1z0-809.85q

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

1z0-809

Java SE 8 Programmer II

Exam A

QUESTION 1
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 2
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 3
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 (ArithmeticException | 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 | ArithmeticException | 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 4
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 5
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 6
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 7
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 run time.
B. `2:MyPic.jpeg: MyPic.jpeg`
C. `1:Pics:/Pics/ MyPic.jpeg`
D. `2:Pics: MyPic.jpeg`

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 8
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 9
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 10
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 11
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(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: B

Section: (none)
Explanation

Explanation/Reference:

QUESTION 12
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`.

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 13
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.reserveOrder())`

D. `.map(Emp::getfName).sorted(Comparator.reserveOrder().map(Emp::getlName).reserved`

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 14
Given the code fragment:

```
Stream<Path> files = Files.walk(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 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 15
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 16
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 17
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 type 1 =
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 18
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 `setName` 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 19
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 20
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 21
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 22
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 23
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 24
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 : 0 : pen`

D. `0 : 1 : 2 : 3 : 4 :`

E. A compilation error occurs.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 25
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 26
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 27
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.

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 28
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().endWith("txt")).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>>
    Exception
```

C. A compilation error occurs at `line n1`.

D. The program prints the content of the three files.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 29
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 30
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 `static` reference to point to the single instance.
E. Implement the `Serializable` interface.

Correct Answer: BD
Section: (none)
Explanation

Explanation/Reference:

QUESTION 31
Given the code fragment:

```
9.          Connection conn = DriveManager.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
Section: (none)
Explanation
Explanation/Reference:

QUESTION 32
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
20
C. A compilation error occurs.
D. A `NumberFormatException` is thrown at run time.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 33
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 34
Given:

`Item table`

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

And given the code fragment:

```
9.          try {
10.         Connection conn = DriveManager.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.getInt("Id"));
16.         System.out.println("Description:    " + rs.getString("Descrip"));
17.         System.out.println("Price:        " + rs.getDouble("Price"));
18.         System.out.println(Quantity:        " + rs.getInt("Quantity"));
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 35
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. Replace `line n1` with `class Master extends Thread {`

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 36
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 37
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 38
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 39
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 n)  - >
System.out.println(n); animal.run(100);
Moveable.walk(20);
```
D. `Movable` cannot be used in a lambda expression.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 40
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 41
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 42
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 43
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 44
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 45
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 46
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 47
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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 48
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 49
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 50
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 51
Given:

```
Book.java:
public class Book   {
    private String read(String bname)    {  return "Read" + bname   }
} EBook.java: public class EBook extends Book   {     public class
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 Programing"); 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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 52
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 53
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 54
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 55
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 56
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 57
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 58
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 59
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 60
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;
    }
}
```

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

```
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;
    }
}
```

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

B. C.

D.

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

QUESTION 61
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 - 1Open-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 62
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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 63
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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 64
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 65
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 66
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 67
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: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 68

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 69
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 70
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 71
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 72
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
Section: (none)
Explanation

Explanation/Reference:

QUESTION 73
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 74
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 75
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 76
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: A
Section: (none)
Explanation

Explanation/Reference:
QUESTION 77
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]
[X, X, X]
```

D.
```
[X, X]
[X, X, X, X]
```

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 78
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 79
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 80
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 81
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 82
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 83
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 84
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: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 85
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
Section: (none)
Explanation

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