1z0-809.67q

Number: 1z0-809
Passing Score: 800
Time Limit: 120 min

1z0-809

Java SE 8 Programmer II

https://www.gratisexam.com/

Exam A

QUESTION 1
Given the definition of the Vehicle class:

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

and this code fragment:

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

What is the result?

A. `Velocity with new speed`
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
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.output::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:

```
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 4
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 on 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=3oYOu2XXN-&sig=uVFS0KB15BqyEgghXnnjJSUdcrE&hl=pt-BR&sa=X&ved=0ahUKEwjlsKe-n6baAhVEhZAKHeiEDTgQ6AEIMDAB#v=onepage&q=You%20want%20to%20share%20code%20among%20several%20closely%20related%20classes.&f=false

QUESTION 5
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)  {
```
LOST

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 6
Given:

```
1. abstract class Shape  {
2.     Shape ( )   { System.out.println  (“Shape”);     }
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 7
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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 8
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 9
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 10
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 11
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 12

Given the code fragment:

```
public static void main (String [ ] args) throws IOException   {
    BufferedReader br = new BufferedReader (new InputStremReader (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 13
Given the code fragment:

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

and assuming that the file `/data/december/log.txt` is accessible and contains:

```
10-Dec-2014 - Executed successfully
```

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 `FileAlreadyExistsException` is thrown at run time.

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 14
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. Unable to scan.`
C. `Scan. Unable to scan.`
D. `Scan. Unable to scan. Printer closed.`

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 15
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 16
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 (2, 30));
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
Section: (none)

Explanation

Explanation/Reference:

QUESTION 17
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, A Guide to Java Tour:3]`
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: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 18
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 19
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 20
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 21
Which statement is true about the `DriverManager` class?

A. It returns an instance of `Connection`.
B. it executes SQL statements against the database.
C. It only queries metadata of the database.
D. it is written by different vendors for their specific database.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
Explanation:
The DriverManager returns an instance of Doctrine\DBAL\Connection which is a wrapper around the underlying driver connection (which is often a PDO instance).
Reference: https://www.doctrine-project.org/projects/doctrine-dbal/en/2.8/reference/configuration.html

QUESTION 22
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. The order of the output is unpredictable.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 23
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 24
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 25
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 -> uo1.apply(lv))
    .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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 26
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 27
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: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 28
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 29
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 30
Given:

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

Which two class definitions compile? (Choose two.)

A.
```
public abstract 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 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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 31
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 32
Given the code fragment:

```
Path path1 = Paths.get("/app/./sys/");
Path res1 = path1.resolve("log");
Path path2 = Paths.get("/server/exe/");
Path res1 = 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 33
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 `AgeOutOfLimitException` 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 34
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 35
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: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 36
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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 37
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 38
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: B

Section: (none)
Explanation

Explanation/Reference:

QUESTION 39
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 40
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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 41
Given the code fragment:

```
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: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 42
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: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 43
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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 44
Given:

```
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.copyObject();  // 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: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 45
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 46
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))                    // line n1
     .map(n -> n - 1)                                              // line n2
     .peek(n -> System.out.println(" New Price " + n));            // 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: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 47
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 48
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 49
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
Section: (none)
Explanation

Explanation/Reference:
Reference: https://www.concretepage.com/java/jdk-8/java-8-biconsumer-bifunction-bipredicate-example

QUESTION 50
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 51
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 :110 Refrigerator Price :2100`
B. A compilation error occurs.
C. `TV Price :1000 Refrigerator Price :2000`
D. The program prints nothing.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 52
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: AF
Section: (none)
Explanation

Explanation/Reference:

QUESTION 53
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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 54
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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 55
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: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 56
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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 57

```
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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 58
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: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 59
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 60
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: E
Section: (none)
Explanation

Explanation/Reference:

QUESTION 61
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: CE
Section: (none)
Explanation

Explanation/Reference:

QUESTION 62
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: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 63
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`

C. 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: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 64
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
Section: (none)
Explanation

Explanation/Reference:
Reference: https://docs.oracle.com/javase/tutorial/essential/concurrency/locksync.html

QUESTION 65
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 66
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 67
Given:

```
class Counter extends Thread {
     int i = 10;
     public synchronized void display(Counter obj) {
          try {
               Thread.sleep(5);
               obj.increment(this);
               System.out.printIn(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: