1z0-809.oracle

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

https://www.gratisexam.com/

Exam A

QUESTION 1
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: A
Section: (none)

Explanation

Explanation/Reference:

QUESTION 2
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: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 3
Given:

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

```
        if (this.name.equals(b name))}
            output = true;
        }
        return output;
    }
}
```

and the code fragment:

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

Which statement is true?

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

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 4
Given the content of `/resourses/Message.properties`:

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

and given the code fragment:

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

```
System.out.println(prop.getProperty("welcome3"));
```

What is the result?

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

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 5
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: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 6
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: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 7
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: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 8
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: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 9
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: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 10
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: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 11
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: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 12
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: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 13
Given:

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

    private int value;

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

Which two modifications enable the given code to compile?

A. Nest the `USCurrency` enumeration declaration within the `Coin` class.
B. Make the `USCurrency` enumeration constructor `private`.
C. Remove the `new` keyword from the instantion of `usCoin`.
D. Make the getter method of `value` as a `static` method.
E. Add the `final` keyword in the declaration of `value`.

Correct Answer: AE
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: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 15
Given the structure of the STUDENT table:

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

Given:

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

Assume that:
The required database driver is configured in the classpath.
The appropriate database is accessible with the URL, userName, and passWord exists.
The SQL query is valid.

What is the result?

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

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 16
Given the code fragments:

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

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

and

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

What is the result?

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

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 17
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 18
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: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 19
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: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 20
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: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 21
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: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 22
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: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 23
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, Beginning with Java:2]`
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 24
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: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 25
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 26
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: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 27
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: E
Section: (none)
Explanation

Explanation/Reference:

QUESTION 28
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: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 29
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 30
Which statement is true about the single abstract method of the `java.util.function.Function` interface?

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

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Reference: http://winterbe.com/posts/2014/03/16/java-8-tutorial/ (functions)

QUESTION 31
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: http://doctrine-dbal.readthedocs.org/en/latest/reference/configuration.html

QUESTION 32
Given the code fragment:

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

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

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

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 33
Given:

```
public final class IceCream {
    public void prepare()   {}
}
public class Cake {
    public final void bake(int min, int temp)  {}
```

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

Which statement is true?

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

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 34
Which two statements are true about localizing an application?

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

Correct Answer: AE
Section: (none)
Explanation

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

QUESTION 35
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 36
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: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 37
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`?

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: AC
Section: (none)
Explanation

Explanation/Reference:

QUESTION 38
You want to create a singleton class by using the Singleton design pattern.
Which two statements enforce the singleton nature of the design?

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: AB
Section: (none)
Explanation

Explanation/Reference:

QUESTION 39
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: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 40
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
C. A compilation error occurs.
D. A `NumberFormatException` is thrown at run time.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 41
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: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 42
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: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 43
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: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 44
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 45

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 46
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: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 47
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: B
Section: (none)

Explanation

Explanation/Reference:

QUESTION 48
Which two code blocks correctly initialize a Locale variable?

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 49
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 50
Given the definition of the `Emp` class:

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

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

and code fragment:

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

What is the result?

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

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 51
For which three objects must a vendor provide implementations in its JDBC driver?

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

Correct Answer: CDE
Section: (none)
Explanation

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

QUESTION 52
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: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 53
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 54
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: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 55
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: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 56
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: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 57
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 58
Given:

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

Which two class definitions compile?

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: CD
Section: (none)
Explanation

Explanation/Reference:

QUESTION 59
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: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 60
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: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 61
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: D
Section: (none)
Explanation

Explanation/Reference: