67

Given the code fragment:

```
BiFunction<Integer, Double, Integer> val = (t1, t2) -> t1 + t2;      //line n1
System.out.println(val.apply(10, 10.5));
```

What is the result?

- A. 20
- B. 20.5
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 68

Which statement is true about `java.time.Duration`?

- A. It tracks time zones.
- B. It preserves daylight saving time.
- C. It defines time-based values.
- D. It defines date-based values.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

Reference: <http://tutorials.jenkov.com/java-date-time/duration.html#accessing-the-time-of-a-duration>

QUESTION 69

Given the code fragment:

```
UnaryOperator<Integer> uo1 = s -> s*2;          line n1
List<Double> loanValues = Arrays.asList(1000.0, 2000.0);
```

```

loanValues.stream()
    .filter(lv -> lv >= 1500)
    .map(lv -> uol.apply(lv))
    .foreach(s -> System.out.print(s + " ")); //line 2

```

What is the result?

- A. 4000.0
- B. 4000
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 70

You have been asked to create a ResourceBundle which uses a properties file to localize an application. Which code example specifies valid keys of menu1 and menu2 with values of File Menu and View Menu?

- A. <key name = 'menu1">File Menu</key>
 <key name = 'menu2">View Menu</key>
- B. <key>menu1</key><value>File Menu</value>
 <key>menu2</key><value>View Menu</value>
- C. menu1, File Menu, menu2, View Menu Menu
- D. menu1 = File Menu
 menu2 = View Menu

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 71

Given the records from the Employee table:

eid	ename
111	Tom
112	Jerry
113	Donald

and given the code fragment:

```

try {
    Connection conn = DriverManager.getConnection (URL, userName, passWord);
    Statement st = conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
        ResultSet.CONCUR_UPDATABLE);
    st.execute("SELECT*FROM Employee");
    ResultSet rs = st.getResultSet();
    while (rs.next()) {
        if (rs.getInt(1) ==112) {
            rs.updateString(2, "Jack");
        }
    }
}

```

```

        }
    }
    rs.absolute(2);
    System.out.println(rs.getInt(1) + " " + rs.getString(2));
} catch (SQLException ex) {
    System.out.println("Exception is raised");
}

```

Assume that:

The required database driver is configured in the classpath.

The appropriate database accessible with the URL, userName, and password exists.

What is the result?

- A. The Employee table is updated with the row:

112 Jack

and the program prints:

112 Jerry

- B. The Employee table is updated with the row:

112 Jack

and the program prints:

112 Jack

- C. The Employee table is not updated and the program prints:

112 Jerry

- D. The program prints Exception is raised.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 72

Given:

```

class RateOfInterest {
    public static void main (String[] args) {
        int rateOfInterest = 0;
        String accountType = "LOAN";
        switch (accountType) {
            case "RD";
                rateOfInterest = 5;
                break;
            case "FD";
                rateOfInterest = 10;
                break;
            default:
                assert false: "No interest for this account"; //line n1
        }
        System.out.println ("Rate of interest:" + rateOfInterest);
    }
}

```

and the command:

```
java -ea RateOfInterest
```

What is the result?

- A. Rate of interest: 0

- B. An `AssertionError` is thrown.
- C. No interest for this account
- D. A compilation error occurs at line n1.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 73

Given the code fragment:

```
class CallerThread implements Callable<String> {
    String str;
    public CallerThread(String s) {this.str=s;}
    public String call() throws Exception {
        return str.concat("Call");
    }
}
```

and

```
public static void main (String[] args) throws InterruptedException,
ExecutionException
{
    ExecutorService es = Executors.newFixedThreadPool(4);           //line n1
    Future f1 = es.submit (newCallerThread("Call"));
    String str = f1.get().toString();
    System.out.println(str);
}
```

Which statement is true?

- A. The program prints `Call Call` and terminates.
- B. The program prints `Call Call` and does not terminate.
- C. A compilation error occurs at line n1.
- D. An `ExecutionException` is thrown at run time.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 74

Given the code fragment:

```
public class FileThread implements Runnable {
    String fName;
    public FileThread(String fName) { this.fName = fName; }
    public void run () System.out.println(fName);
    public static void main (String[] args) throws IOException,
InterruptedException {
        ExecutorService executor = Executors.newCachedThreadPool();
        Stream<Path> listOffiles = Files.walk(Paths.get("Java Projects"));
        listOffiles.forEach(line -> {
```

```

        executor.execute(new FileThread(line.getFileName().toString
())));
        //line n1
    });
    executor.shutdown();
    executor.awaitTermination(5, TimeUnit.DAYS);
}
line n2
}
}

```

The `Java Projects` directory exists and contains a list of files.

What is the result?

- A. The program throws a runtime exception at line n2.
- B. The program prints files names concurrently.
- C. The program prints files names sequentially.
- D. A compilation error occurs at line n1.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 75

Given:

```

class CheckClass {
    public static int checkValue (String s1, String s2) {
        return s1.length() - s2.length();
    }
}

```

and the code fragment:

```

String[] strArray = new String [] {"Tiger", "Rat", "Cat", "Lion"};
//line n1
for (String s : strArray) {
    System.out.print (s + " ");
}

```

Which code fragment should be inserted at line n1 to enable the code to print Rat Cat Lion Tiger?

- A. `Arrays.sort(strArray, CheckClass :: checkValue);`
- B. `Arrays.sort(strArray, (CheckClass :: new) :: checkValue);`
- C. `Arrays.sort(strArray, (CheckClass :: new).checkValue);`
- D. `Arrays.sort(strArray, CheckClass :: new :: checkValue);`

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 76

Given the code fragments:

```

class TechName {
    String techName;
    TechName (String techName) {
        this.techName=techName;
    }
}

```

and

```

List<TechName> tech = Arrays.asList      (
    new TechName("Java-"),
    new TechName("Oracle DB-"),
new TechName("J2EE-")
);
Stream<TechName> stre = tech.stream();
//line n1

```

Which should be inserted at line n1 to print Java-Oracle DB-J2EE-?

- A. stre.forEach(System.out::print);
- B. stre.map(a-> a.techName).forEach(System.out::print);
- C. stre.map(a-> a).forEachOrdered(System.out::print);
- D. stre.forEachOrdered(System.out::print);

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 77

Given that /green.txt and /colors/yellow.txt are accessible, and the code fragment:

```

Path source = Paths.get("/green.txt");
Path target = Paths.get("/colors/yellow.txt");
Files.move(source, target, StandardCopyOption.ATOMIC_MOVE);
Files.delete(source);

```

Which statement is true?

- A. The green.txt file content is replaced by the yellow.txt file content and the yellow.txt file is deleted.
- B. The yellow.txt file content is replaced by the green.txt file content and an exception is thrown.
- C. The file green.txt is moved to the /colors directory.
- D. A FileAlreadyExistsException is thrown at runtime.

Correct Answer: D

=> B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 78

Given:

```

interface Doable {
    public void doSomething (String s);
}

```

Which two class definitions compile? (Choose two.)

- A. public class Task implements Doable {
 public void doSomethingElse(String s) { }
- B. public abstract class Work implements Doable {
 public abstract void doSomething(String s) { }
 public void doYourThing(Boolean b) { }
- C. public abstract class Job implements Doable {
 public void doSomething(Integer i) { }
- D. public class Action implements Doable {
 public void doSomething(Integer i) { }
 public String doThis(Integer j) { }
- E. public class Do implements Doable {
 public void doSomething(Integer i) { }
 public void doSomething(String s) { }
 public void doThat (String s) { }

Correct Answer: AE

⇒ CE

⇒ Section: (none)

Explanation

Explanation/Reference:

QUESTION 79

Given the code fragment:

```

List<Integer> list1 = Arrays.asList(10, 20);
List<Integer> list2 = Arrays.asList(15, 30);
//line n1

```

Which code fragment, when inserted at line n1, prints 10 20 15 30?

- A. Stream.of(list1, list2)
 .flatMap(list -> list.stream())
 .forEach(s -> System.out.print(s + " "));
- B. Stream.of(list1, list2)
 .flatMap(list -> list.intStream())
 .forEach(s -> System.out.print(s + " "));
- C. list1.stream()
 .flatMap(list2.stream().flatMap(e1 -> e1.stream()))
 .forEach(s -> System.out.println(s + " "));
- D. Stream.of(list1, list2)
 .flatMapToInt(list -> list.stream())
 .forEach(s -> System.out.print(s + " "));

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:**QUESTION 80**

Given the code fragment:

```
public static void main (String[] args) throws IOException {
    BufferedReader brCopy = null;
    try (BufferedReader br = new BufferedReader (new FileReader("employee.txt")))
{ // line n1
    br.lines().forEach(c -> System.out.println(c));
    brCopy = br;           //line n2
}
brCopy.ready();      //line n3;
}
```

Assume that the ready method of the BufferedReader, when called on a closed BufferedReader, throws an exception, and employee.txt is accessible and contains valid text.

What is the result?

- A. A compilation error occurs at line n3.
- B. A compilation error occurs at line n1.
- C. A compilation error occurs at line n2.
- D. The code prints the content of the employee.txt file and throws an exception at line n3.

Correct Answer: D**Section: (none)****Explanation****Explanation/Reference:****QUESTION 81**

Given:

```
Book.java:
public class Book {
    public String read(String bname) { return "Read" + bname }
}

EBook.java:
public class EBook extends Book {
    public String read (String url) { return "View" + url }
}

Test.java:
public class Test {
    public static void main (String[] args) {
        Book b1 = new Book();
        b1.read("Java Programming");
        Book b2 = new EBook();
        b2.read("http://ebook.com/ebook");
    }
}
```

What is the result?

- A. Read Java Programming
View http:// ebook.com/ebook

- B. Read Java Programming
Read <http://ebook.com/ebook>
- C. The EBook.java file fails to compile.
- D. The Test.java file fails to compile.

Correct Answer: D

=> A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 82

Given the code fragment:

```
ZonedDateTime depart = ZonedDateTime.of(2015, 1, 15, 3, 0, 0, 0, ZoneID.of("UTC-7"));
ZonedDateTime arrive = ZonedDateTime.of(2015, 1, 15, 9, 0, 0, 0, ZoneID.of("UTC-5"));
long hrs = ChronoUnit.HOURS.between(depart, arrive); //line n1
System.out.println("Travel time is" + hrs + "hours");
```

What is the result?

- A. Travel time is 4 hours
- B. Travel time is 6 hours
- C. Travel time is 8 hours
- D. An exception is thrown at line n1.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 83

Given the code fragment:

```
Path path1 = Paths.get("/app/.sys/");
Path res1 = path1.resolve("log");
Path path2 = Paths.get("/server/exe/");
Path res2 = path1.resolve("/readme/");
System.out.println(res1);
System.out.println(res2);
```

What is the result?

- A. /app/sys/log
/readme/server/exe
- B. /app/log/sys
/server/exe/readme
- C. /app/.sys/log
/readme
- D. /app/.sys/log
/server/exe/readme

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 84

Given the code fragment:

```
List<String> colors = Arrays.asList("red", "green", "yellow");
Predicate<String> test = n -> {
    System.out.println("Searching...");
    return n.contains("red");
};
colors.stream()
    .filter(c -> c.length() > 3)
    .allMatch(test);
```

What is the result?

- A. Searching...
- B. Searching...
 Searching...
- C. Searching...
 Searching...
 Searching...
- D. A compilation error occurs.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 85

Given:

```
class UserException extends Exception      { }
class AgeOutOfLimitException extends UserException      { }
```

and the code fragment:

```
class App  {
    public void doRegister(String name, int age)
        throws UserException, AgeOutOfLimitException  {
        if (name.length () < 6)  {
            throw new UserException ();
        } else if (age >= 60)  {
            throw new AgeOutOfLimitException ();
        } else {
            System.out.println("User is registered.");
        }
    }
    public static void main(String[ ] args) throws UserException  {

        App t = new App ();
        t.doRegister("Mathew", 60);
    }
}
```

}

What is the result?

- A. User is registered.
- B. An AgeOutOfBoundsException is thrown.
- C. A UserException is thrown.
- D. A compilation error occurs in the main method.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 86

Given the code fragment:

```
Path path1 = Paths.get("/software/./sys/readme.txt");
Path path2 = path1.normalize();
Path path3 = path2.relativize(path1);
System.out.print(path1.getNameCount());
System.out.print(" : " + path2.getNameCount());
System.out.print(" : " + path3.getNameCount());
```

What is the result?

- A. 5 : 3 : 6
- B. 6 : 5 : 6
- C. 3 : 3 : 4
- D. 4 : 4 : 4

Correct Answer: D => A

Section: (none) **Explanation**

Explanation/Reference:

QUESTION 87

Given:

```

class Product {
    String name;
    int qty;
    public String toString(){
        return name;
    }
    public Product(String name, int qty) {
        this.name = name;
        this.qty = qty;
    }
    static class ProductFilter {
        public boolean isAvailable(Product p) { // line n1
            return p.qty >= 10;
        }
    }
}

```

and the code fragment:

```

List<Product> products = Arrays.asList(
    new Product("MotherBoard", 5),
    new Product("Speaker", 20));
products.stream()
    .filter(Product.ProductFilter::isAvailable) // line n2
    .forEach(p -> System.out.println(p));

```

Which modification enables the code fragment to print Speaker?

- A. Implement Predicate in the Product.ProductFilter class and replace line n2 with .filter (p -> p.ProductFilter.test (p))
- B. Replace line n1 with:


```
public static boolean isAvailable (Product p) {
```
- C. Replace line n2 with:


```
.filter (p -> p.ProductFilter: :isAvailable (p))
```
- D. Replace line n2 with:


```
.filter (p -> Product: :ProductFilter: :isAvailable ())
```

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 88

Given the content:

MessagesBundle.properties file:

```
username = Enter User Name  
password = Enter Password
```

MessagesBundle_fr_FR.properties file:

```
username = Entrez le nom d'utilisateur  
password = Entrez le mot de passe
```

and the code fragment:

```
Locale currentLocale = new Locale.Builder().setRegion("FR").setLanguage("fr").build();  
ResourceBundle messages = ResourceBundle.getBundle("MessagesBundle", currentLocale);  
Enumeration<String> names = messages.getKeys();  
while (names.hasMoreElements()) {  
    String key = names.nextElement();  
    String name = messages.getString(key);  
    System.out.println(key + " = " + name);  
}
```

What is the result?

- A. username = Entrez le nom d'utilisateur
password = Entrez le mot de passe
- B. username = Enter User Name
password = Enter Password
- C. A compilation error occurs.
- D. The program prints nothing.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 89

Given:

```
public class StrMan {
    public static void doStuff(String s) {
        try {
            if (s == null) {
                throw new NullPointerException();
            }
        } finally {
            System.out.println("-finally-");
        }
        System.out.println("-doStuff-");
    }
    public static void main (String[] args) {
        try {
            doStuff(null);
        } catch (NullPointerException npe) {
            System.out.println("-catch-");
        }
    }
}
```

What is the result?

- A. -catch-
-finally-
-dostuff-
- B. -catch-
- C. -finally-
-catch-
- D. -finally
-dostuff-
-catch-

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

Explanation:

Your Code ...

```
1+ public class StrMan {  
2+     public static void doStuff(String s) {  
3+         try {  
4+             if (s == null) {  
5+                 throw new NullPointerException();  
6+             }  
7+         } finally {  
8+             System.out.println("-finally-");  
9+         }  
10+        System.out.println("-doStuff-");  
11+    }  
12+    public static void main (String[] args) {  
13+        try {  
14+            doStuff(null);  
15+        } catch (NullPointerException npe) {  
16+            System.out.println("-catch-");  
17+        }  
18+    }  
19+ }
```

CommandLine Arguments ...

Stdin Inputs...

↻ Exec

Result...

CPU Time: 0.22 sec(s), Memory: 30280 kilobyte(s)

```
-finally-  
-catch-
```

QUESTION 90

Given:

```
public class Foo {
    public void methodB(String s) { System.out.println("Foo " + s); }
}

public class Bar extends Foo {
    public void methodB(String s) { System.out.println("Bar " + s); }
}

public class Baz extends Bar {
    public void methodB(String s) { System.out.println("Baz " + s); }
}

public class Daze extends Baz{
    private Bar bb = new Bar();
    public void methodB(String s) {
        bb.methodB(s);
        super.methodB(s);
    }
}

public class TestClass {
    public static void main(String[] args) {
        Baz d = new Daze();
        d.methodB("Hello");
    }
}
```

What is the result?

- A. Bar Hello
 Foo Hello
- B. Bar Hello
 Baz Hello
- C. Baz Hello
- D. A compilation error occurs in the Daze class.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 91

Given the content of the employee.txt file:

Every worker is a master.

Given that the employee.txt file is accessible and the file allemp.txt does NOT exist, and the code fragment:

```

try {
    List<String> content = Files.readAllLines(Paths.get("employee.txt"));
    content.stream().forEach(line -> {
        try {
            Files.write(
                Paths.get("allemp.txt"),
                line.getBytes(),
                StandardOpenOption.APPEND
            );
        } catch (IOException e) { System.out.println("Exception 1"); }
    });
} catch (IOException e) { System.out.println("Exception 2"); }

```

What is the result?

- A. Exception 1
- B. Exception 2
- C. The program executes, does NOT affect the system, and produces NO output.
- D. allemp.txt is created and the content of employee.txt is copied to it.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 92

Given:

```

public class Job {
    String name;
    Integer cost;
    Job(String name, Integer cost) {
        this.name = name;
        this.cost = cost;
    }
    String getName() { return name; }
    int getCost() { return cost; }
    public static void main(String[] args) {
        Job j1 = new Job("IT", null);
        DoubleSupplier js1 = j1::getCost;
        System.out.println(j1.getName() + ":" + js1.getAsDouble());
    }
}

```

What is the result?

- A. IT:null
- B. A NullPointerException is thrown at run time.
- C. A compilation error occurs.
- D. IT:0.0

Correct Answer: D => B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 93

Given the code fragment:

```
List<String> li = Arrays.asList("Java", "J2EE", "J2ME", "JSTL", "JSP", "Oracle DB");
Predicate<String> val = p -> p.contains("J");
List<String> neLi = li.stream().filter(x -> x.length() > 3)
    .filter(val).collect(Collectors.toList());
System.out.println(neLi);
```

What is the result?

- A. A compilation error occurs.
- B. [Java, J2EE, J2ME, JSTL, JSP]
- C. null
- D. [Java, J2EE, J2ME, JSTL]

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 94

Given:

```
class Product {
    String pname;
    public Product(String pname) {
        this.pname = pname;
    }
}
```

and the code fragment:

```
Product p1 = new Product("PowerCharger");
Product p2 = p1;
System.out.println(p1.equals(p2));
Product p3 = new Product("PowerCharger");
System.out.println(p1.equals(p3));
```

What is the result?

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

- D. true
- false

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 95

Given:

```
class DataConverter {  
    public void copyFlatFilesToTables() { }  
    public void close() throws Exception {  
        throw new RuntimeException(); // line n1  
    }  
}
```

and the code fragment:

```
public static void main(String[] args) throws Exception {  
    try (DataConverter dc = new DataConverter()) // line n2  
    { dc.copyFlatFilesToTables(); }  
}
```

What is the result?

- A. A compilation error occurs at line n2.
- B. A compilation error occurs because the try block doesn't have a catch or finally block.
- C. A compilation error occurs at line n1.
- D. The program compiles successfully.

Correct Answer: B

=> A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 96

Given the code fragment:

```

try {
    Properties prop = new Properties();
    prop.put("user", userName);
    prop.put("password", passWord);
    Connection conn = DriverManager.getConnection(dbURL, prop);
    if(conn != null){
        System.out.print("Connection Established");
    }
} catch (Exception e) {
    System.out.print(e);
}

```

and the information:

- The required database driver is configured in the classpath.
- The appropriate database is accessible with the dbURL, username, and password exists.

What is the result?

- A. A ClassNotFoundException is thrown at runtime.
- B. The program prints nothing.
- C. The program prints Connection Established.
- D. A SQLException is thrown at runtime.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 97

In 2015, daylight saving time in New York, USA, begins on March 8th at 2:00 AM. As a result, 2:00 AM becomes 3:00 AM.

Given the code fragment:

```

ZoneId zone = ZoneId.of("America/New_York");
ZonedDateTime dt = ZonedDateTime.of(LocalDate.of(2015, 3, 8), LocalTime.of(1, 0),
zone);
ZonedDateTime dt2 = dt.plusHours(2);
System.out.print(DateTimeFormatter.ofPattern("H:mm - ").format(dt2));
System.out.println("difference: " + ChronoUnit.HOURS.between(dt, dt2));

```

Which is the result?

- A. 3:00 - difference: 2
- B. 2:00 - difference: 1
- C. 4:00 - difference: 3
- D. 4:00 - difference: 2

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 98

Given the code fragment:

```
for (Course a : Course.values()) {
    System.out.print(a + " Fees " + a.getCost() + " ");
}
```

Which is the valid definition of the Course enum?

- A.

```
enum Course { JAVA(100), J2ME(150);
    private int cost;
    public Course(int c) {
        this.cost = c;
    }
    int getCost() {
        return cost;
    }
}
```
- B.

```
enum Course { JAVA(100), J2ME(150);
    private static int cost;
    private Course(int c) {
        this.cost = c;
    }
    static int getCost() {
        return cost;
    }
}
```
- C.

```
final enum Course { JAVA(100), J2ME(150);
    private int cost;
    public Course(int c) {
        this.cost = c;
    }
    int getCost() {
        return cost;
    }
    void setCost(int c) {
        this.cost = c;
    }
}
```

```
D. enum Course { JAVA(100), J2ME(150);
    private int cost;
    Course(int c) {
        this.cost = c;
    }
    int getCost() {
        return cost;
    }
}
```

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 99

Given:

```
class Resource implements AutoCloseable {
    public void close() throws Exception {
        System.out.print("Close-");
    }
    public void open() {
        System.out.print("Open-");
    }
}
```

and this code fragment:

```
Resource res1 = new Resource();
try {
    res1.open();
    res1.close();
} catch (Exception e) {
    System.out.println("Exception - 1");
}
try (res1 = new Resource()) { // line n1
    res1.open();
} catch (Exception e) {
    System.out.println("Exception - 2");
}
```

What is the result?

- A. Open-Close-
Exception - 1
Open-Close-
- B. Open-Close-Open-Close-
- C. A compilation error occurs at line n1.

D. Open-Close-Open-

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 100

Given the code fragment:

```
List<String> cs = Arrays.asList("Java", "Java EE", "Java ME");
// line n1
System.out.print(b);
```

Which code fragment, when inserted at line n1, ensures false is printed?

- A. boolean b = cs.stream() .findAny() .get() .equals("Java");
- B. boolean b = cs.stream() .anyMatch (w -> w.equals ("Java"));
- C. boolean b = cs.stream() .findFirst() .get() .equals("Java");
- D. boolean b = cs.stream() .allMatch(w -> w.equals("Java"));

Correct Answer: B

=> D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 101

Given the code fragment:

```
final String str1 = "Java";
StringBuffer strBuf = new StringBuffer("Course");
UnaryOperator<String> u = (str2) -> str1.concat(str2); // line n1
UnaryOperator<String> c = (str3) -> str3.toLowerCase();
System.out.println(u.apply(c.apply(strBuf))); // line n2
```

What is the result?

- A. A compilation error occurs at line n1.
- B. courseJava
- C. Javacourse
- D. A compilation error occurs at line n2.

Correct Answer: A => D

Section: (none) Explanation

Explanation/Reference:

QUESTION 102

Given:

```

class Engine {
    double fuelLevel;
    Engine(int fuelLevel) { this.fuelLevel = fuelLevel; }
    public void start() {
        // line n1
        System.out.println("Started");
    }
    public void stop() { System.out.println("Stopped"); }
}

```

Your design requires that:

- fuelLevel of Engine must be greater than zero when the start() method is invoked. The
- code must terminate if fuelLevel of Engine is less than or equal to zero.

Which code fragment should be added at line n1 to express this invariant condition?

- A. assert (fuelLevel) : "Terminating...";
- B. assert (fuelLevel > 0) : System.out.println ("Impossible fuel");
- C. assert fuelLevel < 0: System.exit(0);
- D. assert fuelLevel > 0: "Impossible fuel" ;

Correct Answer: C =>

D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 103

Given the code fragment:

```

List<Integer> li = Arrays.asList(10, 20, 30);
Function<Integer, Integer> fn = f1 -> f1 + f1;
Consumer<Integer> conVal = s -> System.out.print("Val:" + s + " ");
li.stream().map(fn).forEach(conVal);

```

What is the result?

- A. Val:20 Val:40 Val:60
- B. Val:10 Val:20 Val:30
- C. A compilation error occurs.
- D. Val: Val: Val:

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 104

Given the code fragments:

```

public static Optional<String> getCountry(String loc) {
    Optional<String> couName = Optional.empty();
    if ("Paris".equals(loc))
        couName = Optional.of("France");
    else if ("Mumbai".equals(loc))
        couName = Optional.of("India");
    return couName;
}

```

and

```

Optional<String> city1 = getCountry("Paris");
Optional<String> city2 = getCountry("Las Vegas");
System.out.println(city1.orElse("Not Found"));
if (city2.isPresent())
    city2.ifPresent(x -> System.out.println(x));
else
    System.out.println(city2.orElse("Not Found"));

```

What is the result?

- A. France
Optional[NotFound]
- B. Optional [France]
Optional [NotFound]
- C. Optional[France]
Not Found
- D. France
Not Found

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 105

Given the code fragment:

```
//line n1
System.out.println(ip);
```

Which code fragment, when inserted at line n1, enables the code to print /First.txt?

- A. Path ip = new Paths ("/First.txt");
- B. Path ip = Paths.toPath ("/First.txt");
- C. Path ip = new Path ("/First.txt");
- D. Path ip = Paths.get ("/", "First.txt");

Correct Answer: D

Section: (none)**Explanation****Explanation/Reference:****QUESTION 106**

Given the code fragment:

```
10. try {  
11.     Connection conn = DriverManager.getConnection(dbURL, userName, passWord);  
12.     String query = "SELECT * FROM Employee WHERE ID = 110";  
13.     Statement stmt = conn.createStatement();  
14.     ResultSet rs = stmt.executeQuery(query);  
15.     System.out.println("Employee ID: " + rs.getInt("ID"));  
16. } catch (Exception se) {  
17.     System.out.println("Error");  
18. }
```

Assume that:

The required database driver is configured in the classpath.

The appropriate database is accessible with the dbURL, userName, and passWord exists

The Employee table has a column ID of type integer and the SQL query matches one record.

What is the result?

- A. Compilation fails at line 14.
- B. Compilation fails at line 15.
- C. The code prints the employee ID.
- D. The code prints Error.

Correct Answer: A

⇒ D

⇒ **Section:**

(none)

Explanation

Explanation/Reference:**QUESTION 107**

```
public static void main(String[] args) {  
    Console console = System.console();  
    char[] pass = console.readPassword("Enter password:"); // line n1  
    String password = new String(pass); // line n2  
}
```

What is the result?

- A. A compilation error occurs at line n1.
- B. A compilation error occurs at line n2.
- C. The code reads the password without echoing characters on the console.
- D. A compilation error occurs because the IOException isn't declared to be thrown or caught?

Correct Answer: A => C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 108

Locale	Currency Symbol	Currency Code
US	\$	USD

and the code fragment?

```
double d = 15;
Locale l = new Locale("en", "US");
NumberFormat formatter = NumberFormat.getCurrencyInstance(l);
System.out.println(formatter.format(d));
```

What is the result?

- A. \$15.00
- B. 15 \$
- C. USD 15.00
- D. USD \$15

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 109

Given the code fragment:

```
Stream<List<String>> strs = Stream.of(
    Arrays.asList("text1", "text2"),
    Arrays.asList("text2", "text3"));
Stream<String> bs2 = strs
    .filter(b -> b.contains("text1"))
    .flatMap(rs -> rs.stream());
bs2.forEach(b -> System.out.print(b));
```

What is the result?

- A. text1text2
- B. text1text2text2text3
- C. text1
- D. [text1, text2]

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 110

Given:

```
public interface LengthValidator {  
    public boolean checkLength(String str);  
}
```

and

```
public class Txt {  
    public static void main(String[] args) {  
        boolean res = new LengthValidator() {  
            public boolean checkLength(String str) {  
                return str.length() > 5 && str.length() < 10;  
            }  
        }.checkLength("Hello");  
    }  
}
```

Which interface from the `java.util.function` package should you use to refactor the class `Txt`?

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

Correct Answer: C

=> B

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://docs.oracle.com/javase/8/docs/api/java/util/function/package-summary.html>

QUESTION 111

Given:

```
public class Product {  
    public double applyDiscount(double price) {  
        assert (price > 0); // line n1  
        return price * 0.50;  
    }  
    public static void main(String[] args) {  
        Product p = new Product();  
        double newPrice =  
            p.applyDiscount(Double.parseDouble(args[0]));  
        System.out.println("New Price: " + newPrice);  
    }  
}
```

and the command:

```
java Product 0
```

What is the result?

- A. An `AssertionError` is thrown.
- B. A compilation error occurs at line n1.
- C. New Price: 0.0
- D. A `NumberFormatException` is thrown at run time.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 112

Given the code fragment:

```
LocalTime now = LocalTime.now();
long timeToBreakfast = 0;
LocalTime office_start = LocalTime.of(7, 30);
if (office_start.isAfter(now)) {
    timeToBreakfast = now.until(office_start, MINUTES);
} else {
    timeToBreakfast = now.until(office_start, HOURS);
}
System.out.println(timeToBreakfast);
```

Assume that the value of `now` is 6:30 in the morning.

What is the result?

- A. An exception is thrown at run time.
- B. 0
- C. 60
- D. 1

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 113

Given the code fragments:

```
class R implements Runnable {  
    public void run() { System.out.println("Run..."); }  
}  
  
class C implements Callable<String> {  
    public String call() throws Exception { return "Call..."; }  
}
```

and

```
ExecutorService es = Executors.newSingleThreadExecutor();  
es.execute(new R()); // line n1  
Future<String> f1 = es.submit(new C()); // line n2  
System.out.println(f1.get());  
es.shutdown();
```

What is the result?

- A. The program prints Run... and throws an exception.
- B. A compilation error occurs at line n1.
- C. Run...
 Call...
- D. A compilation error occurs at line n2.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 114

Which two are elements of a singleton class? (Choose two.)

- A. a transient reference to point to the single instance
- B. a public method to instantiate the single instance
- C. a public static method to return a copy of the singleton reference
- D. a private constructor to the class
- E. a public reference to point to the single instance

Correct Answer: BD

Section: (none)

Explanation

Explanation/Reference:

QUESTION 115

Given the code fragment:

```
Deque<String> queue = new ArrayDeque<>();
queue.add("Susan");
queue.add("Allen");
queue.add("David");
System.out.println(queue.pop());
System.out.println(queue.remove());
System.out.println(queue);
```

What is the result?

- A. David
 David
 [Susan, Allen]
- B. Susan
 Susan
 [Susan, Allen]
- C. Susan
 Allen
 [David]
- D. David
 Allen
 [Susan]
- E. Susan
 Allen
 [Susan, David]

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

Explanation:

```
1 import java.util.ArrayDeque;
2
3 public class Program {
4     public static void main(String[] args) {
5
6     ArrayDeque<String> queue = new ArrayDeque<>();
7     queue.add("Susan");
8     queue.add("Allen");
9     queue.add("David");
10    System.out.println(queue.pop());
11    System.out.println(queue.remove());
12    System.out.println(queue);
13 }
14 }
15
16
```

CommandLine Arguments ...

Stdin Inputs...

 Execute Save

Result...

CPU Time: 0.16 sec(s), Memory: 29452 kilobyte(s)

```
Susan
Allen
[David]
```

QUESTION 116

Given:

```

public class Vehicle {
    int vId;
    String vName;
    public Vehicle(int vIdArg, String vNameArg) {
        this.vId = vIdArg;
        this.vName = vNameArg;
    }
    public int getVId() { return vId; }
    public String getVName() { return vName; }
    public String toString() {
        return vName;
    }
}

```

and the code fragment:

```

List<Vehicle> vehicle = Arrays.asList(
    new Vehicle(2, "Car"),
    new Vehicle(3, "Bike"),
    new Vehicle(1, "Truck"));
vehicle.stream()
    // line n1
    .forEach(System.out::print);

```

Which two code fragments, when inserted at line n1 independently, enable the code to print
TruckCarBike?

- A. .sorted ((v1, v2) -> v1.getVId() < v2.getVId())
- B. .sorted (Comparable.comparing (Vehicle: :getVName)).reversed ()
- C. .map (v -> v.getVid())
 .sorted ()
- D. .sorted((v1, v2) -> Integer.compare(v1.getVId(), v2.getVId()))
- E. .sorted(Comparator.comparing ((Vehicle v) -> v.getVId()))

Correct Answer: DE

Section: (none)

Explanation

Explanation/Reference:

QUESTION 117

Given the code fragment:

```

List<String> valList = Arrays.asList("", "George", "", "John", "Jim");
Long newVal = valList.stream()           // line n1
    .filter(x -> !x.isEmpty())
    .count();                           // line n2
System.out.print(newVal);

```

What is the result?

- A. A compilation error occurs at line n2.
- B. 3
- C. 2
- D. A compilation error occurs at line n1.

Correct Answer: A => B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 118

Given the code fragment:

```
// Login time:2015-01-12T21:58:18.817Z
Instant loginTime = Instant.now();
Thread.sleep(1000);

// Logout time:2015-01-12T21:58:19.880Z
Instant logoutTime = Instant.now();

loginTime = loginTime.truncatedTo(ChronoUnit.MINUTES); // line n1
logoutTime = logoutTime.truncatedTo(ChronoUnit.MINUTES);

if (logoutTime.isAfter(loginTime))
    System.out.println("Logged out at:"+logoutTime);
else
    System.out.println("Can't logout");
```

What is the result?

- A. A compilation error occurs at line n1.
- B. Logged out at: 2015-01-12T21:58:19.880Z
- C. Can't logout
- D. Logged out at: 2015-01-12T21:58:00Z

Correct Answer: D => C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 119

Given the code fragment:

```
List<String> words = Arrays.asList("win", "try", "best", "luck", "do");
Predicate<String> test1 = w -> {
    System.out.println("Checking...");
    return w.equals("do");                                // line n1
};
Predicate test2 = (String w) -> w.length() > 3; // line n2
words.stream()
    .filter(test2)
    .filter(test1)
    .count();
```

What is the result?

- A. A compilation error occurs at line n1.
- B. Checking...
- C. Checking...
 Checking...
- D. A compilation error occurs at line n2.

Correct Answer: A

=> D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 120

Assume `customers.txt` is accessible and contains multiple lines.

Which code fragment prints the contents of the `customers.txt` file?

- A. Stream<String> stream = Files.find (Paths.get ("customers.txt"));
 stream.forEach((String c) -> System.out.println(c));
- B. Stream<Path> stream = Files.find (Paths.get ("customers.txt"));
 stream.forEach(c) -> System.out.println(c));
- C. Stream<Path> stream = Files.list (Paths.get ("customers.txt"));
 stream.forEach(c) -> System.out.println(c));
- D. Stream<String> lines = Files.lines (Paths.get ("customers.txt"));
 lines.forEach(c -> System.out.println(c));

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 121

```
' class MyClass implements AutoCloseable {
    int test;
    public void close() { }
    public MyClass copyObject() { return this; }
}'
```

and the code fragment:

```
MyClass obj = null;
try (MyClass obj1 = new MyClass()) {
    obj1.test = 100;
    obj = obj1.copyOf(); // line n1
}
System.out.println(obj.test); // line n2
```

What is the result?

- A. An exception is thrown at line n2.
- B. 100
- C. A compilation error occurs because the try block is declared without a catch or finally block.
- D. A compilation error occurs at line n1.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 122

Which two methods from the `java.util.stream.Stream` interface perform a reduction operation? (Choose two.)

- A. count ()
- B. collect ()
- C. distinct ()
- D. peek ()
- E. filter ()

Correct Answer: AB

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://docs.oracle.com/javase/8/docs/api/java/util/stream/package-summary.html>

QUESTION 123

Which code fragment is required to load a JDBC 3.0 driver?

- A. Connection con = Connection.getDriver
("jdbc:xyzdata://localhost:3306/EmployeeDB");
- B. Class.forName("org.xyzdata.jdbc.NetworkDriver");
- C. Connection con = DriverManager.getConnection
("jdbc:xyzdata://localhost:3306/EmployeeDB");
- D. DriverManager.loadDriver ("org.xyzdata.jdbc.NetworkDriver");

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 124

Given:

```
public class Foo<K, V> {  
    private K key;  
    private V value;  
  
    public Foo(K key, V value) { this.key = key; this.value = value; }  
  
    public static <T> Foo<T, T> twice(T value) { return new Foo<T, T>(value, value); }  
  
    public K getKey() { return key; }  
    public V getValue() { return value; }  
}
```

Which option fails?

- A. Foo<String, Integer> mark = new Foo<String, Integer> ("Steve", 100);
- B. Foo<String, String> pair = Foo.<String>twice ("Hello World!");
- C. Foo<Object, Object> percentage = new Foo<String, Integer>("Steve", 100);
- D. Foo<String, String> grade = new Foo <> ("John", "A");

Correct Answer: A

=> C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 125

Given the code fragment.

```
List<Integer> prices = Arrays.asList(3, 4, 5);  
prices.stream()  
    .filter(e -> e > 4)  
    .peek(e -> System.out.print("Price " + e))  
    .map(n -> n - 1)  
    .peek(n -> System.out.println(" New Price " + n));
```

// line n1
// line n2
// line n3

Which modification enables the code to print Price 5 New Price 4?

- A. Replace line n2 with .map (n -> System.out.println ("New Price" + n -1)) and remove line n3
- B. Replace line n2 with .mapToInt (n -> n - 1);
- C. Replace line n1 with .forEach (e -> System.out.print ("Price" + e))
- D. Replace line n3 with .forEach (n -> System.out.println ("New Price" + n));

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 126

Given the definition of the Book class:

```
public class Book {  
    private int id;  
    private String name;  
    public Book(int id, String name) {this.id = id; this.name = name;}  
    public int getId() { return id; }  
    public String getName() { return name; }  
    public void setId(int id) { this.id = id; }  
    public void setName(String name) { this.name = name; }  
}
```

Which statement is true about the Book class?

- A. It demonstrates encapsulation.
- B. It is defined using the factory design pattern.
- C. It is defined using the singleton design pattern.
- D. It demonstrates polymorphism.
- E. It is an immutable class.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 127

Given the code fragment:

```
ProductCode<Number, Integer> c1 = new ProductCode<Number, Integer>(); /* c1  
instantiation */  
ProductCode<Number, String> c2 = new ProductCode<Number, String>(); /* c2  
instantiation */
```

You have been asked to define the ProductCode class. The definition of the ProductCode class must allow c1 instantiation to succeed and cause a compilation error on c2 instantiation.

Which definition of ProductCode meets the requirement?

- A.

```
class ProductCode<T, S<Integer>> {  
    T c1;  
    S c2;  
}
```
- B.

```
class ProductCode<T, S extends T> {  
    T c1;  
    S c2;  
}
```
- C.

```
class ProductCode<T, S> {  
    T c1;  
    S c2;  
}
```

D. `class ProductCode<T, S super T> {
 T c1;
 S c2;
}`

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 128

Given the code fragment:

```
Map<Integer, Integer> mVal = new HashMap<>();  
mVal.put(1, 10);  
mVal.put(2, 20);  
//line n1  
c.accept(1, 2);  
mVal.forEach(c);
```

Which statement can be inserted into line n1 to print 1,2; 1,10; 2,20;?

- A. BiConsumer<Integer, Integer> c = (i, j) -> {System.out.print (i + "," + j+ ");};
- B. BiFunction<Integer, Integer, String> c = (i, j) -> {System.out.print (i + "," + j+ " ; ")};
- C. BiConsumer<Integer, Integer, String> c = (i, j) -> {System.out.print (i + "," + j+ " ; ")};
- D. BiConsumer<Integer, Integer, Integer> c = (i, j) -> {System.out.print (i + "," + j+ " ; ")};

Correct Answer: B

=> A

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://www.concretewebpage.com/java/jdk-8/java-8-biconsumer-bifunction-bipredicate-example>

QUESTION 129

Given the code fragment:

```
List<String> nums = Arrays.asList("EE", "SE");  
String ans = nums  
    .parallelStream()  
    .reduce("Java ", (a, b) -> a.concat(b));  
System.out.print(ans);
```

What is the result?

- A. Java EEJava EESE
- B. Java EESE
- C. The program prints either:
Java EEJava SE

or
Java SEJava EE
D. Java EEJava SE

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 130

Given the code fragments :

```
public class Product {  
    String name;  
    Integer price;  
    Product(String name, Integer price) {  
        this.name = name;  
        this.price = price;  
    }  
    public void printVal(){ System.out.print(name + " Price:" + price + " "); }  
    public void setPrice(int price) { this.price = price; }  
    public Integer getPrice() { return price; }  
}
```

and

```
List<Product> li = Arrays.asList(new Product("TV", 1000), new Product("Refrigerator",  
2000));  
Consumer<Product> raise = e -> e.setPrice(e.getPrice() + 100);  
li.forEach(raise);  
li.stream().forEach(Product::printVal);
```

What is the result?

- A. TV Price :1100 Refrigerator Price :2100
- B. A compilation error occurs.
- C. TV Price :1000 Refrigerator Price :2000
- D. The program prints nothing.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 131

Given:

```
interface P { public void method1(); }

interface Q extends P { public void method1(); }

interface R extends P { public void method2(); }

interface S { public default void method() { } }

interface T { public void method1(); public void method2(); }

interface U { public void method1(); public abstract void method2(); }
```

Which two interfaces can you use to create lambda expressions? (Choose two.)

- A.T
- B.R
- C.P
- D.S
- E.Q
- F.U

Correct Answer: CD

=> CE

Section: (none)

Explanation

Explanation/Reference:

QUESTION 132

Given the code fragment:

```
final List<String> list = new CopyOnWriteArrayList<>();
final AtomicInteger ai = new AtomicInteger(0);
final CyclicBarrier barrier = new CyclicBarrier(2, new Runnable() {
    public void run() { System.out.println(list); }
});
Runnable r = new Runnable() {
    public void run() {
        try {
            Thread.sleep(1000 * ai.incrementAndGet());
            list.add("X");
            barrier.await();
        } catch (Exception ex) {
        }
    }
};
new Thread(r).start();
new Thread(r).start();
new Thread(r).start();
new Thread(r).start();
```

What is the result?

- A. [X]
[X, X]

- [X, X, X]
 - [X, X, X, X]
- B. [X, X]
- C. [X]
[X, X]
- D. [X, X]
[X, X, X, X]

Correct Answer: A => D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 133

Given that these files exist and are accessible:

```
/company/emp/info.txt  
/company/emp/benefits/b1.txt
```

and given the code fragment:

```
// line n1  
stream.forEach(s -> System.out.print(s));
```

Which code fragment can be inserted at line n1 to enable the code to print only /company/emp?

- A. Stream<Path> stream = Files.list (Paths.get ("/company"));
- B. Stream<Path> stream = Files.find(
 Paths.get ("/company"), 1,
 (p,b) -> b.isDirectory (), FileVisitOption.FOLLOW_LINKS);
- C. Stream<Path> stream = Files.walk (Paths.get ("/company"));
- D. Stream<Path> stream = Files.list (Paths.get ("/company/emp"));

Correct Answer: B

=> A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 134

Given:

```

class Person {
    String name;
    int age;
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public String getName(){ return name; }
    public int getAge(){ return age; }
}

```

and the code fragment:

```

List<Person> sts = Arrays.asList(
    new Person("Jack", 30),
    new Person("Mike Hill", 21),
    new Person("Thomas Hill", 24));
Stream<Person> resList = sts.stream().filter(s -> s.getAge() >= 25); // line n1
long count = resList.filter(s -> s.getName().contains("Hill")).count();
System.out.print(count);

```

What is the result?

- A. 0
- B. A compilation error occurs at line n1.
- C. An Exception is thrown at run time.
- D. 2

Correct Answer: B => A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 135

Which class definition compiles?

- A.

```
class Vehicle {
    int id;
    public void start() {
        public class Engine { int eNo = id; }
    }
}
```
- B.

```
class Computer {
    private Card sCard = new SoundCard();
    private abstract class Card { }
    private class SoundCard extends Card { }
}
```

```
C. class Block {
    int bno;
    static class Counter {
        int locator;
        Counter() { locator = bno; }
    }
}

D. class Product {
    interface Moveable { void move(); }
    Moveable mProduct = new Moveable() {
        void move() { }
    };
}
```

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 136

Given the code fragment:

```
Deque<Integer> nums = new ArrayDeque<>();
nums.add(1000);
nums.push(2000);
nums.add(3000);
nums.push(4000);
Integer i1 = nums.remove();
Integer i2 = nums.pop();
System.out.println(i1 + " : " + i2);
```

What is the result?

- A. 4000 : 2000
- B. 4000 : 1000
- C. 1000 : 4000
- D. 1000 : 2000

Correct Answer: B => A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 137

Given that `version.txt` is accessible and contains:

1234567890

and given the code fragment:

```

try (FileInputStream fis = new FileInputStream("version.txt");
     InputStreamReader isr = new InputStreamReader(fis);
     BufferedReader br = new BufferedReader(isr)) {
    if (br.markSupported()) {
        System.out.print((char) br.read());
        br.mark(2);
        System.out.print((char) br.read());
        br.reset();
        System.out.print((char) br.read());
    }
} catch (Exception e) {
    e.printStackTrace();
}

```

What is the result?

- A. 121
- B. 122
- C. 135
- D. The program prints nothing.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 138

```

7. BiPredicate<String, String> bp = (String s1, String s2) -> s1.contains("SG") &&
   s2.contains("Java");
8. BiFunction<String, String, Integer> bf = (String s1, String s2) -> {
9.     int fee = 0;
10.    if (bp.test(s1, s2)) {
11.        fee = 100;
12.    }
13.    return fee;
14. };
15. int fee1 = bf.apply("D101SG", "Java Programming");
16. System.out.println(fee1);

```

What is the result?

- A. A compilation error occurs at line 7.
- B. 100
- C. A compilation error occurs at line 8.
- D. A compilation error occurs at line 15.

Correct Answer: A => B

Section: (none) **Explanation**

Explanation/Reference:

QUESTION 139

Given the content:

`MessagesBundle.properties` file:

```
inquiry = How are you?
```

`MessagesBundle_de_DE.properties` file:

```
inquiry = Wie geht's?
```

and given the code fragment:

```
Locale currentLocale;
// line 1
 ResourceBundle messages = ResourceBundle.getBundle("MessagesBundle", currentLocale);
 System.out.println(messages.getString("inquiry"));
```

Which two code fragments, when inserted at line 1 independently, enable the code to print "Wie geht's?"

- A. `currentLocale = new Locale ("de", "DE");`
- B. `currentLocale = new Locale.Builder ().setLanguage ("de").setRegion ("DE").build();`
- C. `currentLocale = Locale.GERMAN;`
- D. `currentlocale = new Locale();
 currentLocale.setLanguage ("de");
 currentLocale.setRegion ("DE");`
- E. `currentLocale = Locale.getInstance(Locale.GERMAN, Locale.GERMANY);`

Correct Answer: BD

=> AB

Section: (none)

Explanation

Explanation/Reference:

QUESTION 140

Given the code fragment:

```
List<String> qwords = Arrays.asList("why ", "what ", "when ");
BinaryOperator<String> operator = (s1, s2) -> s1.concat(s2); // line n1
String sen = qwords.stream()
    .reduce("Word: ", operator);
System.out.println(sen);
```

What is the result?

- A. Word: why what when
- B. Word: why Word: why what Word: why what when
- C. Word: why Word: what Word: when
- D. Compilation fails at line n1.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 141

Given:

```
interface Interface1 {
    public default void sayHi() {
        System.out.println("Hi Interface-1");
    }
}

interface Interface2 {
    public default void sayHi() {
        System.out.println("Hi Interface-2");
    }
}
public class MyClass implements Interface1, Interface2 {
    public static void main(String[] args) {
        Interface1 obj = new MyClass();
        obj.sayHi();
    }
    public void sayHi() {
        System.out.println("Hi MyClass");
    }
}
```

What is the result?

- A. Hi Interface-2
- B. A compilation error occurs.
- C. Hi Interface-1
- D. Hi MyClass

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 142

Given:

```
class Block {
    String color;
    int size;
    Block(int size, String color) {
        this.size = size;
        this.color = color;
    }
}
```

and the code fragment:

```
List<Block> blocks = new ArrayList<>();
blocks.add(new Block(10, "Green"));
blocks.add(new Block(7, "Red"));
blocks.add(new Block(12, "Blue"));
Collections.sort(blocks, new ColorSorter());
```

Which definition of the `ColorSorter` class sorts the blocks list?

- A.

```
class ColorSorter implements Comparable<Block> {
    public boolean compare(Block o1, Block o2) {
        return o1.color.equals(o2.color);
    }
}
```
- B.

```
class ColorSorter implements Comparable<Block> {
    public int compareTo(Block o1, Block o2) {
        return o1.color.compareTo(o2.color);
    }
}
```
- C.

```
class ColorSorter implements Comparator<Block> {
    public int compare(Block o1, Block o2) {
        return o1.color.compareTo(o2.color);
    }
}
```
- D.

```
class ColorSorter implements Comparator<Block> {
    public boolean compare(Block o1, Block o2) {
        return o1.color.compareTo(o2.color);
    }
}
```

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 143

Given the code fragment:

```
public static void main(String[] args) {
    Stream.of("Java", "Unix", "Linux")
        .filter(s -> s.contains("n"))
        .peek(s -> System.out.println("PEEK: " + s))
    // line n1
}
```

Which two code fragments, when inserted at line n1 independently, result in the output PEEK: Unix?

- A. .anyMatch () ;
- B. .allMatch () ;
- C. .findAny () ;
- D. .noneMatch () ;
- E. .findFirst () ;

Correct Answer: CE

Section: (none)

Explanation

Explanation/Reference:

QUESTION 144

Given the code fragments:

```
class Person // line n1
{
    String name;
    Person(String name) {
        this.name = name;
    }
    // line n2
}
```

and

```
List<Person> emps = new ArrayList<>();
/* code that adds objects of the Person class to the emps list goes here */
Collections.sort(emps);
```

Which two modifications enable to sort the elements of the `emps` list? (Choose two.)

- A. Replace line n1 with
 - class Person extends Comparator<Person>
- B. At line n2 insert
 - public int compareTo (Person p) {
return this.name.compareTo (p.name);
}
- C. Replace line n1 with
 - class Person implements Comparable<Person>
- D. At line n2 insert
 - public int compare (Person p1, Person p2) {
return p1.name.compareTo (p2.name);
}
- E. At line n2 insert:
 - public int compareTo (Person p, Person p2) {
return p1.name.compareTo (p2.name);
}
- F. Replace line n1 with
 - class Person implements Comparator<Person>

Correct Answer: BC

Section: (none)

Explanation

Explanation/Reference:

QUESTION 145

Given:

```
class Person {  
    private String firstName;  
    private int salary;  
    public Person(String fn, int sal) {  
        this.firstName = fn;  
        this.salary = sal;  
    }  
    public int getSalary() { return salary; }  
    public String getFirstName() { return firstName; }  
}
```

and the code fragment:

```
List<Person> prog = Arrays.asList(  
    new Person("Smith", 1500),  
    new Person("John", 2000),  
    new Person("Joe", 1000));  
double dVal = prog.stream()  
    .filter(s -> s.getFirstName().startsWith("J"))  
    .mapToInt(Person::getSalary)  
    .average()  
    .getAsDouble();  
System.out.print(dVal);
```

What is the result?

- A. 0.0
- B. 1500.0
- C. A compilation error occurs.
- D. 2000.0

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 146

Given the code fragment:

```

Connection con = null;
try {
    // line n1
    if(con != null){
        System.out.print("Connection Established.");
    }

} catch (Exception e) {
    System.out.print(e);
}

```

Assume that dbURL, userName, and password are valid.

Which code fragment can be inserted at line n1 to enable the code to print Connection Established?

- A. Properties prop = new Properties();
 prop.put ("user", userName);
 prop.put ("password", password);
 con = DriverManager.getConnection (dbURL, prop);
- B. con = DriverManager.getConnection (userName, password, dbURL);
- C. Properties prop = new Properties();
 prop.put ("userid", userName);
 prop.put ("password", password);
 prop.put("url", dbURL);
 con = DriverManager.getConnection (prop);
- D. con = DriverManager.getConnection (dbURL);
 con.setClientInfo ("user", userName);
 con.setClientInfo ("password", password);

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 147

Given the Greetings.properties file, containing:

```

HELLO_MSG = Hello, everyone!
GOODBYE_MSG = Goodbye everyone!

```

and given:

```

import java.util.Enumeration;
import java.util.Locale;
import java.util.ResourceBundle;

public class ResourcesApp {
    public void loadResourceBundle() {
        ResourceBundle resource = ResourceBundle.getBundle("Greetings", Locale.US);
        System.out.println(resource.getObject(1));
    }
    public static void main(String[] args) {
        new ResourcesApp().loadResourceBundle();
    }
}

```

What is the result?

- A. Compilation fails.
- B. GOODBY_MSG
- C. Hello, everyone!
- D. Goodbye everyone!
- E. HELLO_MSG

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 148

Given the code fragments:

```
public class Test {  
    List<String> list = null;  
    public void printValues() {  
        System.out.print(getList());  
    }  
    public List<String> getList(){ return list; }  
    public void setList(List<String> newList){ list = newList; }  
}
```

and

```
List<String> li = Arrays.asList("Dog", "Cat", "Mouse");  
Test t = new Test();  
t.setList(li.stream().collect(Collectors.toList()));  
t.getList().forEach(Test::printValues);
```

What is the result?

- A. null
- B. A compilation error occurs.
- C. DogCatMouse
- D. [Dog, Cat, Mouse]

Correct Answer: D => B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 149

Given the records from the STUDENT table:

sid	sname	semail
111	James	james@uni.com
112	Jane	jane@uni.com
114	John	john@uni.com

Given the code fragment:

```
public static void main(String[] args) throws SQLException {
    //code to load and register valid jdbc driver go here
    Connection con = DriverManager.getConnection(URL, username, password);
    Statement st = con.createStatement	ResultSet.TYPE_SCROLL_INSENSITIVE,
                                         ResultSet.CONCUR_UPDATABLE);
    st.execute("SELECT * FROM student");
    ResultSet rs = st.getResultSet();
    rs.absolute(3);
    rs.moveToInsertRow();
    rs.updateInt(1, 113);
    rs.updateString(2, "Jannet");
    rs.updateString(3, "jannet@uni.com");
    rs.updateRow();
    rs.refreshRow();
    System.out.println(rs.getInt(1) + " : " + rs.getString(2) + " : " + rs.getString
(3));
}
```

Assume that the URL, username, and password are valid.

What is the result?

- A. The STUDENT table is not updated and the program prints:
114 : John : john@uni.com
- B. The STUDENT table is updated with the record:
113 : Jannet : jannet@uni.com
and the program prints:
114 : John : john@uni.com
- B. The STUDENT table is updated with the record:
113 : Jannet : jannet@uni.com
and the program prints:
113 : Jannet : jannet@uni.com
- D. A SQLException is thrown at run time.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 150

Given the code fragment:

```
5. IntConsumer consumer = e -> System.out.println(e);
6. Integer value = 90;
7. /* insert code fragment here */
8. consumer.accept(result);
```

Which code fragment, when inserted at line 7, enables printing 100?

- A. Function<Integer> funRef = e -> e + 10;
 Integer result = funRef.apply(value);
- B. IntFunction funRef = e -> e + 10;
 Integer result = funRef.apply (10);
- C.ToIntFunction<Integer> funRef = e -> e + 10;
 int result = funRef.applyAsInt (value);
- D.ToIntFunction funRef = e -> e + 10;
 int result = funRef.apply (value);

Correct Answer: A

=> C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 151

Which two statements are true about the Fork/Join Framework? (Choose two.)

- A. The RecursiveTask subclass is used when a task does not need to return a result.
- B. The Fork/Join framework can help you take advantage of multicore hardware.
- C. The Fork/Join framework implements a work-stealing algorithm.
- D. The Fork/Join solution when run on multicore hardware always performs faster than standard sequential solution.

Correct Answer: AC

=> BC

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://www.logicbig.com/tutorials/core-java-tutorial/java-multi-threading/fork-and-join.html>

QUESTION 152

Which two statements are true about synchronization and locks? (Choose two.)

- A. A thread automatically acquires the intrinsic lock on a synchronized statement when executed.
- B. The intrinsic lock will be retained by a thread if return from a synchronized method is caused by an uncaught exception.
- C. A thread exclusively owns the intrinsic lock of an object between the time it acquires the lock and the time it releases it.
- D. A thread automatically acquires the intrinsic lock on a synchronized method's object when entering that method.
- E. Threads cannot acquire intrinsic locks on classes.

Correct Answer: AB

=> AC

Section: (none)

Explanation

Explanation/Reference:

Reference: <https://docs.oracle.com/javase/tutorial/essential/concurrency/locksSync.html>

QUESTION 153

Given the code fragment:

```
//line n1  
Double d = str.average().getAsDouble();  
System.out.println("Average = " + d);
```

Which should be inserted into line n1 to print Average = 2.5?

- A. IntStream str = Stream.of (1, 2, 3, 4);
- B. IntStream str = IntStream.of (1, 2, 3, 4);
- C. DoubleStream str = Stream.of (1.0, 2.0, 3.0, 4.0);
- D. Stream str = Stream.of (1, 2, 3, 4);

Correct Answer: C

= B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 154

Given the structure of the Student table:

```
Student (id INTEGER, name VARCHAR)
```

Given the records from the STUDENT table:

ID	NAME
102	Edwin
103	Edward
103	Edwin

Given the code fragment:

```
Connection conn = DriverManager.getConnection(dbURL, userName, passWord);  
Statement st = conn.createStatement();  
String query = "DELETE FROM Student WHERE id = 103";  
System.out.println("Status: " + st.execute(query));
```

Assume that:

- The required database driver is configured in the classpath.
- The appropriate database is accessible with the dbURL, userName, and password exists.

What is the result?

- A. The program prints Status: true and two records are deleted from the Student table.
- B. The program prints Status: false and two records are deleted from the Student table.
- C. A SQLException is thrown at runtime.
- D. The program prints Status: false but the records from the Student table are not deleted.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 155

Given the code fragments:

```
public class Video {  
    public void play() throws IOException {  
        System.out.print("Video played.");  
    }  
}  
  
public class Game extends Video {  
    public void play() throws Exception {  
        super.play();  
        System.out.print("Game played.");  
    }  
}
```

and

```
try {  
    new Game().play();  
} catch (Exception e) {  
    System.out.print(e.getClass());  
}
```

What is the result?

- A. Video played.Game played.
- B. A compilation error occurs.
- C. class java.lang.Exception
- D. class java.io.IOException

Correct Answer: C

=> B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 156

What is true about the `java.sql.Statement` interface?

- A. It provides a session with the database.
- B. It is used to get an instance of a `Connection` object by using JDBC drivers.
- C. It provides a cursor to fetch the resulting data.
- D. It provides a class for executing SQL statements and returning the results.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

Reference: https://docs.oracle.com/cd/E13222_01/wls/docs45/classdocs/java.sql.Statement.html

QUESTION 157

Given that data.txt and alldata.txt are accessible, and the code fragment:

```
public void writeFiles() throws IOException {
    BufferedReader br = new BufferedReader(new FileReader("data.txt"));
    BufferedWriter bw = new BufferedWriter(new FileWriter("alldata.txt"));
    String line = null;
    while ((line = br.readLine()) != null) {
        bw.append(line + "\n");
    }
    // line n1
}
```

What is required at line n1 to enable the code to overwrite alldata.txt with data.txt?

- A. br.close();
- B. bw.writeln();
- C. br.flush();
- D. bw.flush();

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 158

Given:

```
class Student {
    String course, name, city;
    public Student(String name, String course, String city) {
        this.course = course; this.name = name; this.city = city;
    }
    public String toString() {
        return course + ":" + name + ":" + city;
    }
    public String getCourse() { return course; }
    public String getName() { return name; }
    public String getCity() { return city; }
}
```

and the code fragment:

```
List<Student> stds = Arrays.asList(  
    new Student ("Jessy", "Java ME", "Chicago"),  
    new Student ("Helen", "Java EE", "Houston"),  
    new Student ("Mark", "Java ME", "Chicago"));  
stds.stream()  
    .collect(Collectors.groupingBy(Student::getCourse))  
    .forEach(src, res) -> System.out.println(scr));
```

What is the result?

- A. [Java EE: Helen:Houston]
[Java ME: Jessy:Chicago, Java ME: Mark:Chicago]
- B. Java EE
Java ME
- C. [Java ME: Jessy:Chicago, Java ME: Mark:Chicago]
[Java EE: Helen:Houston]
- D. A compilation error occurs.

Correct Answer: D

=> B

Section: (none)

Explanation

Explanation/Reference:

Explanation:

Your Code ...

```
1- public class Student {  
2     String course, name, city;  
3-     public Student (String name, String course, String cit  
4         this.course = course; this.name = name; this.city  
5     }  
6-     public String toString() {  
7         return course + ":" + name + ":" + city;  
8     }  
9     public String getCourse() {return course; }  
10    public String getName() {return name; }  
11    public String getCity() {return city; }  
12  
13    List<Student> stds = Arrays.asList (  
14        new Student ("Jessy", "Java ME", "Chicago"),  
15        new Student ("Helen", "Java ME", "Houston"),  
16        new Student ("Mark", "Java ME", "Chicago"));  
17    stds.stream()  
18        .collect (Collectors.groupBy(Student::getCourse))  
19        .forEach (src, res) -> System.out.println(src);  
20    }  
21
```

CommandLine Arguments ...

Stdin Inputs...

Execute  Save 

Result...

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

```
/Student.java:17: error: <identifier> expected  
    stds.stream()  
          ^  
/Student.java:17: error: ';' expected  
    stds.stream()  
          ^  
2 errors
```

QUESTION 159

Given:

```

class Counter extends Thread {
    int i = 10;
    public synchronized void display(Counter obj) {
        try {
            Thread.sleep(5);
            obj.increment(this);
            System.out.println(i);
        } catch (InterruptedException ex) { }
    }
    public synchronized void increment (Counter obj) {
        i++;
    }
}
public class Test {
    public static void main(String[] args) {
        final Counter obj1 = new Counter();
        final Counter obj2 = new Counter();
        new Thread(new Runnable() {
            public void run() {obj1.display(obj2); }
        }).start();
        new Thread(new Runnable() {
            public void run() { obj2.display(obj1); }
        }).start();
    }
}

```

From what threading problem does the program suffer?

- A. race condition
- B. deadlock
- C. starvation
- D. livelock

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 160

Given the definition of the Employee class:

```

class Employee {
    String dept, name;
    public Employee(String d, String n) {
        dept = d;
        name = n;
    }
    public String toString() {
        return getDept() + ":" + getName();
    }
    public String getDept() { return dept; }
    public String getName() { return name; }
}

```

and this code fragment:

```

List<Employee> emps = Arrays.asList(new Employee("sales", "Ada"),
                                      new Employee("sales", "Bob"),
                                      new Employee("hr", "Bob"),
                                      new Employee("hr", "Eva"));
Stream<Employee> s = emps.stream()
    .sorted(Comparator.comparing((Employee e) -> e.getDept())
            .thenComparing((Employee e) -> e.getName()));
List<Employee> eSorted = s.collect(Collectors.toList());
System.out.println(eSorted);

```

What is the result?

- A. [sales:Ada, hr:Bob, sales:Bob, hr:Eva]
- B. [Ada:sales, Bob:sales, Bob:hr, Eva:hr]
- C. [hr:Eva, hr:Bob, sales:Bob, sales:Ada]
- D. [hr:Bob, hr:Eva, sales:Ada, sales:Bob]

Correct Answer: A

=> D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 161

Given the code fragments:

```

class ThreadRunner implements Runnable {
    public void run () { System.out.print ("Runnable") ; }
}
class ThreadCaller implements Callable {
    Public String call () throws Exception {return "Callable"; }
}

```

and

```
ExecutorService es = Executors.newCachedThreadPool ();
Runnable r1 = new ThreadRunner ();
Callable c1 = new ThreadCaller ();
// line n1
es.shutdown();
```

Which code fragment can be inserted at line n1 to start r1 and c1 threads?

- A. Future<String> f1 = (Future<String>) es.submit (r1);
es.execute (c1);
- B. es.execute (r1);
 Future<String> f1 = es.execute (c1) ;
- C. Future<String> f1 = (Future<String>) es.execute(r1);
Future<String> f2 = (Future<String>) es.execute(c1);
- D. es.submit(r1);
 Future<String> f1 = es.submit (c1);

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 162

Given the code fragment:

```
List<Double> doubles = Arrays.asList (100.12, 200.32);
DoubleFunction funD = d -> d + 100.0;
doubles.stream () . forEach (funD); // line n1
doubles.stream(). forEach(e -> System.out.println(e)); // line n2
```

What is the result?

- A. A compilation error occurs at line n2.
- B. 200.12
 300.32
- C. 100.12
 200.32
- D. A compilation error occurs at line n1.

Correct Answer: A

=> D

Section: (none)

Explanation

Explanation/Reference:

Explanation:

```

/ArraysAsListExample.java:10: error: illegal character: '\u2013'
DoubleFunction funD = d -> d + 100.0;
^
/ArraysAsListExample.java:10: error: not a statement
DoubleFunction funD = d -> d + 100.0;
^
/ArraysAsListExample.java:12: error: illegal character: '\u2013'
doubles.stream(). forEach(e -> System.out.println(e)); // line n2
^
/ArraysAsListExample.java:12: error: illegal start of expression
doubles.stream(). forEach(e -> System.out.println(e)); // line n2
^
/ArraysAsListExample.java:12: error: ';' expected
doubles.stream(). forEach(e -> System.out.println(e)); // line n2
^
/ArraysAsListExample.java:12: error: ';' expected
doubles.stream(). forEach(e -> System.out.println(e)); // line n2
^
6 errors

```

QUESTION 163

Given:

```

public class Product {
int id; int price;
public Product (int id, int price) {
this.id = id;
this.price = price;
}
Public String toString () { return id + ":" + price;}
}

```

and the code fragment:

```

List<Product> products = new ArrayList <> (Arrays.asList(new Product(1, 10),
    new Product (2, 30),
    new Product (3, 20));
Product p = products.stream().reduce(new Product (4, 0), (p1, p2) -> {
    p1.price+=p2.price;
    return new Product (p1.id, p1.price);});
products.add(p);
products.stream().parallel()
    .reduce((p1, p2) -> p1.price > p2.price ? p1 : p2)
    .ifPresent(System.out::println);

```

What is the result?

- A. 4:60
- B. 2:30
- C. 4:60
2:30
3:20
1:10
- D. 4:0
- E. The program prints nothing

Correct Answer: C

=> A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 164

Given:

```
class Student {  
    String course, name, city;  
    public Student (String name, String course, String city) {  
        this.course = course; this.name = name; this.city = city;  
    }  
    public String toString() {  
        return course + ":" + name + ":" + city;  
    }  
    public String getCourse() {return course;}  
    public String getName() {return name;}  
    public String getCity() {return city;}}
```

and the code fragment:

```
List<Student> stds = Arrays.asList(  
    new Student ("Jessy", "Java ME", "Chicago"),  
    new Student ("Helen", "Java EE", "Houston"),  
    new Student ("Mark", "Java ME", "Chicago"));  
stds.stream()  
.collect(Collectors.groupingBy(Student::getCourse))  
.forEach(src, res) -> System.out.println(scr));
```

What is the result?

- A. A compilation error occurs.
- B. Java EE
Java ME
- C. [Java EE: Helen:Houston]
[Java ME: Jessy:Chicago, Java ME: Mark:Chicago]
- D. [Java ME: Jessy:Chicago, Java ME: Mark:Chicago]
[Java EE: Helen:Houston]

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 165

Given:

```

1. class MyClass implements Runnable {
2.     public int value
3.     public void run() {
4.         while (value < 100{
5.             value++;
6.             system.out.printIn("value: " + value);
7.         }
8.     }
9. }
10. public class TestThread {
11.     public static void main(String[] args) {
12.         MyClass mc = new Thread(mc);
13.         Thread a = new Thread(mc);
14.         a.start();
15.         Thread b = new Thread(mc);
16.         b.start();
17.     }
18. }

```

What change should you make to guarantee a single order of execution (printed values 1 -100 in order)?

- A. Line 3: public synchronized void run() {
- B. Line 1: class MyClass extends Thread {
- C. Line 2: public volatile int value;
- D. Line 2: public synchronized int value;

Correct Answer: B

=> A sửa : MyClass

mc = new

MyClass()

Section: (none)

Explanation

```

class MyThread implements Runnable {
    private String src[ ] = {"A", "B", "C"};
    private int count = 0;      // line n1
    public void run() {        // line n2
        while (count < src.length) {
            System.out.print(src[count]);
        }
    }
}

```

QUESTION 166

Given:

and the code fragment:

```

MyThread mt = new MyThread();
Thread t1 = new Thread(mt);
Thread t2 = new Thread(mt);
t1.start();
t2.start();

```

The threads `t1` and `t2` execute asynchronously and possibly prints ABCA or AACB.
 You have been asked to modify the code to make the threads execute synchronously and prints ABC.

Which modification meets the requirement?

- A. start the threads `t1` and `t2` within a synchronized block.
- B. Replace line n1 with:
`private synchronized int count = 0;`
- C. Replace line n2 with:
`public synchronized void run () {`
- D. Replace line n2 with:
`volatile int count = 0;`

Correct Answer: A

=> C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 167

Given that these files exist and are accessible:

```

/sports/info.txt
/sports/cricket/players.txt
/sports/cricket/data/ODI.txt

```

and given the code fragment:

```

int maxDepth =2;
Stream<Path> paths = Files.find(Paths.get("/sports"),
    maxDepth,
    (p, a) -> p.getFileName().toString().endsWith ("txt"),
    FileVisitOption.FOLLOW_LINKS);
Long fCount = paths.count();
System.out.println(fCount);

```

Assuming that there are NO soft-link/symbolic links to any of the files in the directory structure, what is the result?

- A .1
- B .2
- C .3
- D. An Exception is thrown at runtime.

Correct Answer: D

=> B

Section: (none)

Explanation

Explanation/Reference:**QUESTION 168**

Which statement is true about the single abstract method of the `java.util.function.Predicate` interface?

- A. It accepts one argument and returns `void`.
- B. It accepts one argument and returns `boolean`.
- C. It accepts one argument and always produces a result of the same type as the argument.
- D. It accepts an argument and produces a result of any data type.

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

Reference: <https://www.javatpoint.com/java-predicate-interface>

QUESTION 169**Given:**

```
class FuelNotAvailException extends Exception {      }
class Vehicle {
    void ride() throws FuelNotAvailException {      //line n1
        System.out.println("Happy Journey!");
    }
}
class SolarVehicle extends Vehicle {
    public void ride () throws FuelNotAvailException {      //line n2
        super ride ();
    }
}
```

and the code fragment:

```
public static void main (String[] args) throws Exception {
    Vehicle v = new SolarVehicle ();
    v.ride();
}
```

Which modification enables the code fragment to print Happy Journey!?

- A. Replace line n1 with `public void ride() throws FuelNotAvailException {`
- B. Replace line n1 with `protected void ride() throws Exception {`
- C. Replace line n2 with `public void ride() throws FuelNotAvailException, Exception {`
- D. Replace line n2 with `private void ride() throws FuelNotAvailException {`

Correct Answer: C**=> A****Section: (none)****Explanation****Explanation/Reference:****QUESTION 170****Given:**

```

public class Counter {
    public static void main (String[ ] args) {
        int a = 10;
        int b = -1;
        assert (b >=1) : "Invalid Denominator";
        int c = a / b;
        System.out.println (c);
    }
}

```

What is the result of running the code with the `-da` option?

- A. -10
- B. 0
- C. An `AssertionError` is thrown.
- D. A compilation error occurs.

Correct Answer: A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 171

Given records from the Player table:

PID	PName
1	Dave
2	Jack
3	Sam

and given the code fragment:

```

try {
Connection conn = DriverManager.getConnection(URL, username, password);
    Statement st= conn.createStatement(
                    ResultSet.TYPE_SCROLL_SENSITIVE,
                    ResultSet.CONCUR_UPDATABLE);
    st.execute ("SELECT * FROM Player");
    st.setMaxRows(2);
    ResultSet rs = st.getResultSet();
    rs.absolute(3);
    while (rs.next ()) {
        System.out.println(rs.getInt(1) + " " + rs.getString(2));
    }
} catch (SQLException ex) {
    System.out.print("SQLException is thrown.");
}

```

Assume that:

The required database driver is configured in the classpath.

The appropriate database is accessible with URL, username, and password.

The SQL query is valid.

What is the result?

- A. 2 Jack

- 3 Sam
- B. The program prints nothing.
- C. 3 Sam
- D. SQLException is thrown.

Correct Answer: D

> B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 172

Given:

Item table

- ID, INTEGER: PK
- DESCRIPT, VARCHAR(100)
- PRICE, REAL
- QUANTITY< INTEGER

And given the code fragment:

```

10. try {Connection conn = DriverManager.getConnection(dbURL, username, password);
11.     String query = "Select * FROM Item WHERE ID = 110";
12.     Statement stmt = conn.createStatement();
13.     ResultSet rs = stmt.executeQuery(query);
14.     while(rs.next()) {
15.         System.out.println("ID:           " + rs.getString(1));
16.         System.out.println("Description:   " + rs.getString(2));
17.         System.out.println("Price:        " + rs.getString(3));
18.         System.out.println("Quantity:      " + rs.getString(4));
19.     }
20. } catch (SQLException se) {
21.     System.out.println("Error");
22. }
```

Assume that:

The required database driver is configured in the classpath.

The appropriate database is accessible with the dbURL, userName, and passWord exists.

The SQL query is valid.

What is the result?

- A. An exception is thrown at runtime.
- B. Compilation fails.
- C. The code prints Error.
- D. The code prints information about Item 110.

Correct Answer: B

=> D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 173

Given the code fragments:

```
interface CourseFilter extends Predicate<String> {
    public default boolean test (String str) {
        return str.contains ("Java");
    }
}
```

and

```
List<String> strs = Arrays.asList("Java", "Java EE", "Embedded Java");
Predicate<String> cf1 = s -> s.length() > 3;
Predicate cf2 = new CourseFilter() { //line n1
    public boolean test (String s) {
        return s.startsWith ("Java");
    }
};
long c = strs.stream()
    .filter(cf1)
    .filter(cf2) //line n2
    .count();
System.out.println(c);
```

What is the result?

- A. 2
- B. 3
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

Correct Answer: D

=> A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 174

Given:

```
public class Foo<K, V> {
    private K key;
    private V value;

    public Foo(K key, V value) { this.key = key; this.value = value; }

    public static <T> Foo<T, T> twice(T value) { return new Foo<T, T>(value, value); }

    public K getKey() { return key; }
    public V getValue() { return value; }
}
```

Which option fails?

- A. Foo<String, Integer> mark = new Foo<Object, Object> ("Steve", 100);
- B. Foo<String, String> pair = Foo.<String>twice ("Hello World!");
- C. Foo<Object, Object> percentage = new Foo<Object, Object>("Steve", 100);
- D. Foo<String, String> grade = new Foo <> ("John", "A");

Correct Answer: C

>=A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 175

Given the code fragment:

```
Map<Integer, String> books = new TreeMap<>();  
books.put (1007, "A");  
books.put (1002, "C");  
books.put (1003, "B");  
books.put (1003, "B");  
System.out.println (books);
```

What is the result?

- A. {1007=A, 1003=B, 1002=C}
- B. {1007=A, 1003=B, 1003=B, 1002=C}
- C. {1007=A, 1002=C, 1003=B, 1003=B}
- D. {1002=C, 1003=B, 1007=A}

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 176

Given:

```
public class Canvas implements Drawable {  
    public void draw ()    { }  
}  
  
public abstract class Board extends Canvas {    }  
  
public class Paper extends Canvas {  
    protected void draw (int color)    {    }  
}  
public class Frame extends Canvas implements Drawable {  
    public void resize ()    { }  
    abstract void open ();  
}  
public interface Drawable {  
    public abstract void draw ();  
}
```

Which statement is true?

- A. Board **does not compile**.
- B. Paper **does not compile**.
- C. Frame **does not compile**.
- D. Drawable **does not compile**.
- E. All classes compile successfully.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 177

Given the code fragment:

```
UnaryOperator<Double> uo1 = s -> s*2;           //line n1
List<Double> loanValues = Arrays.asList(1000.0, 2000.0);
loanValues.stream()
    .filter(lv -> lv >= 1500)
    .map(lv -> uo1.apply(lv))                  //line n2
    .forEach(s -> System.out.print(s + " "));
```

What is the result?

- A. 4000.0
- B. 4000
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

Correct Answer: D

=> A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 178

Given the code fragment:

```
Path path1 = Paths.get("/app/./sys/");
Path res1 = path1.resolve("log");
Path path2 = Paths.get("/server/exe/");
Path res2 = path2.resolve("/readme/");
System.out.println(res1);
System.out.println(res2);
```

What is the result?

- A. /app/sys/log
/readme/server/exe
- B. /app/log/sys
/server/exe/readme
- C. /app/./sys/log
/readme
- D. /app/./sys/log
/server/exe/readme

Correct Answer: D

=> C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 179

Given the code fragment:

```
List<String> nL = Arrays.asList("Jim", "John", "Jeff");
Function<String, String> funVal = s -> "Hello : ".concat(s);
nL.stream()
    .map(funVal)
    .forEach(s-> System.out.print (s));
```

What is the result?

- A. Hello : Jim Hello : John Hello : Jeff
- B. Jim John Jeff
- C. The program prints nothing.
- D. A compilation error occurs.

Correct Answer: C

=>

Section: (none)

Explanation

Explanation/Reference:

Explanation:

The program prints nothing because the method is concat.

QUESTION 180

Given the code fragment:

```
List<String> colors = Arrays.asList("red", "green", "yellow");
Predicate<String> test = n -> {
    System.out.println("Searching...");
    return n.contains("red");
};
colors.stream()
    .filter(c -> c.length() >= 3)
    .allMatch(test);
```

What is the result?

- A. Searching...
- B. Searching...
 - Searching...
- C. Searching...
 - Searching...
 - Searching...
 - Searching...
- D. A compilation error occurs.

Correct Answer: C

=> B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 181

Given the definition of the Emp class:

```

public class Emp
    private String eName;
    private Integer eAge;

    Emp(String eN, Integer eA)      {
        this.eName = eN;
        this.eAge = eA;
    }
    public Integer getEAge () {return eAge;}
    public String getEName () {return eName;}
}

```

and code fragment:

```

List<Emp>li = Arrays.asList(new Emp("Sam", 20), new Emp("John", 60), new Emp
("Jim", 51));
Predicate<Emp> agVal = s -> s.getEAge() <= 60;           //line n1
li = li.stream().filter(agVal).collect(Collectors.toList());
Stream<String> names = li.stream().map(Emp::getEName);   //line n2
names.forEach(n -> System.out.print(n + " "));

```

What is the result?

- A. Sam John Jim
- B. John Jim
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

Correct Answer: C

=> A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 182

Given:

```

class Book {
    int id;
    String name;
    public Book (int id, String name)  {
        this.id = id;
    }
    public boolean equals (Object obj) {           //line n1
        this.name.equals(name);
        boolean output = false;
        Book b = (Book) obj;
        if (this.id == b.id) {
            output = true;
        }
        return output;
    }
}

```

and the code fragment:

```

Book b1 = new Book (101, "Java Programming");
Book b2 = new Book (102, "Java Programming");
System.out.println (b1.equals(b2));           //line n2

```

Which statement is true?

- A. The program prints `true`.
- B. The program prints `false`.
- C. A compilation error occurs. To ensure successful compilation, replace line n1 with:
`boolean equals (Book obj) {`
- D. A compilation error occurs. To ensure successful compilation, replace line n2 with:
`System.out.println (b1.equals((Object) b2));`

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 183

Given the code fragment:

```
LocalDate valentinesDay = LocalDate.of(2015, Month.FEBRUARY, 14);
LocalDate next15days = valentinesDay.plusDays (15);
LocalDate nextYear = next15days.plusYears(1); // line n1
System.out.println(nextYear);
```

What is the result?

- A. 2016-03-01
- B. A `DateTimeException` is thrown.
- C. 2016-02-29
- D. A compilation error occurs at line n1.

Correct Answer: D

=> A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 184

Given:

```
class UserException extends Exception      { }
class AgeOutOfLimitException extends UserException      { }
```

and the code fragment:

```
class App  {
    public void doRegister(String name, int age)
        throws UserException, AgeOutOfLimitException  {
        if (name.length () <= 60)  {
            throw new UserException ();
        } else if (age > 60)  {
            throw new AgeOutOfLimitException ();
        } else {
            System.out.println("User is registered.");
        }
    }
}
```

```

public static void main(String[ ] args) throws UserException {
    App t = new App ();
    t.doRegister("Mathew", 60);
}
}

```

What is the result?

- A. User is registered.
- B. An AgeOutOfLimitException is thrown.
- C. A UserException is thrown.
- D. A compilation error occurs in the main method.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 185

Given the code fragment:

```

public class Foo {
    public static void main (String [ ] args) {
        Map<Integer, String> unsortMap = new HashMap<> ( );
        unsortMap.put (10, "z");
        unsortMap.put (5, "b");
        unsortMap.put (1, "d");
        unsortMap.put (7, "e");
        unsortMap.put (50, "j");

        Map<Integer, String> treeMap = new TreeMap <Integer, String> (new
        Comparator<Integer> ( ) {
            @Override public int compare (Integer o1, Integer o2) {return
            o2.compareTo
            (o1); } } );

        treeMap.putAll (unsortMap);

        for (Map.Entry<Integer, String> entry : treeMap.entrySet () ) {
            System.out.print (entry.getValue () + " ");
        }
    }
}

```

What is the result?

- A. A compilation error occurs.
- B. d b e z j
- C. j z e b d
- D. z b d e j

Correct Answer: D

=> C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 186

Given the code fragments:

```
class Caller implements Callable<String>      {
    String str;
    public Caller (String s) {this.str=s;}
    public String call()throws Exception { return str.concat ("Caller");}
}
class Runner implements Runnable      {
    String str;
    public Runner (String s) {this.str=s;}
    public void run () { System.out.println (str.concat ("Runner")); }
}
```

and

```
public static void main (String[] args) throws InterruptedException,
ExecutionException      {
    ExecutorService es = Executors.newFixedThreadPool(2);
    Future f1 = es.submit (new Caller ("Call"));
    Future f2 = es.submit (new Runner ("Run"));
    String str1 = (String) f1.get();           //line n1
    System.out.println(str1+ ":" + str2);
    es.shutdown();
}
```

What is the result?

- A. The program prints:

```
Run Runner
Call Caller : null
And the program does not terminate.
```

- B. The program terminates after printing:

```
Run Runner
Call Caller : Run
```

- C. A compilation error occurs at line n1.

- D. An ExecutionException is thrown at run time.

Correct Answer: D

=> A (bỏ

es.shutdown ())

Section: (none)

Explanation

Explanation/Reference:

QUESTION 187

```
class Vehicle implements Comparable<Vehicle>{
Given:
    int vno;
    String name;

    public Vehicle (int vno, String name)      {
        this.vno = vno,;
        this.name = name;
    }
    public String toString ()      {
```

```

        return vno + ":" + name;
    }
    public int compareTo(Vehicle o) {
        return this.name.compareTo(o.name);
}

```

and this code fragment:

```

Set<Vehicle> vehicles = new TreeSet <> ();
vehicles.add(new Vehicle (10123, "Ford"));
vehicles.add(new Vehicle (10124, "BMW"));
System.out.println(vehicles);

```

What is the result?

- A. [10123:Ford, 10124:BMW]
- B. [10124:BMW, 10123:Ford]
- C. A compilation error occurs.
- D. A ClassCastException is thrown at run time.

Correct Answer: A

=> B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 188

Given that course.txt is accessible and contains:

```
Course : : Java
```

and given the code fragment:

```

public static void main (String[ ] args)      {
    int i;
    char c;
    try (FileInputStream fis = new FileInputStream ("course.txt");
        InputStreamReader isr = new InputStreamReader(fis);) {
        while (!isr.close()) { //line n1
            isr.skip(2);
            i = isr.read ();
            c = (char) i;
            System.out.print(c);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

What is the result?

- A. ur :: va
- B. ueJa
- C. The program prints nothing.
- D. A compilation error occurs at line n1.

Correct Answer: C => D

Section: (none)**Explanation****Explanation/Reference:****QUESTION 189**

Given the code fragment:

```
ZonedDateTime depart = ZonedDateTime.of(2015, 1, 15, 1, 0, 0, 0, ZoneID.of("UTC-7"));
ZonedDateTime arrive = ZonedDateTime.of(2015, 1, 15, 9, 0, 0, 0, ZoneID.of("UTC-5"));
long hrs = ChronoUnit.HOURS.between(depart, arrive); //line n1
System.out.println("Travel time is" + hrs + "hours");
```

What is the result?

- A. Travel time is 4 hours
- B. Travel time is 6 hours
- C. Travel time is 8 hours
- D. An exception is thrown at line n1.

Correct Answer: C

=> B

Section: (none)**Explanation****Explanation/Reference:****QUESTION 190**

Given the code fragment:

```
Path file = Paths.get ("courses.txt");
// line n1
```

Assume the courses.txt is accessible.

Which code fragment can be inserted at line n1 to enable the code to print the content of the courses.txt file?

- A. List<String> fc = Files.list(file);
 fc.stream().forEach (s -> System.out.println(s));
- B. Stream<String> fc = Files.readAllLines (file);
 fc.forEach (s -> System.out.println(s));
- C. List<String> fc = Files.readAllLines(file);
 fc.stream().forEach (s -> System.out.println(s));
- D. Stream<String> fc = Files.list (file);
 fc.forEach (s -> System.out.println(s));

Correct Answer: C

Section: (none)**Explanation****Explanation/Reference:****QUESTION 191**

Given the code fragment:

```

Stream<Path> files = Files.list(Paths.get(System.getProperty("user.home")));
files.forEach ( fName -> {                                     //line n1
    try {
        Path aPath = fName.toAbsolutePath();           //line n2
        System.out.println(fName + ":"
            + Files.readAttributes(aPath,
Basic.File.Attributes.class).creationTime
());
    } catch (IOException ex) {
        ex.printStackTrace();
    });
}

```

What is the result?

- A. All files and directories under the `home` directory are listed along with their attributes.
- B. A compilation error occurs at line n1.
- C. The files and folders in the `home` directory are listed along with their attributes.
- D. A compilation error occurs at line n2.

Correct Answer: C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 192

Given the code fragment:

```

BiFunction<Integer, Double, Integer> val = (t1, t2) -> t1 + t2;      //line n1
    //line n2
System.out.println(val.apply(10, 10.5));

```

What is the result?

- A. 20
- B. 20.5
- C. A compilation error occurs at line n1.
- D. A compilation error occurs at line n2.

Correct Answer: D

=> C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 193

Given:

```

class Student {
    String course, name, city;
    public Student (String name, String course, String city) {
        this.course = course; this.name = name; this.city = city;
    }
    public String toString() {

```

```

        return course + ":" + name + ":" + city;
    }
    public String getCourse() {return course;}
    public String getName() {return name;}
    public String getCity() {return city;}
}

```

and the code fragment:

```

List<Student> stds = Arrays.asList(
    new Student ("Jessy", "Java ME", "Chicago"),
    new Student ("Helen", "Java EE", "Houston"),
    new Student ("Mark", "Java ME", "Chicago"));
stds.stream()
    .collect(Collectors.groupingBy(Student::getCourse))
    .forEach(src, res) -> System.out.println(res));

```

What is the result?

- A. A compilation error occurs.
- B. Java EE
Java ME
- C. [Java EE: Helen:Houston]
[Java ME: Jessy:Chicago, Java ME: Mark:Chicago]
- D. [Java ME: Jessy:Chicago, Java ME: Mark:Chicago]
[Java EE: Helen:Houston]

Correct Answer: B

=> C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 194

Given the code fragments:

```

class Employee  {
    Optional<Address> address;
    Employee (Optional<Address> address)      {
        this.address = address;
    }
    public  Optional<Address> getAddress()      {      return address;      }
}

class Address  {
    String city = "New York";
    public String getCity  {      return city:      }
    public String toString()      {
        return city;
    }
}

```

and

```

Address address = new Address;
Optional<Address> addrs1 = Optional.ofNullable (address);
Employee e1 = new Employee (addrs1);
String eAddress = (addrs1.isPresent()) ? addrs1.get().getCity() : "City Not
available";
System.out.println(eAddress);

```

What is the result?

- A. New York
- B. City Not available
- C. null
- D. A NoSuchElementException is thrown at run time.

Correct Answer: C

=> A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 195

Given the code fragment:

```
public void recDelete (String dirName) throws IOException {  
    File [ ] listOfFiles = new File (dirName) .listFiles();  
    if (listOfFiles != null && listOfFiles.length >0) {  
        for (File aFile : listOfFiles) {  
            if (!aFile.isDirectory ()) {  
                if (aFile.getName () .endsWith (".class"))  
                    aFile.delete ();  
            }  
        }  
    }  
}
```

Assume that Projects contains subdirectories that contain .class files and is passed as an argument to the recDelete () method when it is invoked.

What is the result?

- A. The method deletes all the .class files in the Projects directory and its subdirectories.
- B. The method deletes the .class files of the Projects directory only.
- C. The method executes and does not make any changes to the Projects directory.
- D. The method throws an IOException.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 196

Given the code fragment:

```
List<String> empDetails = Arrays.asList("100, Robin, HR", "200, Mary,  
AdminServices", "101, Peter, HR");  
empDetails.stream()  
    .filter(s-> s.contains("r"))  
    .sorted()  
    .forEach(System.out::println); //line n1
```

What is the result?

- A. 100, Robin, HR

- 101, Peter, HR
 B. E. A compilation error occurs at line n1.
 C. 101, Peter, HR
 200, Mary, AdminServices
 D. 100, Robin, HR
 200, Mary, AdminServices
 101, Peter, HR

Correct Answer: D

=> C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 197

Given the EMPLOYEE table:

EMP_ID	EMP_NAME
100	Tom
101	Mary
102	Peter
103	Robin

Given the code fragment:

```
try (Connection con = DriverManager.getConnection (dbURL, user, passwd);) {
    Statement stmt = con.createStatement (ResultSet.TYPE_SCROLL_SENSITIVE,
    ResultSet.CONCUR_UPDATABLE);
    ResultSet rs = stmt.executeQuery("select emp_id, emp_name from employee");
    rs.absolute(-2) ; // Line1
    rs.moveToInsertRow();
    rs.updateInt(1, 104);
    rs.updateString(2, "Michael");
    rs.insertRow();
    rs.moveToCurrentRow ();
    System.out.println ("Employee Id: " + rs.getInt(1) + ", Employee Name: " +
    rs.getString(2));
}
```

Assuming the database supports scrolling and updating, what is the result?

- A. The program throws a runtime exception at Line 1.
- B. A compilation error occurs.
- C. A new record is inserted and Employee Id: 102, Employee Name: Peter is displayed.
- D. A new record is inserted and Employee Id: 104, Employee Name: Michael is displayed.

Correct Answer: D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 198

Given the code fragment:

```
List<String> codes = Arrays.asList ("B", "C", "A", "D");
codes.parallelStream().forEach(s -> System.out.print(s));
System.out.println("");
codes.parallelStream().forEachOrdered(s1 -> System.out.print(s1));
```

What is the result?

- A. ACBD // in random order
ABCD
- B. ABCD // in random order
ABCD // in random order
- C. ABCD // in random order
BCAD
- D. A compile time error occurs.

Correct Answer: B

=> C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 199

Given the information: Emp table has 10 records.

Given the code fragment:

```
try (Connection con = DriverManager.getConnection("Connection String");
      Statement stmt = con.createStatement (ResultSet.TYPE_FORWARD_ONLY,
      ResultSet.CONCUR_READ_ONLY);
      ResultSet rs = stmt.executeQuery ("SELECT * FROM EMP")) {
    // Line n1
}
```

Which code fragment, inserted at Line n1, helps you determine the number of records in ResultSet?

- A. ResultSetMetaData rsmd = rs.getMetaData();
 int totRows = rsmd.getRowCount();
- B. int totRows=0;
 while(rs.next()) {
 totRows++;
 }
- C. rs.last();
 int totRows = rs.getRowCount();
- D. rs.last();
 int totRows = rs.getRowId(1);

Correct Answer: A

=> B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 200

Given:

```
class Bird {  
    String name;  
    Bird(String name) {  
        this.name = name;  
    }  
    public void eat () {  
        System.out.println(name + " is eating");  
    }  
}  
interface BirdInt {  
    Bird getBird(String name);  
}  
public class App{  
    public static void main(String[] args) {  
        // insert code here  
        b.eat();  
    }  
}
```

Which code fragment can be inserted to print: Peacock is eating?

- A. BirdInt b = new::Bird("Peacock");
- B. BirdInt b1 = Bird::new;
 Bird b = b1.getBird("Peacock");
- C. BirdInt b = new Bird ("Peacock");
- D. Bird b = Bird::new("Peacock");

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

Explanation:

```
12 - class Bird {  
13     String name;  
14 -    Bird(String name) {  
15        this.name = name;  
16    }  
17 -    public void eat () {  
18        System.out.println(name + " is eating");  
19    }  
20}  
21 }  
22 interface BirdInt {  
23     Bird getBird(String name);  
24 }  
25 public class App{  
26     public static void main(String[] args) {  
27         BirdInt b1 = Bird::new;  
28         Bird b = b1.getBird("Peacock");  
29         b.eat();  
30     }  
31 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

CommandLine Arguments

Result

CPU Time: 0.23 sec(s), Memory: 33296 kilobyte(s)

Peacock is eating

QUESTION 201

Given:

```
public class Emp {  
    public void calcLeave() {  
        System.out.println("12");  
    }  
}
```

and the code fragment:

```
public class TestAnn {  
    public static void main (String[] args) {  
        // insert code here  
        e.calcLeave();  
    }  
}
```

Which code fragment can be inserted to print 13?

- A.

```
Emp e = () -> {  
    public void calcLeave() {  
        System.out.println("13");  
    }  
};
```
- B.

```
Emp e = new Emp() {  
    public void calcLeave() {  
        System.out.println("13");  
    }  
};
```
- C.

```
Emp e = new Emp{ public void calcLeave() {  
    System.out.println("13");  
}  
}();
```
- D.

```
Emp e = new Emp(){static class Emp {  
    public void calcLeave() {  
        System.out.println("13");  
    }  
};
```

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 202

Given the code fragment:

```

9. Connection conn = DriverManager.getConnection(dbURL, userName, passWord);
10. String query = "SELECT id FROM Employee";
11. try (Statement stmt = conn.createStatement()) {
12.     ResultSet rs = stmt.executeQuery(query);
13.     rs = stmt.executeQuery("SELECT id FROM Customer");
14.     while (rs.next()) {
15.         //process the results
16.         System.out.println("Employee ID: " + rs.getInt("id"));
17.     }
18. } catch (Exception e) {
19.     System.out.println ("Error");
20. }

```

Assume that:

- The required database driver is configured in the classpath.
- The appropriate database is accessible with the dbURL, userName, and passWord exists.
- The Employee and Customer tables are available and each table has id column with a few records and the SQL queries are valid.

What is the result of compiling and executing this code fragment?

- A. The program prints employee IDs.
- B. The program prints customer IDs.
- C. The program prints Exception.
- D. compilation fails on line 13.

Correct Answer: C

=>B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 203

```

List<String> str = Arrays.asList ("pen", "is", "not", "a", "pencil");
Predicate<String> test = s -> {
    int i = 0;
    boolean result = s.contains ("pen");
    System.out.print((i++) + ":");
    return result;
};
str.stream()
    .filter(test)
    .findFirst()
    .ifPresent(System.out ::print);

```

What is the result?

- A. 0 : 0 : pen
- B. 0 : 1 : pen
- C. 0 : pen
- D. 0 : 1 : 2 : 3 : 4 :

E. A compilation error occurs.

Correct Answer: E

=> C

Section: (none)

Explanation

Explanation/Reference:

Explanation:

```
16 public class Emp {  
17     List<String> str = Arrays.asList ("pen", "is", "not", "a", "pencil");  
18     Predicate<String> test = s -> {  
19         int i = 0;  
20         boolean result = s.contains ("pen");  
21         System.out.print((i++) + ":");  
22         <identifier> expected  
23         ';' expected  
24     str.stream()  
25         .filter(test)  
26         .findFirst()  
27         .ifPresent(System.out::print);  
28     }  
29 }  
30 }
```

QUESTION 204

Given:

```
class UserException extends Exception      {}  
class AgeOutOfLimitException extends UserException    {}
```

and the code fragment:

```
class App {  
    public void doRegister(String name, int age)  
throws UserException, AgeOutOfLimitException {  
    if (name.length () < 5) {  
        throw new UserException ();  
    } else if (age > 60) {  
        throw AgeOutOfLimitException ();  
    } else {  
        System.out.println("User is registered.");  
    }  
}  
public static void main(String[ ] args) throws UserException {  
  
    App t = new App ();  
    t.doRegister("Mathew", 60);  
}
```

What is the result?

- A. User is registered.
- B. An AgeOutOfLimitException is thrown.

- C. A UserException is thrown.
- D. A compilation error occurs in the doRegister method.

Correct Answer: B

=> A

Section: (none)

Explanation

Explanation/Reference:

QUESTION 205

Given the definition of the Vehicle class:

```
class Vehicle {  
    String name;  
    void setName (String name) {  
        this.name = name;  
    }  
    String getName () {  
        return name;  
    }  
}
```

Which action encapsulates the Vehicle class?

- A. Make the Vehicle class private.
- B. Make the name variable public.
- C. Make the getName() and setName() method public.
- D. Make the name variable private.
- E. Make the getName() and setName() method private.

Correct Answer: B

=> D

Section: (none)

Explanation

Explanation/Reference:

QUESTION 206

Given the code fragment:

```
Path path1 = Paths.get("/app/./sys/");  
Path res1 = path1.resolve("log");  
Path path2 = Paths.get("/server/exe/");  
Path res2 = path2.resolve("/readme/");  
System.out.println(res1);  
System.out.println(res2);
```

What is the result?

- A. /app/sys/log
/readme/server/exe
- B. /app/log/sys
/server/exe/readme

- C. /app/.sys/log
/readme
- D. /app/.sys/log
/server/exe/readme

Correct Answer: B => C

Section: (none)

Explanation

Explanation/Reference:

QUESTION 207

Given the code fragment:

```
public class Foo {  
    public static void main (String [ ] args) {  
        Map<Integer, String> unsortMap = new HashMap<> ( );  
        unsortMap.put (10, "z");  
        unsortMap.put (5, "b");  
        unsortMap.put (1, "d");  
        unsortMap.put (7, "e");  
        unsortMap.put (50, "j");  
  
        Map<Integer, String> treeMap = new TreeMap <Integer, String> (new  
        Comparator<Integer> ( ) {  
            @Override public int compare (Integer o1, Integer o2) {return  
            o1.compareTo  
            (o2); } } );  
  
        treeMap.putAll (unsortMap);  
  
        for (Map.Entry<Integer, String> entry : treeMap.entrySet () ) {  
            System.out.print (entry.getValue () + " ");  
        }  
    }  
}
```

What is the result?

- A. A compilation error occurs.
- B. d b e z j
- C. j z e b d
- D. z b d e j

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference: