

DUMP 1

Pg No 1 – 15 :

Q1)

Which two can be considered good practices for serializing Java objects?

A) Implement serialization for long-term data storage.

B) Always override the readObject/writeObject methods from the java.io.Serializable interface.

C) Assign null value by default while serializing and deserializing a transient variable.

D) Ensure that the class definition used is the same as the class definition used by Java runtime at the time when the object was serialized.

E) Implement secure serialization by generating secure object hash or using encryption.

Q2)

Given TripleThis.java:

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

Compiling TripleThis.java gives this compiler warning:

Note: TripleThis.java uses unchecked or unsafe operations.

Which two replacements remove this compiler warning and prints 12?

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

Q3)

Which two are valid statements?

- A) BiPredicate test = (final var x, y) -> (x.equals(y));** 
- B) BiPredicate test = (final Integer x, var y) -> (x.equals(y));
- C) BiPredicate test = (Integer x, final Integer y) -> (x.equals(y));**
- D) BiPredicate test = (var x, final var y) -> (x.equals(y));**
- E) BiPredicate test = (Integer x, final var y) -> (x.equals(y));

Q4)

```
public class Resource {
```

```
    public Worker owner;
```

```
}
```

```
public class Worker {  
    private boolean ready = true;  
    public synchronized boolean isReady () {  
        return ready;  
    }  
    public synchronized void work (Worker other, Resource resource) {  
        while (ready) {  
            while (resource.owner != this) {  
                try{  
                    wait(10);  
                }  
                catch (InterruptedException e) {}  
            }  
            if (other.isReady()) {  
                resource.owner = other;  
            }  
            else {  
                // do work with resource  
                ready false;  
                resource.owner = other;  
            }  
        }  
    }  
}
```

And given this fragment:

```
Worker wl = new Worker();
Worker w2 = new Worker();
Resource r = new Resource();
resource.owner = w1;
new Thread( () -> {
    wl.work(w2, r);
}).start();
new Thread( () -> {
    w2.work(wl, r);
} ).start();
```

Which describes the fragment?

- A)It is subject to livelock.
- B)It is subject to deadlock.
- C)The code does not compile.
- D)It throws an IllegalMonitorStateException.

Q5)

Automobile.java

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

```
}
```

Car.java

```
public class Car extends Automobile {  
  
    void wheels(int i) {      // line 2  
        System.out.print(4);    // line 3  
    }  
  
    public static void main(String[] args) {  
        Automobile ob new Car(); // line 4  
        ob.wheels();  
    }  
}
```

What must you do so that the code prints 4?

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



Q6)

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

```
public String toString() {  
    return name;  
}  
}
```

and

```
public class Tester {  
    static Person p = null;  
  
    public static void main(String[] args) {  
        p = checkPerson(p);  
        System.out.println(p);  
        Person p1 = new Person("Joe");  
        p1 = checkPerson(p);  
        System.out.println(p1);  
    }  
  
    public static Person checkPerson (Person p) {  
        if (p == null) {  
            p = new Person("Mary");  
        }  
        return p;  
    }  
}
```

What is the result?

A) Mary

Mary

B) Joe

Joe

C) Marry

Joe

D) null

null

Q7)

```
public class ExSuper extends Exception {  
    private final int eCode;  
  
    public ExSuper (int eCode, Throwable cause) {  
        super (cause);  
        this.eCode = eCode;  
    }  
  
    public ExSuper (int eCode, String msg, Throwable cause) {  
        super (msg, cause);  
        this.eCode = eCode;  
    }  
  
    public String getMessage() {  
        return this.eCode+": "+super.getMessage()+"-"+this.getCause().getMessage();  
    }  
}  
  
public class ExSub extends ExSuper {
```

```
public ExSub (int eCode, String msg, Throwable cause)
{ super(eCode, msg, cause); }

}
```

and the code fragment:

```
try {
    String param1 = "Oracle";
    if (param1.equalsIgnoreCase("oracle")) {
        throw new ExSub (9001, "APPLICATION ERROR-9001", new
FileNotFoundException ("MyFile.txt"));
    }
    throw new ExSuper (9001, new FileNotFoundException("MyFile.txt"));
} catch (ExSuper ex) {
    System.out.println(ex.getMessage());
}
```

What is the result?

- A)9001: APPLICATION ERROR-9001-MyFile.txt
- B)Compilations fails at Line 1.
- C)9001: java.io.FileNotFoundException: MyFile.txt-MyFile.txt
- D)9001: APPLICATION ERROR-9001-MyFile.txt
- E)9001; java.io.FileNotFoundException: MyFile.txt-MyFile.txt

Q8)

```
public class Person {
    private String name;
    private Person child;
```

```
public Person (String name, Person child) {  
    this (name);  
    this.child = child;  
}  
  
public Person(String name) {  
    this.name = name;  
}  
  
public String toString() {  
    return name+" "+child;  
}  
}
```

and

```
public class Tester {  
    public static Person createPeople() {  
        Person jane = new Person("Jane");  
        Person john = new Person("John", jane);  
        return jane;  
    }  
  
    public static Person createPerson (Person person) {  
        person = new Person("Jack", person);  
        return person;  
    }  
  
    public static void main(String[] args) {  
        Person person = createPeople();
```

```
/* line 1 */  
person = createPerson (person);  
/* line 2 */  
String name = person.toString();  
System.out.println(name);  
}  
}
```

Which statement is true?

- A)The memory allocated for Jane object can be reused in line 2.
- B)The memory allocated for Jack object can be reused in line 2.
- C)The memory allocated for John object can be reused in line 1. X
- D)The memory allocated for Jane object can be reused in line 1. ✓



Q9)

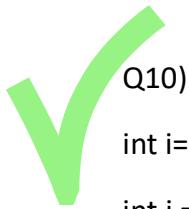
```
public class Employee {  
  
    private String name;  
  
    private String neighborhood;  
  
    private int salary;  
  
    // Constructors and setter and getter methods go here  
  
}
```

and the code fragment:

```
List roster new ArrayList<>();  
Predicate p e-> e.getSalary() > 30;  
Function<?> f = e-> Optional.ofNullable(e.getNeighborhood());
```

Which two Map objects group all employees with a salary greater than 30 by neighborhood?

- A)Map<Optional<String>, List<Employee>> r5 = roster.stream()
.collect (Collectors.groupingBy (Employee::getNeighborhood, Collectors.filtering (p,
Collectors.toList())));
- B)Map<String, List<Employee>> r2 = roster.stream().filter(p) .collect
(Collectors.groupingBy(f, Employee::getNeighborhood));
- C)Map<String, List<Employee>> r1 = roster.stream().collect (Collectors.groupingBy
(Employee::getNeighborhood, Collectors.filtering(p, Collectors.toList())));
- D)Map<Optional<String>, List<Employee>> r4 = roster stream().collect
(Collectors.groupingByf, Collectors filteringip. Collectors.toList()));
- E)Map<Optional<String>, List<Employee>> r3 = roster.stream().filter(p).collect
(Collectors.groupingBy(p));



```
int i= 3;  
int j = 25;  
  
System.out.println(i > 2?i > 10 ?i (j+10):1*j +5:1);
```

What is the result?

- A)80
- B)The compilation fails.
- C)3
- D)385
- E)2

Page 16-31

1) Which three initialization is correct

a) short sh = (short)'A';

b)float x = 1f;

c) int x = 12_34;

d)byte b = 10;

char c = b;

e) boolean false = (4 != 4);

f)int[] [][] e = ((1,1,1), (2,2,2));

g)String contact# = "(+2) (999) (232)";//a,b,c



2) Given:

```
public class Tester {  
    public static void main(String[] args){  
        float x = 2, y = 4,z = 4;  
        float a = y / x, b = y/z;  
  
        if(a > b) {  
  
            System.out.println(a + b);  
  
        }  
  
    }  
}
```

What is the result?

a)2.0

b)An exception is thrown at runtime.

c)1.0

d)The program prints nothing.

e)3.0

3)

```
public class DNASynth {  
    int aCount;  
    int tCount;  
    int cCount;  
    int gCount;  
    void setACount (int cCount) {  
        cCount cCount;  
    }  
    void setTCount() {  
        this.tCount = tCount;  
    }  
    int setCCount() {  
        return cCount;
```

```
}

int setGCount (int g) {
    gCount = g;
    return gCount;
}

void setAllCounts (int x) {
    aCount = tCount = this.cCount = setGCount(x);
}

}
```

Which two methods modify field values

- a)setAcount
- b)setTCount
- c)setGCount
- d)setCCount
- e)~~setGCount~~ setAllCount

4)

Given the code fragment:

```
public class FileHandler{

    public static void main(String[] args) {

        try (FileInputStream in = new FileInputStream("foo.txt")) {}

        catch (FileNotFoundException e) {}

    }
}
```

}

(FileInputStream will throw
FileNotFoundException but
AutoCloseable will throw IOException)

Which two actions, independently, enable the code to compile?"

a)Replacing the catch block with:

```
catch (FileNotFoundException | Exception e) {}  
finally {in.close(); }
```

b)Replacing the catch block with:

```
catch (Exception | IOException e) {}
```

c)Replacing the catch block with:

```
catch (Exception e) {}
```

d)Adding throws FileNotFoundException declaration at the main () method

e)Adding throws IOException declaration at the main () method

f)Inserting:

```
finally { in.close(); }
```

5)

```
class MyType<T> {  
    private T value;  
  
    public T getValue() {  
        return value;  
    }  
  
    public void setValue(T value) {  
        this.value = value;  
    }  
}
```

and

```
public class Test {  
    public static void main(String... args) {  
        MyType<String> strType = new MyType<>();  
        MyType<? extends Number> type = new MyType<>();  
        strType.setValue("test");  
        type.setValue(null);  
        System.out.println(strType.getValue() + ":" + type.getValue());  
    }  
}
```

What is the result?

- a)The compilation fails.
- b)An Exception is thrown at runtime.

This declares a MyType object where the type parameter T is an unknown type that extends Number (e.g., Integer, Double, etc.).

- c)test:0
- d)test:null**
- e)null:null

However, when using wildcards like ? extends Number, the setValue(T value) method is restricted because the exact type of T is unknown. The only allowed value is null since null is compatible with any type.

6)

```
public interface ExampleInterface {  
    static String origin "Interface";  
    void exampleMethod (String first);  
}  
  
public abstract class ExampleAbstractClass{  
    static String origin "Abstract Class";  
    abstract void exampleMethod(String first, String second) ;  
}  
  
public class ExampleClass extends ExampleAbstractClass implements  
ExampleInterface  
{  
    public void exampleMethod (String first) {}  
    public void exampleMethod (String first, String second){}  
    public static void main(String[] args) {  
        ExampleInterface theInstance = new ExampleClass();  
        //line n1  
        }  
    }
```

Which two, when inserted at line n1 independently, will cause a compilation error?

- a)theInstance.exampleMethod(origin).**
- b)theInstance. exampleMethod(ExampleAbstractClass.origin);

c)((ExampleClass) theInstance).exampleMethod("Japan", "Mexico");

d)the

instance.exampleMethod(ExampleAbstractclass.origin,ExampleInterface.origin);

e)theInstance.exampleMethod("France");

7)Given:

```
public enum Status {  
    BRONZE (5), SILVER(10), GOLD(15);  
  
    private int rate;  
  
    private Status (int rate) {  
        this.rate = rate;  
    }  
  
    public int getRate() { return rate; }  
  
    public Status addStatus (int rate) {  
        return new Status (20);  
    }  
}
```

and

```
public class Test {  
    public static void main(String[] args) {  
        Status silver = Status.SILVER;  
        System.out.println(silver+silver.getRate());  
        Status platinum = Status.addStatus (20);  
        System.out.println(platinum+platinum.getRate());  
    }  
}
```

}

}

What is the result?

a) SILVER10

platinum20

b) An exception is thrown at runtime.

c) The compilation fails.

d) SILVER10

PLATINUM20

e) SILVER10

20

8)

✓
public interface A {
 public Iterable a();
}

public interface B extends A {
 public Collection a();
}

public interface C extends A {
 public Path a();
}

public interface D extends B, C {
}

Java allows method overriding with covariant return types, meaning a subclass or implementing class can declare a more specific return type (e.g., a subclass of the original return type).

However, in this case:

A declares a() with a return type of Iterable.

B overrides a() with a return type of Collection (a subtype of Iterable).

C overrides a() with a return type of Path (which is not related to Iterable or Collection).

Since Collection and Path are unrelated types, the return types of a () in B and C are incompatible. This causes a compilation error.

Why does D cause a compilation error?

a) D does not define any method.

b) D extends more than one interface.

c) D inherits a () from B and c but the return types are incompatible.

d) D inherits a () only from c.

9)

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

and

.

```
public abstract class B {  
    public abstract void z();  
}
```

.

and

```
public class C extends B implements A {  
    /* insert code here */  
}
```

What code inserted into class C would allow it to compile?

a) void x() {}

```
public void y() {}  
public void z() {}
```

b) void x() {}

```
public void z() { }
```

c) void x() { super.y(); }

```
public void z() { }
```

This attempts to call super.y() inside the x() method.

However, `y()` is a default method from `A`, and default methods in interfaces cannot be called with `super` in this context.

This will result in a compilation error.

d) public void x() { }

```
protected void y() { super.y(); }
```

```
public void z() {}
```

e)public void x() ()

```
public void z() { }
```

This has a syntax error in `public void x() {}`. Methods in a concrete class cannot have abstract method declarations (without a body). This will result in a compilation error.



10)

Given the data of the EMP table:

ID NAME DEPT

101SMITH HR

102JONES KNG

103WEAVERAR

Assuming that `jdbcURL`, `username`, and `password` are declared and initialised.

```
try (Connection conn = DriverManager.getConnection (jdbcURL, username,  
password);
```

```
PreparedStatement query = conn.prepareStatement ("SELECT ID, NAME FROM  
EMP where
```

```
?");  
  
PreparedStatement update = conn.prepareStatement ("INSERT INTO RECRUITING(  
VALUES (?, ?)")) (  
  
query.setString(1, "HR");  
  
ResultSet rs = query.executeQuery()  
  
while (rs.next()) {  
  
update.setObject (1, rs.getObject (1, Integer.class), JDBCType.INTEGER),  
update.setObject (2, rs.getObject (2, String.class), JDBCType. VARCHAR)  
  
update.execute();  
  
}  

```

Which two happen upon execution?

- a)Memory leaks because Connection, PreparedStatements, and Resultset are not closed.
- b)Three SQL statements are executed.
- c)A SQLException is thrown because the Resultset is not closed.
- d)Three PreparedStatement objects are created.
- e)Two SQL statements are executed.
- f)Two PreparedStatement objects are created.

11)

Given the content from the courses.txt file:

123: Java:1

124:MySQL:2

125: Java Server Pages: 3

Given the code fragment:

```
Path filePath = Paths.get("course.txt");

try {
    /* line 1 */

} catch (IOException ex) {
    System.out.format("File 10 Exception is thrown.", ex);
}
```

Which code fragment at line 1 prints the lines that contain Java from the course.

a)List<String> lines2 = Files.readAllLines(filePath).filter(s->
s.contains("Java"));
for (String line : lines2) {
System.out.println(line);
}

b)List<String> lines1 =
Files.readAllLines(filePath).contains("Java");
for (String line : lines2) {
System.out.println(line); }

c)System.out.println(Files.readString(filePath).contains ("Java"));

d)Files.lines(filePath).filter(s -> s.contains("Java")).forEach (System.out::println);

e)Files.lines (filePath).map(s->
s.contains("Java")).forEach(System.out::println);

12)

Which two commands are used to identify class and module dependencies

a)java --show-module-resolution

b)jar --show-module-resolution

c)java Hello.java

d)jmod describe

e)jdeps --list-deps

Page 32-47

QUES 12.

Which code fragment does a service use to load the service provider with a Print Interface?

1. private java.util.ServiceLoader<Print> loader = ServiceLoader.load(Print.class);
2. private Print print = com.service.Provider.getInstance();
3. private Print print = new com.service.Provider.PrintImpl();
4. private java.util.ServiceLoader<Print> loader = new java.util.ServiceLoader();

**QUES 5 :**

123:Java:1
124:MySQL:2
125:Java Server Pages:3

Given the code fragment:

```
Path filePath = Paths.get("course.txt");
try {
    /* line 1 */
} catch (IOException ex) {
    System.out.format("File IOException is thrown.", ex);
}
```

Which code fragment at line 1 prints the lines that contain "Java" from the course?

A. List<String> lines2 = Files.readAllLines(filePath).filter(s ->
s.contains("Java"));

```
for (String line : lines2) {
    System.out.println(line);
}
```

B. List<String> lines1 = Files.readAllLines(filePath).contains("Java");

```
for (String line : lines2) {
    System.out.println(line);
}
```

C. System.out.println(Files.readString(filePath).contains("Java"));

D. Files.lines(filePath).filter(s -> s.contains("Java")).forEach(System.out::println);

E. Files.lines(filePath).map(s -> s.contains("Java")).forEach(System.out::print);

Ans :- D - Correct, this reads the file line by line as a stream, filters lines containing "Java", and prints each matching line.

A - Incorrect , the Files.readAllLines() method returns a List<String>, and filter() is not a method of the List interface, so this code won't compile.

B - Incorrect, the contains() method checks if "Java" exists in the entire list, not within each line. This isn't applicable to the whole file read as a list. Also, lines2 is not declared.

C – Incorrect, this checks if "Java" is anywhere in the entire file and prints true or false, not the lines themselves.



Ques-14

```
class Separators {  
    public static String separator = "/";  
    public static String pathSeparator = ":"; }
```

To secure this code, you want to make sure that the client code cannot modify the values. Which code will accomplish this?

A. abstract class Separators {
 public static String separator = "/";
 public static String pathSeparator = ":";
}

B. class Separators {
 private static String separator = "/";
 private static String pathSeparator = ":";
}

```
C. interface Separators {  
    String separator = "/";  
    String pathSeparator = ":";  
}
```

```
D. enum Separators {  
    separator,  
    pathSeparator; }
```

Ques – 13

Which two expressions create a valid Java Path instance?

- A. Paths.get(URI.create("file:///domains/oracle/test.txt"))
- B. Paths.get("foo")
- C. new Path("foo")
- D. Path.get(new URI ("file:///domains/oracle/test.txt"))
- E. Paths.getPath("too")

Ans:- A & B (Correct)

C (Incorrect) , Path is an interface, and you cannot instantiate an interface directly using new.

D (Incorrect), The correct method to create a Path from a URI is Paths.get(URI uri), not Path.get.

E (Incorrect) , There is no method getPath in the Paths class. The correct method is Paths.get.

Ques- 15

Given the code fragment from Box.java:

```
Public class Box implements Serializable {  
    private int boxId;  
    private String size;  
    private List items ;  
}
```

Given the code fragment from Item.java:

```
Public class Item {  
    private int id;  
    private String name;  
}
```

The classes Box and Item are encapsulated with getter and setter methods.

The classes Box and Item contain required constructors source code.

And the code fragment :

```
public static void main(String[] args) throws IOException {  
    List items1 = new ArrayList<>();  
    items1.add(new Item(1, "Pen"));  
    items1.add(new Item(2, "Ruler"));  
    Box b1 = new Box(123, "s", items1);  
    try ( FileOutputStream fout = new FileOutputStream("boxser.txt");  
        ObjectOutputStream out = new ObjectOutputStream(fout); ) {  
        out.writeObject(b1);  
        out.flush();  
        out.close();  
    } catch (Exception e) {  
        System.out.println("Unable to Serialize");  
    }
```

}

Which action serializes the b1 object?

- A. Override readObject() and writeObject() methods in the Book class.
- B. Add serialVersionUID to the Box and Item class.
- C. Handle NotSerializableException in the try clause or throw in the main() method declaration.
- D. **Implement the Serializable interface in the Item class.**
- E. Remove out.flush() method invocation.

Ans - D , to serialize an object, the class must implement the Serializable interface.

This indicates that objects of the class can be converted into a byte stream for storage or transmission. In the provided code, Box contains Item objects, so both Box and Item should implement Serializable.

Ques – 16

Which code fragment represent a valid Comparator implementation?

- 
- A. public class Comps implements Comparator {
 public boolean compare(Object obj1, Object obj2) {
 return obj1.equals(obj2);
 }
}
 - B. new Comparator() {
 public int compareTo(String str1, String str2) {
 return str1.compareTo(str2);
 }
}
 - C. public class Comps implements Comparator {
 public int compare(String str1, String str2) {

```

        return str1.length() - str2.length();

    }

}

D. new Comparator() {

    public int compare(String str1, String str2) {

        return str1.compareTo(str2);

    }

}

```

Ans – D (Correct), this fragment correctly implements the compare() method from the Comparator interface, comparing two strings based on their natural ordering.

Ques – 19

Given :

```

public class Option {

    public static void main(String[] args) {

        System.out.println("Ans: " + convert("a").get());

    }

    private static Optional convert(String s) {

        try {

            return Optional.of(Integer.parseInt(s));

        } catch (Exception e) {

            return Optional.empty();

        }

    }

}

```

What is the result?

- A. Ans :
- B. The compilation fails.
- C. Ans : a
- D. A `java.util.NoSuchElementException` is thrown at runtime.

Ans : D (Correct) , The `convert("a")` method attempts to parse the string "a" into an integer, which throws a `NumberFormatException` because "a" is not a valid integer. This exception is caught, and `Optional.empty()` is returned. Calling `.get()` on an empty `Optional` result in a `java.util.NoSuchElementException` at runtime.

Ques – 18



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

What is the result?

- A. hat at store 4
- B. hat at store 1
- C. at once 0
- D. at once 1
- E. An `IndexOutOfBoundsException` is thrown at the runtime.

Ques-21

```
class Super {  
    final int num; //line 1
```



```

public Super (int num) {
    this.num = num;
}

final void method() {
    System.out.println("Output from Super");
} }

class Sub extends Super {
    int num; //line 2
    Sub(short num) { //line 3
        super(num);
    }
    protected void method() //line 4 {
        System.out.println("Output from Sub");
    }
}

```

Which line of the code results in a compilation error?

- A. line n1
- B. line n2
- C. line n3
- D. line n4**

Ans- D (Correct) , Final method cannot be overridden.

 **Ques 22 .**

```

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

```

```
}
```

And

```
public class Bar extends Foo {  
    public void foo(List arg) {  
        System.out.println("Hello world!");  
    }  
    public static void main(String... args) {  
        List<String> li = new ArrayList<>();  
        Collection<String> co = li;  
        Bar b = new Bar();  
        b.foo(li);  
        b.foo(co);  
    }  
}
```

What is the output?

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

 Ques 20

```

public class Test {

    public static void main (String... args) {

        int number = 20 ;

        Predicate<Integer> p = a -> a%2 != 0;

        //line 1

        System.out.println(number + " is odd. ");

    } else {

        System.out.println(number + "is even. ");

    }

}

}

```

Which statement on line 1 enables the Test class to compile?

- A. If(p.accept (number)) {
- B. If(p.apply (number)) {
- C. If(p.get (number)) {
- D. If(p.test (number)) {

Ques 23 –

```

public interface Builder {

    public A build (String str);

}

```

And

```

public class BuilderImpl implements Builder {

    @Override

    public B build(String str) {

        return new B(str);
    }
}

```

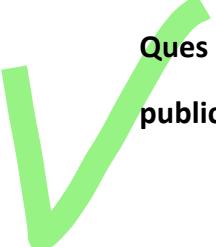
Not sure , please cross-check.

```
    }  
}  
  
}
```

If this code complies correctly, which three statements are true?

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

Ques 17-



```
public class StrBldr {  
    static StringBuilder sb1 = new StringBuilder("yo ");  
    static StringBuilder sb2 = new StringBuilder("hi ");  
  
    public static void main(String[] args) {  
        sb1 = sb1.append(new StrBldr().foo(new StringBuilder("hey")));  
        System.out.println(sb1);  
    }  
}
```

```
StringBuilder foo(StringBuilder s) {  
    sb2 = sb2.append(s+ " oh ");  
    return sb2;  
}
```

}

What is the result ?

- A. hey oh yo hi
- B. yo hi
- C. hey oh hi yo
- D. A compile time error occurs.
- E. yo hi hey oh

 **Ques 25 .**

```
package test.t1;

public class A {

    public int x = 42;

    protected A() {}          //line1

}
```

And

```
package test.t2;

import test.t1.*;

public class B extends A {

    int x = 17;              //line2

    public B() { super(); }   //line3

}
```

And

```
Package test;

import test.t1.*;
import test.t2.*;

public class Tester {
```

```
public static void main(String[] args) {  
    A obj = new B();           // line 4  
    System.out.println(obj.x); //line 5  
}  
}
```

What is the result?

- A. The compilation fails due to an error in line 3
- B. 17
- C. The compilation fails due to an error in line 2
- D. The compilation fails due to an error in line 4
- E. The compilation fails due to an error in line 1
- F. 42
- G. The compilation fails due to an error in line 5

DUMPS 01

(Page 48 - 63)

24. Given :

```
List original = new ArrayList<>(Arrays.asList(1,2,3,4,5));
```

Which two code fragments remove the elements from the original list?

Options:

- 1) Queue clq = new ConcurrentLinkedQueue<>(original);
for(Integer w : clq)
clq.remove(w);
- 2) List s1 = Collections.synchronizedList(original);

```
for(Integer w : s1)
```

```
s1.remove(w);
```

3) List a1 = new ArrayList<>(original);

```
for(Integer w : a1)
```

```
a1.remove(w);
```

4) List cwa = new CopyOnWriteArrayList<>(original);

```
for(Integer w : cwa)
```

```
cwa.remove(w);
```

 27. A company has an existing Java app that includes two Java 8 jar files , sales-8.10.jar and clients-10.2.jar.

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

They have decided to modularize clients-10.2.jar.

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

Options :

1) module com.company.clients {
 exports com.company.clients.Client ;
}

2) module com.company.clients {
 requires com.company.clients;
}

3) module com.company.clients {
 uses com.company.clients;
}

4) module com.company.clients {
 exports com.company.clients;

}



26. Given :

```
package a;  
abstract class A {  
    void print() {  
        System.out.print("Base Class");  
    }  
}
```

And

```
package a;  
public class B extends A {  
    protected void print() {  
        System.out.print("Derived Class");  
    }  
    public static void main(String args[]) {  
        B b = new B();  
        ((A)b).print();  
    }  
}
```

What is the output ?

- 1) The compilation fails.
- 2) An exception is thrown at runtime.
- 3) Derived Class**
- 4) Base Class



29. Given the code fragment:

```
Locale l = new Locale("en", "US");

LocalDate today = LocalDate.of(2018, 12, 17);

String mToday =
today.format(DateTimeFormatter.ofLocalizedDate(FormatStyle.MEDIUM));

String sToday =
today.format(DateTimeFormatter.ofLocalizedDate(FormatStyle.SHORT));

System.out.println(mToday);

System.out.println(sToday);
```

What is the output?

- 1) December 17, 2018
12/17/18
- 2) Dec 17, 2018
12/17/18
- 3) Friday, December 17, 2018
December 17, 2018
- 4) 12/17/18
Dec 17, 2018



24. Given:

```
public class Option {

    public static void main(String[] args) {

        System.out.println("Ans : "+ convert("a").get());
```

```
}

private static Optional convert(String s){

    try {

        return Optional.of(Integer.parseInt(s));

    } catch (Exception e) {

        return Optional.empty();

    }

}

}
```

Options :

- 1) A **java.util.NoSuchElementException** is thrown at runtime.
- 2) Ans : a
- 3) Ans :
- 4) The compilation fails

26. Given the code fragment :

```
module citizen {

    exports com.name to greeting;

}
```

And

```
module greeting {
```

Which statement is true ?

- 1) Public members in the com.name package are accessible to all modules.

- 2) All members in the com.name package are accessible only to the greeting module.
- 3) Inserting “requires citizen” at greeting’s module-info.java , enables com.name members accessible to the greeting module. ~~com.name members accessible to the greeting module.~~
- 4) All members of com.name are accessible only to the citizen and greeting module.
- 5) Public members in the com.name package are accessible only to the greeting module.

30. Which declaration of an annotation type is legal?

Options :

- 1) `@interface Author {
 String name() default "";
 String date();
}`
- 2) `@interface Author {
 String name() default null;
 String date();
}` This is invalid because annotation element default values cannot be null
- 3) `@interface Author {
 String name();
 String date default "";
}`
- 4) `@interface Author {
 String name();
 String date;
}`
- 5) `@interface Author extends Serializable{
 String name() default "";
 String date();
}` This is invalid because annotations in Java cannot extend interfaces or other classes. Annotations implicitly extend java.lang.annotation.Annotation.

29. Given the code fragment :

9. `Integer[] ints = {1,2,3,4,5,6,7};`

```
10. var list = Arrays.asList(ints);  
11. UnaryOperator<Integer> uo = x -> x * 3;  
12. list.replaceAll(uo);
```

Options :

- 1) UnaryOperator<Integer> uo = var x -> {return x * 3; };
- 2) UnaryOperator<Integer> uo = x -> { return x * 3 };
- 3) UnaryOperator<Integer> uo = (int x) -> x * 3;
- 4) **UnaryOperator<Integer> uo = (var x) -> (x * 3);**

 27. Which two can be considered good practices for serializing Java objects?

Options :

- 1) Assign null value by default while serializing and deserializing a transient variable.
- 2) **Ensure that the class definition used is the same as the class definition used by Java runtime the time when the object was serialized.**
- 3) **Implement secure serialization by generating secure object hash or using encryption.**
- 4) Always override the readObject/writeObject methods from the java.io.Serializable interface.
- 5) Implement serialization for long term data storage.

 43. Given:

```
import java.util.function.BiFunction;  
  
public class Pair {  
    final BiFunction validator;  
    T left = null;  
    T right = null;  
  
    private Pair() {  
        validator = null;
```

```

}

Pair(BiFunction v, T x, T y) {
    validator = v;
    set( x, y);
}

void set(T x, T y) {
    if ( !validator.apply(x, y)) throw new IllegalArgumentException
    setLeft(x);
    setRight(y);
}

void setLeft(T x) {
    left = x;
}

void setRight(T y) {
    right = y;
}

final boolean isValid() {
    return validator.apply(left , right);
}
}

```

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

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

Options:

- 1) left,right,setLeft and setRight must be private.
- 2) setLeft and setRight must be protected.
- 3) isValid must be public

4) left and right must be private.

45. Given:

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

And the code fragment:

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

You want to examine the items list if it contains an item for which the variable count is below 0?

Options:

1) if(items.stream().allMatch(i -> i.count < 0)) {

This checks if all elements in the list have count values below 0.

2) if(items.stream().filter(i -> i.count < 0).findAny()) { findAny() returns an Optional<Item>, not a boolean. This would result in a compilation error.
3) if(items.stream().anyMatch(i -> i.count < 0)) {
4) if(items.stream().filter(i -> i.count < 0).findFirst()) {

findFirst() retrieves the first matching element. However, like findAny(), it returns an Optional<Item>, which is not compatible with a boolean condition. This would also result in a compilation error.

```
String s1 = new String("Java");
String s2 = s1.intern();
StringBuilder sb1 = new StringBuilder("Java");
String s3 = sb1.toString();
System.out.println(s1 == s2);
System.out.println(s1.equals(sb1.toString()));
System.out.println(s2 == s3);
```

What is result?

- A. False true True
 - B. True False True
 - c. False True False
 - d. False False False
-

5.

Given the content from the courses.txt file:

123:Java:1
124:MYSQL:2
125:Java Server Pages: 3

Given the code fragment:

```
Path filePath = Paths.get("Course.text");
try {
```

```
/*line 1*/  
}  
catch (IOException ex) {  
    System.out.format("File IO Exception is thrown.", ex);  
}
```

Which code fragment at line 1 prints the lines that contain java from the course.txt file

```
List<String> lines2 = Files.readAllLines(filePath).filter(s->  
    s.contains("Java"));  
for(String line: lines2) {  
    System.out.println(line);  
}
```

```
List<String> lines1 = Files.readAllLines(filePath).contains("Java");  
for(String line: lines2) {  
    System.out.println(line);  
}
```

```
System.out.println(Files.readString(filePath.contains("Java")));
```

```
Files.lines(filePath).filter(s->  
    s.contains("Java")).forEach(System.out::println);
```

```
Files.lines(filePath).map(s->  
    s.contains("Java")).forEach(System.out::println);
```



49.

```
public interface ExampleInterface {  
  
    int one = 1;  
  
    static int two = 2;  
  
    static final int three = 3;  
  
}
```

```
public class ExampleClass implements ExampleInterface{
```

```
    public static void main(String[] args) {  
  
        ExampleInterface theInstance = new ExampleClass();  
  
        //line 1  
  
    }
```

Which three statements cause compile error when inserted at line 1?

```
int d = ExampleInterface.one;  
  
int b = two;  
  
int c = three;  
  
int f = ExampleInterface.three;  
  
int h = theInstance.two;
```

```
int i = theInstance.three++;
int e = ExampleInterface.two++;
int a = one++;
int g= theInstance.one;
```

35.

Which three initialization statements are correct?

boolean false = (4 != 4);

short sh = (short)'A';

int x = 12_34;

int[][][] e = ((1,1,1), (2,2,2));

String contact# = "(+2) (999) (232)";

float x = 1f;

byte b = 10;

char c = b;

1.

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

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

What does the transitive modifier mean?

Any module that requires the java.xml module does not need to require the java.xml

Any module that attempts to require the java.se module actually requires the java.xml module instead

Any module that requires the java.se module does not need to require the java.xml.

Only a module that requires the java.se module is permitted to require the java.xml

25.

Which two statements are correct about modules in java?

By default, modules can access each other as long as they run in the same folder.

java.base exports all of the Java platform's core packages.

module-info.java cannot be empty.

A module must be declared in module-info.java file.

module-info.java can be placed in any folder inside module-path.

3.

public class TripleThis {
 public static void main(String[] args) {
 Function tripler = x -> { return (Integer) x + 3; };
 TripleThis.printValue(tripler, 4);
 }
 public static void printValue(Function f, T num) {
 System.out.println(f.apply(num));
 }
}

Compiling TripleThis.java gives this compiler warning:

Note: TripleThis.java uses unchecked or unsafe Operations.

Which two replacements remove this compiler warning and prints 12?

Replace line 12 with public static void printValue(Function f, T num) {

Replace line 9 with Function tripler = x -> {return x*3;|

Replace line 9 with Function tripler = x-> {return x*3;}|

Replace line 12 with public static void printValue(Function f, Integer num) {

Replace line 12 with public static void printValue(Function f, int num) {

Replace line 9 with Function tripler = x-> {return (Integer) x*3; }

2.

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

You want to examine whether path is a directory.

Which code inserted on line 1 will accomplish this?

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

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

BasicFileAttribute attributes = Files.isDirectory(path);

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

5.

Assuming the user credentials are correct, which expression will be correct?

DriverManager.getConection("jdbc.derby.com")

DriverManager.getConnection()

DriverManager.getConection("jdbc.derby:com")

DriverManager.getConection("http://database.jdbc.com", "J_Smith", "dt12%2f3")

DriverManager.getConection("J_Smith", "dt12%2f3");

IntStream.range(1,4)

.peek(System.out::print)

.peek(i-> {

if(i == 3)

throw new RuntimeException("Exception");

});

What is the result?

The program prints: 123 and the RuntimeException is thrown.

The program prints nothing

The program prints: 12 and the RuntimeException is thrown.

The program prints 1234 and a java.lang.IllegalStateException is thrown.

4.

```
char[] characters = new char[100];

try(FileReader reader = new FileReader("file_to_path");) {
    //line1

    System.out.println(String.valueOf(characters));

}

catch(IOException e) {
    e.printStackTrace();

}
```

You want to read data through the reader object.

Which statement inserted on line 1 will accomplish this?

reader.read(characters);

reader.readLine(); *The FileReader class does not have a readLine() method.*

characters = reader.read(); *The read() method of FileReader returns an int (representing a single character or -1 for the end of the file), not a char[]. This will result in a compilation error.*

characters.read(); *characters is a char[] (array of characters) and does not have a read() method.*

8.

var h = new HashMap();



```
String[] k = {"1", "2", null, "3"};
String[] v = {"a", "b", "c", null};

for (int i = 0; i < 4; i++) {
    h.put(k[i], v[i]);
    System.out.print(h.get(k[i]) + " ");
}
```

What is result?

a b c

a b c followed by an exception

a b c null

a b followed by an exception.

10.

int i = 10;

do {

```
    for(int j = i/2; j>0; j--) {
        System.out.print(j + " ");
    }
    i-=2;
}
```

```
while(i>0);
```

What is result?

5

5 4 3 2 1

nothing

5 4 3 2 1 4 3 2 1 3 2 1 2 1 1

9.

public class mcqs {

 public static void main(String[] args) {

```
        var list = new ArrayList(
```

```
            List.of("Coffee", "Cappuccino", "Latte"));
```

```
        list.forEach((item) -> {;
```

```
            list.remove(item);
```

```
        });
```

```
        System.out.println(list);
```

```
    }
```

```
}
```

What is result?

It prints null

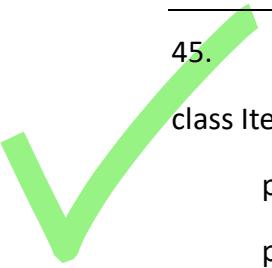
A java.util.ConcurrentModificationException is thrown

It prints[]

[Coffee, Cappuccino, Latte]

A java.lang.NullPointerException is thrown.

45.



```
class Item {  
    public String name;  
    public int count;  
    public Item(String name, int count) {  
        this.name = name;  
        this.count = count;  
    }  
}  
public class mcqs {  
    public static void main(String[] args) {  
        var items = List.of(new Item("A", 10), new Item("B", 2), new Item("C", 12), new  
Item("D", 5), new Item("E", 6));  
        //line 1  
        System.out.println("There is an item for which the variable count is  
below 0");  
    }  
}
```

}

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

Which code fragment at line 1 will accomplish this?

```
If(items.stream().allMatch(i->i.count <0)) {  
    if(items.stream().filter(i-> i.count <0).findany()) {  
        if(items.stream().anyMatch(i->i.count <0)) {  
            if(items.stream().filter(i-> i.count <0).findFirst()) {
```

 **Page number 80:**

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

And

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

And

```
package test;  
import test.t1.*;  
import test.t2.*;
```

```
public class Tester {  
    public static void main(String[] args) {  
        A obj=new B();  
        System.out.println(obj.x);  
    }  
}
```

Answer: 42

Page Number 81:

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

Which three classes successfully override showFirst()?

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

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

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

```
}
```

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

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

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

Page number:82

Why would you choose to use a peek operation instead of a forEach operation on a Stream

To process the current item and return void.

To remove an item from the end of the stream.

To remove an item from the beginning of the stream.

To process the current item and return a stream.

Page number:83

Given

```
public interface ExampleInterface {
```

```
|  
|     int one = 1;  
|  
|     static int two = 2;  
|  
|     static final int three = 3;  
|  
}|  
  
public class ExampleClass implements ExampleInterface {  
  
    public static void main(String[] args) {  
  
        ExampleInterface theInstance = new ExampleClass();  
  
        //line 1  
  
    }  
  
}
```

Which three statements cause a compiler error when inserted at line 1?

```
int d = ExampleInterface.one;  
  
int b = two;  
  
int c = three;  
  
int f = ExampleInterface.three++;  
  
int h = theInstance.two;  
  
int i = theInstance.three++;  
  
int e = ExampleInterface.two++;  
  
int a = one++;
```

```
int g = theInstance.one;
```

PageNumber:84

class Super{

```
    final int num; //line n1  
  
    public Super(int num) {  
  
        this.num=num;
```



```

        }

    final void method() {
        System.out.println("Output from Super");
    }

}

class Sub extends Super{
    int num; //line n2;
    Sub(short num){ //line n3
        super(num);
    }
    protected void method() {//line n4
        System.out.println("Output from Sub");
    }
}

```

Which line of code results in a compilation error?

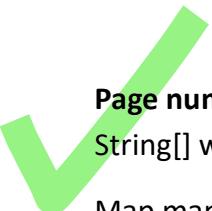
line n1

line n2

line n3

line n4

Page number:85



```

String[] words= {"am","am","first","second","mismatch"};
Map map=Arrays.stream(words).collect(Collectors
    .groupingBy(x->x,Collectors.counting()));
System.out.println(map);
}

```

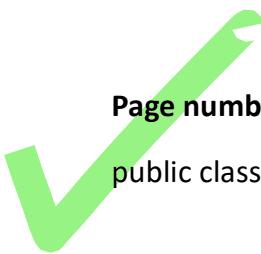
Taking into account that the order of the elements is unpredictable, what is the output?

(mismatch=1, am=2, first=1, second=1)

(mismatch=2, am=2, first=1, second=1)

(am=2, first=1, mismatch=1, second=2)

(1=mismatch, 2=am)

 Page number:86

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

```
    }  
}
```

What is the output?

Bonjour le monde!
Hello world!

Hello world!

Hello world!

Hello world!

Bonjour le monde!

Bonjour le monde!

Bonjour le monde!

Bonjour le monde!

Page number:87



```
public class Tester {  
    public static int reduce(int x) {  
        int y=4;  
        class Computer{  
            int reduce(int x) {  
                return x-y--;  
            }  
        }  
        Computer a=new Computer();  
        return a.reduce(x);  
    }  
}
```

```
    }

    public static void main(String[] args) {
        System.out.println(reduce(1));
    }

}
```

What is the result?

An exception is thrown at runtime.

-2

The compilation fails

-3

Page number:88

 int i=3;

int j=25;

System.out.println(i>2?i>10?i*(j+10):i*j+5:i);

What is the result?

3

80

The compilation fails.

385

 25

Page number:89

```
class MyType<T>{
    private T value;
    public T getValue() {
```

```
        return value;  
    }  
  
    public void setValue(T value) {  
        this.value=value;  
    }  
}
```

and

```
public class Test{
```

```
    public static void main(String[] args) {  
  
        MyType<String> strType=new MyType<>();  
  
        MyType<Integer> intType=new MyType<>();  
  
        MyType<?> type=intType;  
  
        strType.setValue("test");  
  
        type.setValue(1234);
```

```
        System.out.println(strType.getValue()+":"+type.getValue());  
    }
```

```
}
```

MyType<?> is a wildcard generic type that allows the type variable to reference any MyType object, regardless of its specific type parameter. However, with wildcard generics (?), the setValue method is not allowed to be called with any value except null. This restriction ensures type safety because the specific type of T is unknown.

What is the result?

The compilation fails.

null:null

test:null

A ClassCastException is thrown at runtime

test:1

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

And

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

And

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

Class A has a public int x = 42.

What is the result?

42

Class B has its own field int x = 17, but it does not override x from A because it is not declared with public or protected.

17

When accessing x via an A reference (obj), the x from A is used. This is called variable shadowing. The variable x in B is hidden when the object is referred to as an instance of A.

The compilation fails due to an error in line1.

The compilation fails due to an error in line3.

The compilation fails due to an error in line5.

The compilation fails due to an error in line4.

The compilation fails due to an error in line2.

Page number:91

```
Stream<Intefer> data=IntStream.range(1,10000).boxed();
```

```
Integer sum=data.mapToInt(a ->a).sum(); //line 1
```

Which two code fragments, independently, replace line 1 to implement the equivalent reduce operation?

```
Integer sum=data.mapToInt(a ->a).reduce(0,(a,b)->a+b);
```

```
OptionalInt value=data.mapToInt(a -> a).parallel().reduce(0,(a,b)->a+b);
```

```
Integer sum=value.getAsInt();
```

```
Integer sum=data.map(a ->a).reduce((a,b)->a+b);
```

```
OptionalInt value=data.mapToInt(a ->a).parallel().reduce((a,b)->a+b);
```

```
Integer sum=value.getAsInt();
```

```
int s=0;
```



```
Integer sum=data.map(a ->a).reduce(0,(a->a+s));
```

Page number:92

```
public class ConsoleTest {  
  
    public static void main(String[] args) {  
  
        Console console=System.console();  
  
        var name=console.readLine("Input Name: ");  
  
        var password=console.readPassword("Input Password: ");  
  
        System.out.println("Name is: "+name+" Password is:  
"+String.valueOf(password));  
  
    }  
}
```

and the command:

```
java ConsoleTest
```

The user will input Duke and Java when the input is prompted.

What is the output?

Input Name: Duke

Input Password:

Name is: Duke, Password is: Java

Input Name: Duke

Input Password: Java

Name is: Duke, Password is: [C@4f6ee6e4

Input Name: Duke

Input Password: Java

Name is: Duke, Password is: Java

Input Name: Duke

Input Password:

Name is:, Password is:

 **Page number:93**

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

Which is true?

The compilation fails due to an error in line 4.

The compilation fails due to an error in line 2.

The compilation fails due to an error in line 9.

The compilation fails due to an error in line 7.

The compilation fails due to an error in line 6.

The compilation fails due to an error in line 10.

The compilation succeeds.

Page number:94

```
var fruits=List.of("apple","orange","banana","lemon");
Optional<String> result= fruits.stream().filter(f->f.contains("n")).findAny(); //line 1
System.out.println(result.get());
```

You replace the code on line1 to use ParallelStream.

Which one is correct?

The compilation fails.

A NoSuchElementException is thrown at runtime.

The code will produce the same result.

The code may produce a different result.

Page number:95

```
public static void main(String... args){
    String filename = "/u01/work" + args[0];
    //line n1
    //...
}
```

You want to validate a path name before the read file.Before validation, all path names should be canonicalized.

Which code inserted on line n1 will accomplish this?

```
Path file=Paths.get(filename);
Path canonicalPath=file.toAbsolutePath().toString();
FileInputStream fis=new FileInputStream(canonicalPath);
```

```
File file=new File(filename).getAbsoluteFile();
FileInputStream fis= new FileInputStream(file);
```

```
File file=new File(filename);
String canonicalPath=file.getCanonicalPath();
FileInputStream fis=new FileInputStream(f);
```

```
Path file=Paths.get(filename);
String canonicalPath=file.normalize().toString();
FileInputStream fis=new FileInputStream(canonicalPath);
```

Page no-96

```
Public class Test {
    public static void main(String[] args) {
        AnotherClass ac = new AnotherClass();
        SomeClass sc = new AnotherClass();
        ac = sc;
        sc.methodA ();
        ac.methodA();
    }
}
class SomeClass {
```

```

public void methodA() {
    System.out.println("SomeClass#methodA()");
}

}

class AnotherClass extends SomeClass {
    public void methodA () {
        System.out.println("AnotherClass#methodA()");
    }
}

```

What is the result?

a. SomeClass#methodA ()

 SomeClass#methodA ()

b. ClassCastException is thrown at runtime.

c. SomeClass#methodA ()

 AnotherClass#methodA ()

d. AnotherClass#methodA()

 AnotherClass#methodA ()

e. The compilation fails.

Page-97

```

module citizen {
    exports com.name to greeting;
}

And

module greeting {
}

```

Which statement is true?

- a.All members of com.name are accessible only to the citizen and greeting modules.
- b.All members of com.name package are accessible only to the greeting modules.
- ~~c.inserts "requires citizen;" at greeting's module-info.java,enables com.name members accessible to the greeting module.~~
- d.public members in the com.name package are accessible only to the greeting module.
- e.public members in the com.name package are accessible to all modules.

Page – 98

Given the data of the EMP table:

ID	NAME	DEPT
101	SMITH	HR
102	JONES	ENG
103	WEAVER	HR

Assuming that jdbcURL, username, and password are declared and initialised.

```
try (Connection conn = DriverManager.getConnection (jdbcURL, username,
password); PreparedStatement query = conn.prepareStatement ("SELECT ID, NAME
FROM EMP WHERE DEPT = ?"));

PreparedStatement update = conn.prepareStatement ("INSERT INTO RECRUITING
(ID, NAME)
VALUES (?, ?)");

query.setString (1, "HR");
ResultSet rs = query.executeQuery();
while (rs.next()) {
    update.setObject (1, rs.getObject (1, Integer.class), JDBCType. INTEGER);
    update.setObject (2, rs.getObject (2, String.class), JDBCType. VARCHAR);
    update.execute();
}
```

Which two happen upon execution?

- a.Two PreparedStatement objects are created.
 - b.Three Preparedstatement objects are created.
 - c.Memory leaks because Connection, PreparedStatements, and Resultset are not closed. closed
- A SQLException is thrown because the Resultset is not closed
- d.Three SQL statements are executed.
 - e.Two SQL statements are executed.

Page -99

Which two expressions create a valid Java Path instance?

- a.new Path("foo");
- b.Paths.get (URI.create("file: ///domains/oracle/test.txt"))
- c.Path.get(now URI(" file: ///domains/oracle/test.txt"))
- d.Paths.getPath("too")
- e.Paths.get("foo")

Page-100

Which module is required for any application using Swing or AWT?

- a.java.rmi
- b.java.prefs
- c.java.logging
- d.java.se
- e.java.desktop

Page -101

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

}

Which two methods modify field values?

- a.setTCount
- b.setCCount
- c.setGCount
- d.setACount
- e.setAllCounts



Page-102

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

And

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

And

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

What is the result?

- a.42
- b.17
- c.The compilation fails due to an error in line 1.
- d.The compilation fails due to an error in line 3.
- e. The compilation fails due to an error in line 5.
- f. The compilation fails due to an error in line 4.
- g. The compilation fails due to an error in line 2.

Page – 103

to submit your test.

5. Given the content from the coursen.txt file:

123: Java:1

124: MySQL:2

125: Java Server Pages: 3

Given the code fragment:

```
Path filePath = Paths.get("course.txt");

try {
/* line 1 /
} catch (IOException ex) |
System.out.format ("File 10 Exception is thrown.", ex);
```

 **Which code fragment at line 1 prints the lines that contain Java from the course.txt file?**

a. List<String> Lines2 = Files.readAllLines (filePath).filter(s->
s.contains ("Java"));
for (String line: lines2) {
System.out.println(line);
}

b. List<String> lines1-
Files.readAllLines (filePath).contains("Java");
for (String line: lines2)
System.out.print in (line);

c. System.out.println (Files, readstring(filePath).contains("Java"));

d. Files.lines(filePath).filter(s ->
s.contains("Java")).forEach(System.out::println);

e. Files. lines (filePath). map(s->
s.contains("Java").match (System.ont::printin);

 **Page-104**

File file1 = **new** File("file1.txt");

```
File file2 = new File("file2.txt");

try (BufferedReader reader =
        new BufferedReader(new FileReader(file1))) {
    System.out.println(reader.readLine());
    reader = new BufferedReader(new FileReader(file2));
    System.out.println(reader.readLine());
} catch (IOException e) {
    System.out.println("Error reading files");
}
```

What is the result?

- a.The Compilation fails.
- b.Error reading files is printed on the console.
- c.An unchecked exception is thrown at run time.
- d.The content from file1.txt and file2.txt is printed on the console.

 Page – 105

```
Public class Thing {
    int x,y,z;
    public Thing() {
        this(2,1);
        System.out.println(x + " , "+y+ " , "+z);
    }
    public Thing (int x) {
        System.out.println(x + " , "+y+ " , "+z);
    }
}
```

```
public Thing (int x, int y) {  
    this(2);  
    System.out.println(x +", "+y+", "+z);  
}  
}
```

And

```
Public class Tester {  
    Public static void main(String[] args) {  
        Thing t1 = new Thing();  
    }  
}
```

What is the result?

a. 0, 0 , 0

2 , 1 , 0

2 , 1 , 0

B .2 , 0 , 0

2 , 1 , 0

0 , 0 , 0

c. 0, 0 , 0

2 , 1 , 0

2 , 0 , 0

d. 1, 0 , 0

1 , 1 , 0

0 , 0 , 0



Page – 107

```
public class Tester {  
    public static int reduce(int x) {  
        int y= 4;  
        class Computer {  
            int reduce(int x) {  
                return x-y--;  
            }  
        }  
        Computer a = new Computer();  
        return a.reduce(x);  
    }  
public static void main(String[] args) {  
    System.out.println(reduce(1));  
}  
}
```

What is the result?

- a.An exception is thrown at runtime.
- b.-2
- c.The compilation fails.
- d.-3



page-108

```
public class Price {
```

```

private final double value;

public Price(String value) {
    this(Double.parseDouble(value));
}

public Price(Double value) {
    this.value = value;
}

public Price() {
}

public double getValue() {
    return value;
}

public static void main(String[] args) {
    Price p1 = new Price("1.99");
    Price p2 = new Price(2.99);
    Price p3 = new Price();
    System.out.println(p1.getValue() +"," +p2.getValue() +"," +p3.getValue());
}

```

What is the result?

- a.1.99,2.99
- b.1.99,2.99,0.0
- c.1.99,2.99,0
- d.The compilation fails

page -109

public class Person {



```

private String name;
private int age;

public Person(String name,int age) {
    this.name = name;
    this.age = age;
}

public int getAge () {
    return age;
}

public static void main(String[] args) {
    var persons = Arrays.asList(new Person("Max",18),
        new Person("Peter",23),
        new Person("Pamela",23),
        new Person("David",12));

    int num = persons.stream().mapToInt(Person::getAge)
        .filter(p -> p < 20)
        .reduce(0, (a,b) -> a+b);

    System.out.println(num);
}
}

```

What is the output?

- a.46
- b.30

c.41

d.35

Page-110

Given the data of the EMP table:

ID	NAME	DEPT
101	SMITH	HR
102	JONES	ENG
103	WEAVER	HR

Assuming that jdbcURL, username, and password are declared and initialised.

```
try (Connection conn = DriverManager.getConnection (jdbcURL, username,
password): PreparedStatement query = conn.prepareStatement ("SELECT ID,
NAME FROM EMP WHERE DEPT = ?");

PreparedStatement update = conn.prepareStatement ("INSERT INTO RECRUITING
(ID, NAME) VALUES (?, ?)") {
query.setString (1, "HR");
ResultSet rs = query.executeQuery();
while (rs.next()) {
update.setObject (1, rs.getObject (1, Integer.class), JDBCType.INTEGER);
update.setObject (2, rs.getObject (2, String.class), JDBCType.VARCHAR);
update.execute();
```

Which two happen upon execution?

- a. Memory leaks because connection, Preparedstatements, and Resultset are not closed.
- b. Three SQL statements are executed.
- c. A SQLException is thrown because the Resultset is not closed.

- d.Three PreparedStatement objects are created.
- e.Two SQL statements are executed.
- f.Two PreparedStatement objects are created.



page -111

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

What is the result?

- a.The compilation fails at line 9.
- b. The compilation fails at line 16.
- c.10
- d. The compilation fails at line 13.
- e.1



112.

```
import java.beans.PropertyChangeEvent;
import java.beans.PropertyChangeListener;
import java.beans.PropertyChangeSupport;
public class Main {
    private final PropertyChangeSupport pcs = new PropertyChangeSupport(this);
    private String name = "Test";
    public String getName() {
        return name;
    }
    public void setName(String name) {
        String oldName = this.name;
        this.name = name;
        pcs.firePropertyChange("Name", oldName, name);
    }
    public void addListener(PropertyChangeListener listener) {
        pcs.addPropertyChangeListener(listener);
    }
    public static void main(String[] args) {
        Main main = new Main();
        main.addListener(new PropertyChangeListener() {
            public void propertyChange(PropertyChangeEvent event) {
                System.out.println("Changed to " + event.getNewValue());
            }
        });
    }
}
```

```
});  
main.setName("Java");  
}  
}
```

What is the result?

- 1.The compilation fails
- 2.changed to Test
- 3.changed to Java**
- 4.nothing



113. Given the code fragment:

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

What is the result?

- 1.11
- 2.15**
- 3.23
- 4.21
- 5. 5



114. Given the code fragment:

```
module citizen {  
    exports com.name to greeting;  
}  
  
module greeting {  
}
```

Which statement is true?

- 1.All members of [com.name](#) are accessible only to the citizen and greeting modules.
- 2.All members in the [com.name](#) package are accessible only to the greeting module.**
- 3.Inserting "requires citizen;" at greeting's module-info.java, enables [com.name](#) members accessible to the greeting module.
- 4.public members in the [com.name](#) package are accessible only to the greeting module.
- 5.public members in the [com.name](#) package are accessible to all modules.

115. Which module defines the foundational APIs of the Java SE Platform?

- 1.java.lang
- 2.java.object
- 3.java.se**
- 4.java.base**

116. Which module is required for any application using Swing or AWT?

- 1.java.rmi
- 2.java.prefs

3.java.logging

4.java.se

5.java.desktop



117. Given the code fragment:

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

You want to examine whether path is a directory.

Which code inserted on line 1 will accomplish this?

- 1.BasicFileAttributes attributes = Files.getAttribute (path, "isDirectory");
- 2.BasicFileAttributes attributes =Files.readAttributes (path,
FileAttributes.class);

```
3.BasicFileAttributes attributes = Files.readAttributes (path,  
BasicFileAttributes.class);  
  
4.BasicFileAttributes attributes= Files.isDirectory(path);
```

118. Given the code fragment:



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

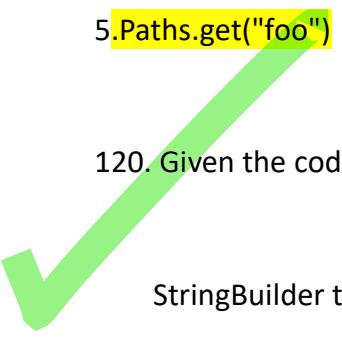
```
        System.out.println("Exception caught: ");
    }
}
}
```

What change on line 1 will make this code compile?

- 1.Add catch (M | Le)
- 2.Add catch (L | N e)**
- 3.Add catch (N | L | M e)
- 4.Add catch (L e)**
- 5.Add catch (L | M | N e)

 119. Which two expressions create a valid Java Path instance?

- 1.new Path("foo")
- 2.Paths.get(URI.create("file:///domains/oracle/test.txt"))**
- 3.Path.get(new URI("file:///domains/oracle/test.txt"))
- 4.Paths.getPath("too")
- 5.Paths.get("foo")**

 120. Given the code fragment:

```
StringBuilder txt1 = new StringBuilder("PPQRRRSTT");
int i = 0;
a:
while (i < txt1.length()) {
```

```
char x = txt1.charAt(i);

int j = 0; | 

i++;

b:

while (j < txt1.length()) {

    char y = txt1.charAt(j);

    if (i != j && y == x) {

        txt1.deleteCharAt(j);

        // line 1

    }

    j++;

}

System.out.println(txt1);
```

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

1.break b;

2.i--;

3.j--;

4.continue b;

5.continue a;

6.break a;



121.

```
public interface InterfaceOne {  
    public void methodA();  
    public void methodB();  
}  
  
public interface InterfaceTwo extends AbstractClass {  
}  
  
public abstract class AbstractClass implements InterfaceOne {  
    public String origin = "Abstract Class";  
    public void methodA() {  
        System.out.println("A");  
    }  
    public abstract void methodC();  
}
```

and

```
public class ConcreteClass extends AbstractClass {  
    public void methodC(String c) {  
        System.out.println(c);  
    }  
}
```

Which three changes make this code compile?

1. Implement methodC() in ConcreteClass
2. Remove methodA () from AbstractClass
3. Implement methodB () in Concreteclass
4. Implement methodA() in Concreteclass
5. Add the keyword abstract to the methodA () and methodB () declarations in InterfaceOne
6. Remove methodA () from InterfaceOne

7.InterfaceTwo should no longer extend AbstractClass

Page No: 122

Question 32:

Given:

```
public enum Status {  
    BRONZE(5), SILVER(10), GOLD(15);  
  
    private int rate;  
  
    private Status(int rate) {  
        this.rate = rate;  
    }  
  
    public int getRate() {  
        return rate;  
    }  
  
    public Status addStatus(int rate) {  
        return new Status(20);  
    }  
}  
  
and  
  
public class Test {  
    public static void main(String[] args) {  
        Status silver = Status.SILVER;  
        System.out.println(silver + silver.getRate());  
        Status platinum = Status.addStatus(20);  
        System.out.println(platinum + platinum.getRate());  
    }  
}
```

What is the result?

A. SILVER10

PLATINUM20

B. SILVER10

20

C. An exception is thrown at runtime.

D. SILVER10

platinum20

E. The compilation fails.

Page No: 123

Question 31:

Given:

```
public interface API { //line 1  
    public void checkValue (Object value)  
        throws IllegalArgumentException; //line 2  
    public boolean isValueANumber (Object val) {  
        if (val instanceof Number) {  
            return true;  
        }else {  
            try {
```

```

        Double.parseDouble(val.toString()));

        return true;

    }catch (NumberFormatException ex) {

        return false;

    }

}

}

```

Which two changes need to be made to make this class compile?

- A. Change Line 2 to an abstract method:

```

public abstract void checkValue (Object value)

throws IllegalArgumentException;

```

- B. Change Line 1 to an abstract class:

```

public abstract class API {

```

- C. Change Line 1 to extend java.lang.AutoCloseable:

```

public interface API extends AutoCloseable {

```

- D. Change Line 1 to a class:

```

public class API {

```

- E. Change Line 2 access modifier to protected:

```

protected void checkValue (Object value)

throws IllegalArgumentException;

```

Question 34:

Which three initialization statements are valid?

- A. var loc Set.of("UK", "US");
- B. var loc = Map.of("UK", 1, "US", 2);"
- C. var loc List.of("UK", "US");
- D. var loc = Set.of("UK", "US", "UK");
- E. var loc = ArrayList.of("UK", "US");
- F. var loc = Arrays.of("UK", "US", "ES");
- G. var loc = List.of("UK", null, "US");

Page No. 125

Question 33:

Given:

```
List<Integer> myList = Arrays.asList(9,8,9,2,7,2);
```

Which statement prints 2789?

- A. myList.stream()
.collect (Collectors.toCollection (SortedSet::new))
.stream().forEach (x -> System.out.print(x));

SortedSet is an interface in Java, not a concrete class like TreeSet. It cannot be instantiated directly, which makes this code invalid.

- B. myList.stream()
.collect (Collectors.toCollection (TreeSet::new))
.stream().forEach (x -> System.out.print(x));

- C. myList.stream()
.distinct()
forEach (x -> System.out.print(x));

The .distinct() method ensures only unique elements are considered. However, it retains the original order of the list rather than sorting it. The output for myList would be 9827

D. myList.stream()

Using HashSet removes duplicates but does not preserve order (since HashSet is unordered).

```
.collect(Collectors.toCollection(HashSet::new))  
.sorted().forEach(x -> System.out.print(x));
```

Page No. 126

Question 35:

Given the code fragment:

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

What is the output?

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

The code uses List.of() to create an empty, immutable list. Lists created with List.of() are not modifiable, meaning you cannot add, remove, or change elements in the list.

Page No. 127

Question 37:

```
class ConSuper {  
    protected ConSuper() {
```

```

        this(2);

        System.out.print("3");

    }

    protected ConSuper (int a) {

        System.out.print(a);

    }

}

and

public class ConSub extends ConSuper{

    ConSub () {

        this (4);

        System.out.print("1");

    }

    ConSub (int a) {

        System.out.print(a);

    }

    public static void main(String[] args) {

        new ConSub (4);

    }

}

```

What is the result?

- A. 2134
- B. 214
- C. 2314
- D. 234

Question 36:

Given:

```
class Test {  
    void display(int i) {  
        System.out.println("one");  
    }  
    void display (long l) {  
        System.out.println("two");  
    }  
    public static void main(String[] args) {  
        new Test().display(0B1010_0101_1001_0110);  
    }  
}
```

What is the result?

- A. The compilation fails.
- B. one**
- C. A NumberFormatException is thrown at runtime.
- D. Two

The binary literal 0B1010_0101_1001_0110 is used in the code.
By default, a binary literal is treated as an int unless it's explicitly suffixed with an L or L, which would make it a long

Page No: 129

Question 38:

Your organization provides a cloud server to your customer to run their Java code. You are reviewing the changes for the next release and you see this change in one of the config files:

old: JAVA_OPTS="\$JAVA_OPTS -Xms8g -Xmx8g"

new: JAVA_OPTS="\$JAVA_OPTS -Xms8g -Xmx8g -noverify"

Which is correct?

- A. You reject the change because -Xms8g -Xmx8g uses too much system memory.
- B. You accept the change because noverify is necessary for your code to run with the latest version of Java.
- C. You reject the change because noverify is a critical security risk.
- D. You accept the change because -noverify is a standard option that has been supported since Java 1.0.

Page No: 130

Question 39:

Given the code fragment:

```
public class FileHandler{  
    public static void main(String[] args) {  
        try (FileInputStream in = new FileInputStream("foo.txt")) {}  
        catch (FileNotFoundException e) {}  
    }  
}
```

Which two actions, independently, enable the code to compile?

Replacing the catch block with:

- A. `catch (FileNotFoundException | Exception e) {}
finally { in.close(); }`
- B. Inserting:
`finally { in.close(); }`
- C. Replacing the catch block with:
`catch (Exception e) {}`
- D. Adding throws IOException declaration at the main () method

E. Replacing the catch block with:

```
catch (Exception | IOException e) {}
```

F. Adding throws FileNotFoundException declaration at the main () method

 **Page No: 131**

Question 41:

Given:

```
String s = "Oracle";  
Runnable r= () -> {  
    System.out.println(s);  
};  
s = "Java";  
Thread t = new Thread(r);  
t.start();
```

What is the result?

- A. Java
- B. Oracle
- C. Compilation error s is effectively final
- D. An exception is thrown at run time.

 **Page No: 132**

Question 40:

Given the code fragment:

```
Supplier supplier = () -> "Hello World";
```

```
// line 1
```

Which statement on line 1 is calling the method of the supplier object correctly?

- A. System.out.println(supplier.test());
- B. System.out.println(supplier.accept());
- C. System.out.println(supplier.get());
- D. System.out.println(supplier.apply());

Page No: 133

Question 43:

Given the code fragment:

```
/* line n1 */  
A() {  
    super ("The Mandatory Criteria Yet to Meet");  
}  
}  
15. public class TestCE {  
16.     public static void main(String[] args) throws A {  
17.         int a = 10, b = 13;  
18.         try {  
19.             if (a < b) {  
20.                 throw new A();  
21.             }  
22.         }  
23.         catch (Exception e) { System.out.println(e); }  
24.         System.out.println("Continue...");  
25.     }
```

26.}

You must define the A exception class. The program execution must be terminated if the condition a is true and an A exception is thrown at line 20.

Which code fragment at line n1 defines A as per the requirement?

- A. class A extends RuntimeException {
- B. class A extends Exception {
- C. class A extends Throwable {
- D. class A extends ArithmeticException {

Page No: 134

Question 42:

Given the code fragment:

```
Locale locale = Locale.US;  
// line 1  
double currency = 1_00.00;  
System.out.println(formatter.format(currency));
```

You want to display the value of currency as \$100.00.

Which code inserted on line 1 will accomplish this?

- A. NumberFormat formatter = NumberFormat.getInstance(locale);
- B. NumberFormat formatter = NumberFormat.getCurrency (locale);
- C. NumberFormat formatter = NumberFormat.getInstance(Locale).getCurrency();
- D. NumberFormat formatter = NumberFormat.getCurrencyInstance (locale);

Page No: 135

Question 21:

Given:

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

What is the result?

- A. at once 1
- B. hat at store 4
- C. at once 0
- D. An IndexOutOfBoundsException is thrown at runtime.
- E. hat at store 1

Page No: 136

Question 19:

Given:

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

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

What is the result?

- A. An exception is thrown at runtime.
- B. Mr. Blue
- C. Mr. Green
- D. Green

Pg no : 137

25. Given:

package com.lti.multithreading;
public interface Converter {

 public static final double POUNDS_PER_KILOGRAM = 2.20462; //Line 1

```

public double tare();

public double net();

public default double gross() { //line 2

    return tare() + net();

}

public default double tare(String units) {

    return toUnit(tare(),units);

}

public default double net(String units) {

    return toUnit(net(),units);

}

public default double gross(String units) {

    return toUnit(gross(),units);

}

private static double toUnit(double kilograms, String unit) { //line3

    switch(unit) {

        case "KILO" : return kilograms;

        case "POUND" : return kilograms * POUNDS_PER_KILOGRAM;

        default : throw new IllegalArgumentException();

    }

}

}

```

Which is true?

- A. It compiles without errors.
- B. Line 3 is the first line to cause a compilation error.
- C. Line 2 is the first line to cause a compilation error.

- D. Line 1 is the first line to cause a compilation error.

Pg no : 138

24. Given:

```
public class A {  
    int a = 0;  
    int b = 0;  
    int c = 0;  
    public void foo(int i) {  
        a += b*i;  
        c -= b*i;  
    }  
    public void setB(int i) {  
        b = i;  
    }  
}
```

Which makes class A thread safe?

- A. Make foo and setB synchronized.
- B. Class A is thread safe.
- C. Make setB synchronized.
- D. Make foo synchronized.
- E. Make A synchronized.

Pg no :139

26.Given:

```
Public interface ExampleInterface{ }
```

Which two statements are valid to be written in this interface?

- A. final void methodG() {

 System.out.println("G");

}
- B. public String methodD();
- C. final void methodE();
- D. public int x;
- E. private abstract void methodC();

foo() modifies a and c based on the value of b. If another thread modifies b (via setB) while foo() is executing, the calculations could become inconsistent.

setB() modifies the value of b. If b is changed by one thread while another thread is executing foo(), the behavior would be unpredictable.

Synchronizing both methods ensures that only one thread can execute either foo() or setB() at a time, preventing interference between threads and maintaining consistent behavior.

F. **public void methodF() {**
 System.out.println("F");
}

G. **public abstract void methodB();**

pg no : 140

29. Given

```
List<String> states = List.of("NY", "CA", "WA", "NC", "CO");  
states.forEach(s -> System.out.println(s)); // line 1
```

Which statement is equivalent to line 1?

- A. **States.forEach((var s) -> System.out.println(s));**
- B. **States.forEach(var s -> {System.out.println(s)});**
- C. **States.forEach(String s -> {System.out.println(s);});**
- D. **States.forEach((s) -> System.out.println(s));**

Pg no : 141

1.Given:

```
public class StrBldr {  
  
    static StringBuilder sb1 = new StringBuilder("yo ");  
  
    static StringBuilder sb2 = new StringBuilder("hi ");  
  
    public static void main(String[] args) {  
  
        sb1 = sb1.append(new StrBldr().foo(new StringBuilder("hey")));  
  
        System.out.println(sb1);  
    }  
  
    StringBuilder foo(StringBuilder s) {  
  
        sb2 = sb2.append(s + " oh ");  
  
        return sb2;  
    }  
}
```

What is the result?

A compile time error occurs.

yo hi

hey oh hi yo

yo hi hey oh

hey oh yo hi

pg no : 142

30.Given:

interface Abacus{

public int calc(int a, int b);

}

public class Main1 {

public static void main(String[] args) {

int result = 0;

//Abacus aba = (a,b)->a*b;

//Abacus aba = (int e, int f)->{return e*f;};

Abacus aba = y,w -≥ x*y;

result = aba.calc(10,20);

System.out.println(result);

}

}

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

Abacus aba = (a,b)->a*b;

Abacus aba = (int e, int f)->{return e*f;};

Abacus aba = (int m, int n) -> {m*n}; Invalid. While parameter types are explicitly defined, the lambda body is missing a return keyword, which is required when using braces {}. This will result in a compilation error.
Abacus aba = (int l, j) -> { return l*j;};

Abacus aba = v,w -> x*y;

Pg no :143

4. Given:

var fruits = List.of("apple","orange","banana","lemon");

Optional<String> result = fruits.stream().filter(f->f.contains("n")).findAny();

System.out.println(result.get());

You replace the code on line 1 to use ParallelStream.

Which one is correct?

The compilation fails.

A NoSuchElementException is thrown at runtime.

The code will produce the same result.

The code may produce a different result.

Pg no : 144

13.Which two statements are true about a class that is marked @Deprecated?

The class cannot be extended.

Using the class is guaranteed to cause errors at runtime.

There is always another class that can be used instead of the deprecated class.

The author of the class wants to discourage people from using the class in any way.

Using the class will cause the Java compiler to give a warning.

Pg no : 145

6. Given the code fragment:



```
public class Main1 {  
  
    public static void main(String[] args) {  
  
        List<String> fruits = List.of("banana", "orange", "apple", "lemon");  
  
        Stream<String> s1 = fruits.stream();  
  
        Stream<String> s2 = s1.peek(i -> System.out.print(i + " "));  
  
        System.out.println("-----");  
  
        Stream<String> s3 = s2.sorted();  
  
        Stream<String> s4 = s3.peek(i -> System.out.print(i + " "));  
  
        System.out.println("-----");  
  
        String strFruits = (String) s4.collect(Collectors.joining(", "));  
    }  
}
```

When s2.peek is triggered: It prints the original order of the elements (banana, orange, apple, lemon).

What is the output?

A. ----

B. ----

banana orange apple lemon apple banana lemon orange

C. banana orange apple lemon apple banana lemon orange

D. banana orange apple lemon

Apple banana lemon orange

When s4.peek is triggered after sorting: It prints the sorted order of the elements (apple, banana, lemon, orange).

Output of collect: Combines the sorted elements into a single string: "apple,banana,lemon,orange".

The collect method is the terminal operation that triggers the lazy execution of all previous intermediate operations. As it collects the elements into a single string using a comma separator, both peek operations execute.

E. -----

Banana orange apple lemon
----apple banana lemon orange

Pg no : 147

35.Given :

```
1. public class Test{  
2.     private static class Greet{  
3.         private void print(){  
4.             System.out.println("Hello World");  
5.         }  
6.     }  
7.     public static void main(String[] args){  
8.         Test.Greet i = new Greet();  
9.         i.print();  
10.    }  
11.}
```

What is the result?

Hello World

The compilation fails at line 9.

The compilation fails at line 2

The compilation fails at line 8.

Pg no : 148

14.Given the code fragment from Box.java!

```
public class Box implements Serializable{
```

```
    private int boxId;  
    private String size;  
    private List items;  
}
```

Given the code fragment from Item.java:

```
public class Item{  
    private int id;  
    private String name;  
}
```

Given the information:

The classes Box and Item are encapsulated with getters and setters methods.

The classes Box and Item contains required constructors source code.

And the code fragment:

```
public static void main(String[] args) throws IOException{  
  
    List items1 = new ArrayList<>();  
  
    Items1.add(new Item(1,"Pen"));  
  
    Items1.add(new Item(2,"Ruler"));  
  
    Box b1 = new Box(123,"3",items1);  
  
    Try(FileOutputStream fout = new FileOutputStream("boxser.txt"));  
  
    ObjectOutputStream out = new ObjectOutputStream(fout);{  
  
        out.writeObject(b1);  
  
        out.flush();  
  
        out.close();  
  
    }catch(Exception e){  
  
        System.out.println("Unable to Serialize");  
  
    }}}
```

Which action serializes the b1 object?

- A. Implement the Serializable interface in the Item class.
- B. Remove out.flush() method invocation.
- C. Override readObject() and writeObject() methods in the Book class.
- D. Add SerialVersionUID to the Box and Item class.
- E. Handle NotSerializableException in the try clause or throw in the main() method definition.



35. Given:

```
1. public class Test{  
2.     private static class Greet{  
3.         private void print(){  
4.             System.out.println("Hello World");  
5.         }  
6.     }  
7.     public static void main(String[] args){  
8.         Test.Greet i = new Greet();  
9.         i.print();  
10.    }  
11.}
```

What is the result?

- A. Hello World
- B. The compilation fails at line 9.
- C. The compilation fails at line 2.
- D. The compilation fails at line 8.



37. Given the code fragment :

```
var i = 1;  
  
var result = IntStream.generate(()->{return i;})  
        .limit(100).sum();  
  
System.out.println(result);
```

Which statement prints the same value of result?

- A. System.out.println(IntStream.range(0,99).count());
- B. System.out.println(IntStream.rangeClosed(0,100).map(x->x).count());
- C. **System.out.println(IntStream.rangeClosed (1,100).count());**
- D. System.out.println(IntStream.range(1,100).count());

Pg no : 150



36. Given TripleThis.java:

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

Compiling TripleThis.java gives this compiler warning:

Note: TripleThis.java uses unchecked or safe operations.

Which two replacements remove this compiler warning and prints 12?

Replace line 12 with public static void printValue(Function f, Integer num){

Replace line 12 with public static void printValue(Function f, T num){

IntStream.range(1, 100) creates a stream from 1 (inclusive) to 100 (exclusive), containing 99 elements.

.count() returns 99, which does not match the result value (100).

IntStream.rangeClosed(1, 100) creates a stream from 1 (inclusive) to 100 (inclusive), containing exactly 100 elements.

Replace line 12 with public static void printValue(Function f, int num){
Replace line 9 with Function tripler = x ->{return (Integer)x*3;}
Replace line 9 with Function tripler = x ->{return x*3;}
Replace line 9 with Function tripler = x ->{return x*3;}

Pg no : 151

38.Given the code fragment:

```
public class Main {  
    public static void main(String[] args) throws IOException{  
        final Reader reader = new FileReader("File1.txt");  
        try(reader){  
            reader.read(); //Line 1  
        }finally {  
            reader.read(); //line 2  
        }  
        reader.read(); //line 3  
    }  
}
```

If file1.txt does exist, what is the result?

The program executes and prints nothing.

The compilation fails.

A java.io.IOException is thrown on line 2.

A java.io.IOException is thrown on line 1.

A java.io.IOException is thrown on line 3.

Question: 39

no: 153

page

```
File file1 = new File("file1.txt");
File file2 = new File("file2.txt");
try (BufferedReader reader = new BufferedReader(new FileReader(file1))) {
    System.out.println(reader.readLine());
    reader = new BufferedReader(new FileReader(file2));
    System.out.println(reader.readLine());
} catch (IOException e) {
    System.out.print("Error reading files");
}
```

What is the result?

- A) An unchecked exception is thrown at run time.
- B) Error reading files is printed on the console.
- C) The content from file1.txt and file2.txt is printed on the console.
- D) The compilation fails.

Question: 41

page

no: 154

When running jdeps, which three ways include dependent nonmodular jar files?

- A) jdeps -cp lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar
- B) jdeps --class-path lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar
- C) jdeps --module-path lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar
- D) jdeps lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar
- E) jdeps -classpath lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar
- F) jdeps application.jar
- G) jdeps --upgrade-module-path lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar



Question: 40

no: 155

page

```
class ConSuper {  
    protected ConSuper() {  
        this(2);  
        System.out.print("3");  
    }  
    protected ConSuper(int a) {  
        System.out.print(a);  
    }  
}  
public class ConSub extends ConSuper {  
    ConSub () {  
        this (4);  
        System.out.print ("1");  
    }  
    ConSub (int a) {  
        System.out.print (a);  
    }  
    public static void main(String[] args) {  
        new ConSub(4);  
    }  
}
```

What is the result?

- A) 214
- B) 2134
- C) 2341

D) 234



Question: 42

no: 156

page

```
import java.sql.Timestamp;  
  
public class Test {  
  
    public static void main(String[] args){  
  
        Timestamp ts = new Timestamp (1);  
  
    }  
  
}
```

and the commands:

```
javac Test.java  
  
jdeps -summary Test.class
```

What is the result on execution of these commands?

- A) On execution, the jdeps command displays an error.
- B) java.sql -> java.base -> Test.class
- C) Test.class -> java.base Test.class -> java.sql
- D) Test.class -> java.base Test.class -> java.sql java.sql -> java.base**

page

Question: 44

no: 157



Given the code fragment:

```
Stream<Integer> data = IntStream.range (1, 10000).boxed();  
  
Integer sum = data.mapToInt (a -> a).sum(); //line 1
```

Which two code fragments, independently, replace line 1 to implement the equivalent reduce operation?

- A) Integer sum = data.mapToInt (a -> a). reduce (0, (a,b)->a+b);**
- B) OptionalInt value = data.mapToInt (a -> a).parallel (). reduce ((a, b) -> a+b);**

- Integer sum = value.getAsInt();**
- C) **OptionalInt value = data.mapToInt (a -> a).parallel ().reduce (0, (a, b) -> a+b);**
- Integer sum = value.getAsInt();**
- D) **Integer sum = data.map (a -> a).reduce ((a, b) -> a+b);**
- E) **int s = 0;**
- Integer sum = data.map (a -> a).reduce(0, (a -> a+s));**

Question: 43

page

no: 158

Given the declaration:

```
@interface Resource {  
    String[] value ();  
}
```

Examine this code fragment:

```
/* Locl */ class Processorders{...}
```

Which two annotations may be applied at Locl in the code fragment?

- A) @Resource
- B) @Resource {"Customer1", "Customer2"})**
- C) @Resource (value={{}})
- D) @Resource ("Customer1")**
- E) @Resource

Question: 46

page

no: 159

Which two are valid statements?

- A) BiPredicate test = (Integer x, final Integer y) ->(x.equals (y));**
- B) BiPredicate test = (var x, final var y)-> (x.equals (y));**

- C) BiPredicate test = (final var x, y)-> (x.equals(y));
- D) BiPredicate test (Integer x, final var y)-> (x.equals(y));
- E) Bi Predicate test = (final Integer x, var y) ->(x.equals(y));

Question: 45

no: 160

page

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

You want to make the reduction operation parallelized.

Which two modifications will accomplish this?

- A) Replace line 1 with int sum = numbers.stream().parallel().reduce (0, (n, m) -> n + m);
- B) Replace line 1 with int sum = numbers. parallelStream().reduce (0, (n, m)->n+m);,
- C) Replace line 1 with int sum = numbers.stream().iterate (0, a -> a+1)
.reduce (0, (n, m)->n+m);.
- D) Replace line 1 with int sum = numbers.stream().flatMap (a -> a).reduce (0, (n, m) -> n
+m);.
- E) Replace line 1 with int sum = numbers. parallel ().stream().reduce (0, (n, m) -> n + m);.

Question: 47

page

no: 161

```
public static void main(String... args) {  
    String filename = "/u01/work" + args[0]; // line n1  
}
```

You want to validate a path name before the read file. Before validation, all path names should be canonicalized.

Which code inserted on line n1 will accomplish this?

A) Path file Paths.get (filename);

```
Path canonicalPath = file.toAbsolutePath().toString(); FileInputStream fis= new  
FileInputStream (canonical Path);
```

B) Path file Paths.get (filename);

```
String canonicalPath = file.normalize().toString(); FileInputStream fis= new  
FileInputStream (canonicalPath);
```

C) File file = new File (filename).getAbsoluteFile();

```
FileInputStream fis = new FileInputStream (file);
```

D) File file = new File(filename);

```
String canonical Path = file.getCanonicalPath();
```

```
FileInputStream fis = new FileInputStream(f);
```

Question: 49

page

no: 162

Assuming that jdbcURL, username, and password are declared and initialised.

```
try (Connection conn = DriverManager.getConnection(jdbcURL, username,  
password);
```

```

PreparedStatement query = conn.prepareStatement("SELECT ID, NAME FROM EMP
WHERE DEPT = ?");

PreparedStatement update = conn.prepareStatement("INSERT INTO RECRUITING
(ID, NAME) VALUES (?, ?)") {

    query.setString(1, "HR");

    ResultSet rs = query.executeQuery();

    while (rs.next()) {

        update.setObject(1, rs.getObject(1, Integer.class), JDBCType.INTEGER);

        update.setObject(2, rs.getObject(2, String.class), JDBCType.VARCHAR);

        update.execute();

    }

}

```

Which two happen upon execution?

- A) A SQLException is thrown because the Resultset is not closed.
- B) Three PreparedStatement objects are created.
- C) Memory leaks because connection, PreparedStatements, and ResultSet are not closed.
- D) Two PreparedStatement objects are created.**
- E) Three SQL statements are executed.**
- F) Two SQL statements are executed.

Question: 48

page

no: 163

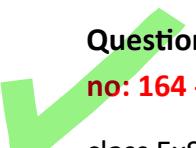
Given:

1. interface Pastry {
2. void get Ingredients();
3. }
4. abstract class Cookie implements Pastry {}
5. class ChocolateCookie implements Cookie {

```
7. public void getIngredients () {}  
8.  
9. class CoconutChocolateCookie extends ChocolateCookie {  
10. void getIngredients (int x) {}  
11.}
```

Which is true?

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

 Question: 50

no: 164 - 165

```
class ExSuper extends Exception {  
    private final int eCode;  
    public ExSuper(int eCode, Throwable cause) {  
        super(cause);
```

page

```

        this.eCode = eCode;
    }

    public ExSuper(int eCode, String msg, Throwable cause) {
        super(msg, cause);
        this.eCode = eCode;
    }

    public String getMessage() {
        return this.eCode + ": " + super.getMessage() + "-" +
            this.getCause().getMessage();
    }
}

class ExSub extends ExSuper {

    public ExSub(int eCode, String msg, Throwable cause) {
        super(eCode, msg, cause);
    }
}

```

And the Code Fragment

```

try {
    String param1 = "Oracle";
    if (param1.equalsIgnoreCase("oracle")) {
        throw new ExSub(9001, "APPLICATION ERROR-9001", new
            FileNotFoundException("MyFile.txt"));
    } throw new ExSuper(9001, new FileNotFoundException("MyFile.txt")); // Line 1
} catch (ExSuper ex) {
    System.out.println(ex.getMessage());
}

```

What is the result?

- A) Compilations fails at Line 1.
- B) 9001: APPLICATION ERROR-9001-MyFile.txt**
- C) 9001: java.io.FileNotFoundException: MyFile.txt-MyFile.txt

D) 9001: APPLICATION ERROR-9001-MyFile.txt

9001: java.io.FileNotFoundException: MyFile.txt-MyFile.txt

Question: 5

166

page no:

```
public enum Season{  
    WINTER ('w'), SPRING ('a'), SUMMER ('h'), FALL('l');  
    char c;  
    private Season (char c) {  
        this.c = c;  
    }  
}
```

and the code fragment:

```
public static void main(String[] args) {  
    Season[] SA = Season.values();  
    // Line n1  
}
```

Which three code fragments, at line n1, prints SPRING?

- A. System.out.println (Season. SPRING);
- B. System.out.println (Season.valueOf("SPRING"));
- C. System.out.println(SA[0]);
- D. System.out.println(Season. value f('s'));
- E. System.out.println (Season. valueof ("SPRING").ordinal());
- F. System.out.println (Season. values (1));
- G. System.out.println(sA[1]);

Question: 4

no: 167

page

```
class MyType<T> {  
    private T value;  
    public T getValue() {  
        return value;  
    }  
    public void setValue (T value) {  
        this.value = value;  
    }  
}
```

and

```
public class Test {  
    public static void main(String... args) {  
        MyType<String> strType = new MyType<>();  
        MyType<? extends Number> type = new MyType<>();  
        strType.setValue("test");  
        type.setValue(null);  
        System.out.println(strType.getValue()+":"+type.getValue());  
    }  
}
```

What is the result?

- A. The compilation fails.
- B. test:0
- C. null:null
- D. An Exception is thrown at runtime.
- E. test:null



Question: 8

no: 168

page

```
var c = new CopyOnWriteArrayList<>(List.of("1", "2", "3", "4"));

Runnable r = () -> {
    try {
        Thread.sleep(150);
    } catch (InterruptedException e) {
        System.out.println(e);
    }
    c.set(3, "four");
    System.out.print(c + " ");
};

Thread t = new Thread(r);
t.start();
for (var s : c) {
    System.out.print(s + " ");
    Thread.sleep(100);
}
```

What is the output?

- A. 12 [1, 2, 3, four] 3 four
- B. 1 2 [1, 2, 3, 4] 3 4
- C. 1 2 [1, 2, 3, four] 3 4
- D. 1 2 [1, 2, 3, 4] 3 four

Question 7:

```
-continent
| a.txt
|- country

| b.txt
|- state

| c.txt
|+ county
```

and

```
BiPredicate pred = (path, fileAttrs) -> {
    return fileAttrs.isDirectory();
};
int depth = 1;
try(var stream = Files.find(Paths.get("/continent"),depth,pred)) {
    stream.forEach(System.out::println);
} catch(IOException e) { }
```

At depth 0: /continent (the root directory itself) is included.

What is the result?

- A. /continent
/continent/country
/continent/country/state
/continent/country/state/county

At depth 1: Immediate children are considered:

a.txt is a file, not a directory, so it's ignored.

country is a directory, so it is included.

- B. /continent
/continent/country

- C. /continent/country/state

- D. /continent/country/state/county

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

You want to implement the java.io.Serializable interface to the MyPersistenceData class.

Which method should be overridden?

- A. The **readExternal** and **writeExternal** method.
- B. Nothing**
- C. The **readExternal** method
- D. The **writeExternal** method

Pg – 171

Question 14:

Which two can be considered good practices for serializing Java objects?

- A. Ensure that the class definition used is the same as the class definition used by Java runtime at the time when the object was serialized.**
- B. Always override the readObject/writeObject methods from the java.io.Serializable interface.
- C. Implement serialization for long-term data storage.
- D. Assign null value by default while serializing and deserializing a transient variable.
- E. Implement secure serialization by generating secure object hash or using encryption.**

Pg – 172

Question 16:

```
public class CreateArrayListExample {  
    public static void main(String[] args) {  
        List vegetables = new ArrayList<>();  
        vegetables.add("Kale");  
        vegetables.add(0,"Lettuce");  
        System.out.println(vegetables);  
        List fish = new ArrayList<>();  
        fish.add("Salmon");
```

```
        fish.add(0,"Seabass");
        System.out.println(fish);

    }

}
```

What is the result?

- A. A compilation error is thrown
- B. [Lettuce, Kale]
[Seabass, Salmon]
- C. [Lettuce, Kale]
- D. [Kale , Lettuce]
[Salmon, Seabass]

 Pg – 173

Question 17:

```
var h = new HashMap();
String[] k = { "1", "2", null, "3" };
String[] v = { "a", "b", "c", null };

for (int i = 0; i < 4; i++) {
    h.put(k[i], v[i]);
    System.out.println(h.get(k[i]) + " ");
}
```

What is the result?

- A. a b c null
- B. a b c
- C. a b followed by an exception
- D. a b c followed by an exception

 Pg – 174

Question 26:

```
public interface ExampleInterface { }
```

Which two statements are valid to be written in this interface?

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

Pg – 175

Question 18:

```
public class Color {  
    String hue;  
    int value;  
    public Color(String hue, int value) {  
        this.hue = hue;  
        this.value = value;  
    }  
    @Override  
    public String toString() {  
        return this.hue + ":" + this.value;  
    }  
    public static void main(String[] args) {  
        List<Color> clrs = List.of(new Color("Red",100),  
            new Color("Yellow", 50),  
            new Color("Red",75),
```

```

        new Color("Yellow",75));
    Comparator hueSrtr = (h1, h2) -> h1.hue.compareTo(h2.hue);
    Comparator valueSrtr = (h1, h2) -> { if ( h1.value >= h2.value) {
                                            return 1;
                                } else {
                                            return -1;
                                }
        };
    clrs.sort(hueSrtr.thenComparing(valueSrtr));
    System.out.println(clrs);
}
}

```

What is the result?

- A. [Red:100, Red:75, Yellow:75, Yellow: 50]
- B. [Red:75, Red:100, Yellow:50, Yellow:75]
- C. [Yellow:50, Yellow: 75, Red:75, Red:100]
- D. An Exception is thrown at runtime
- E. [Yellow:75, Yellow: 50, Red:100, Red:75]

The `List.of` method creates an immutable list. This means the list returned cannot be modified. Attempting to call `clrs.sort(...)` on an immutable list will result in an `UnsupportedOperationException` at runtime.



Pg – 176

Question 27:

```

public interface APIInterface {
    public default void process() { System.out.println("Process() called 1."); }
}

```

and

```

public abstract class AbstractAPI {
    public abstract void process();
}

```

and

```

public class Apimpl extends AbstractAPI implements APIInterface {
    public void process() {
        System.out.println("Process() called 2.");
    }
}

```

```
    }
    public static void main(String[] args) {
        var impl = new Apimpl();
        impl.process();
    }

}
```

What is the result?

- A. A java.lang.NoSuchMethodException is thrown
- B. A java.lang.IllegalAccessException is thrown
- C. The compilation fails
- D. The program prints Process() called 2.**
- E. The program prints Process() called 1.

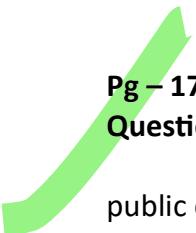
 Pg – 177

Question 29:

```
List<String> states = List.of("NY", "CA", "WA", "NC", "CO");
states.forEach(s -> System.out.println(s)); //line1
```

Which statement is equivalent to line 1?

- A. states.forEach((var s) -> System.out.println(s));**
- B. states.forEach(var s -> System.out.println(s));
- C. states.forEach((String s) ->{ return System.out.println(s);});
- D. states.forEach((s) -> System.out.println(s));

 Pg – 178

Question 28:

```
public class Tester {
    private int x;
    private static int y;
```

```

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

}

```

What is the result?

A. 2,3
4,5
4,5

B. 2,5
4,5
4,5

C. 2,3
4,3
4,5

D. 2,3
4,5
4,3

Pg – 179
Question 30:

```

public interface Abacus {
    public int calc ( int a , int b);
}

```

```

public class Main {
    public static void main(String[] args) {
        int result = 0;
        // line 1
    }
}

```

```
        result = aba.calc(10,20);
        System.out.println(result);
    }

}
```

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

- A. Abacus aba = (a, b) -> a*b;
- B. Abacus aba = (int e, int f) -> { return e * f; };
- C. Abacus aba = (int m, int n) ->{ m*n }; **with curly braces , we should write return statement**
- D. Abacus aba = (int i, j) -> {return i * j; };
- E. Abacus aba = v, w -> x * y;

Pg – 180

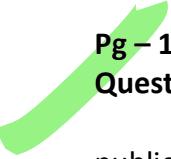
Question 31:

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

public class Bar extends Foo{
    public void foo(Collection arg) {
        System.out.println("Hello world!");
    }
    public static void main(String[] args) {
        List<String> li = new ArrayList<>();
        Bar b = new Bar();
        Foo f = b;
        b.foo(li);
        f.foo(li);
    }
}
```

What is the output?

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

 Pg – 181 & 183

Question 32:

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

and

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

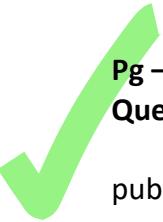
and

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

Which three are true?

- A. f1.foo(c) prints Bonjour le monde!

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

 Pg – 182
Question 33:

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

and

```
public class Bar extends Foo{  
    public void foo(List arg) {  
        System.out.println("Hello world!");  
    }  
    public static void main(String[] args) {  
        List<String> li = new ArrayList<>();  
        Collection<String> co = li;  
        Bar b = new Bar();  
        b.foo(li);  
        b.foo(co);  
    }  
}
```

What is the output?

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

Pg – 184

Question 34:



```
public class DoClass{  
    static String s;  
    public static void main(String[] args) {  
        switch(s) {  
            case "41" : s+="41";  
            default : s+=" def ";  
            case "42" : s+="42";  
        }  
        System.out.println(s);  
    }  
}
```

Since s is null, this will cause a NullPointerException at runtime because the switch statement cannot operate on a null value.

What is the output?

- A. 41 def 42
- B. def 42
- C. null
- D. An exception is thrown at runtime.



185: Which module is required for any application using Swing or AWT?

- a. Java.desktop
- b. Java.logging

- c. Java.se
- d. Java.rmi
- e. Java prefs



186: Given the code fragment

```
Lomate Locale = Locale.US;  
// line 1  
  
double currency = 100.00;  
  
System.out.println(formatter.format (currency));
```

You want to display the value of currency as \$100.00.

Which code inserted on line 1 will accomplish this?

- a. NumberFormat formatter = NumberFormat.getInstance(locale);
- b. NumberFormat formatter = NumberFormat.getCurrency(locale);
- c. NumberFormat formatter = NumberFormat.getCurrencyInstance(locale);**
- d. NumberFormat formatter = NumberFormat.getInstance(locale).getCurrency();



187: Which two statements are true about running code on the class path and the module path?

- a. A modular JAR placed on the -classpath results in a named application module
- b. A Non-modular JAR placed on the path --module--path results in a named application module

c. A modular JAR placed on the module --module-path results in a named application module.

d. A non-modular JAR placed on the -classpath results in an unnamed module

e. A modular JAR placed on the -classpath results in an automatic module.

188:

```
enum Season {  
    WINTER('w'), SPRING('s'), SUMMER('h'), FALL('f');  
  
    char ch;  
  
    private Season(char ch) {  
        this.ch = ch;  
    }  
}
```

and the code fragment:

```
public class Main {  
  
    public static void main(String[] args) {  
        Season[] sA = Season.values();  
        //line n1  
    }  
}
```

Which three code fragments, at line n1, prints SPRING?

a. System.out.println(Season.SPRING);

b. System.out.println(Season.valueOf("SPRING"));

- c. System.out.println(sA[0]);
- d. system.out.println(Season.valueOf('s'));
- e. System.out.printin(Season.valueOf("SPRING").ordinal());
- f. System.out.println(season.values(1));
- g. System.out.println(sA[1]);**



189:

Given the content from the course.txt file:

123:Java:1

124:MySQL:2

125:Java Server Pages: 3

Given the code fragment:

```
Path filePath = Paths.get("course.txt");
try {
    // line 1
    Files.lines(filePath).filter(s -> s.contains("Java")).forEach(System.out::println);
} catch (IOException ex) {
    System.out.format(" file IO Exception is thrown.");
}
```

Which code fragment at line 1 prints the lines that contain Java from the course.txt file?

- a. Files.lines(filePath).map(s ->
s.contains("Java")).forEach(System.out::println);
- b. Files.lines(filePath).filter(s -> s.contains("Java")).forEach(System.out::println);**
- c. List<String> lines2 = Files.readAllLines(filePath).filter(s -> s.contains("Java"));

```
for(String line:lines2){  
    System.out.println(line);  
}  
  
d. System.out.println(Files.readString(filePath).contains("Java"));  
  
e. List<String> lines1 = Files.readAllLines(filepath).contains("Java");  
  
for(String line:lines2){  
    System.out.println(line);  
}
```

190:



```
class MyType<T>{  
    private T value;  
    public T getValue() {  
        return value;  
    }  
    public void setValue(T value) {  
        this.value = value;  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        MyType<String> strType = new MyType<>();  
        MyType<? extends Number> type= new MyType<>();  
        strType.setValue("test");  
        type.setValue(null);  
        System.out.println(strType.getValue()+" : "+ type.getValue());  
    }  
}
```

```
    }  
}  
  
}
```

What is the result:

- a. The compilation fails.
- b. test : 0
- c. null : null
- d. An Exception is thrown at runtime.
- e. test : null**

191:

```
var c = new CopyOnWriteArrayList<>(List.of("1", "2", "3", "4"));  
  
Runnable r=()->{try{Thread.sleep(150);}catch(  
InterruptedException e)  
{  
    System.out.println(e);  
}}  
  
c.set(3,"four");System.out.println("c= "+c+" ");  
  
Thread t = new Thread(r);  
t.start();  
  
for(var n:c)  
{  
    System.out.println(n + " ");  
    Thread.sleep(100);  
}
```

What is the output:

- a. 1 2 [1, 2, 3, four] 3 four
- b. 1 2 [1, 2, 3, 4] 3 4
- c. 1 2 [1, 2, 3, four] 3 4
- d. 1 2 [1, 2, 3, 4] 3 four

192: give the following code fragment:

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

What is the output:

- a. 21
- b. 11
- c. 23
- d. 5
- e. 15

193: given the directory

continent

```
| a.txt  
| country  
| b.txt  
| - state  
| c.txt  
| + country
```

And

```

BiPredicate pred = (Paths, fileAttrs) -> {

    return fileAttrs.isDirectory();      // return ((BasicFileAttributes)
fileAttrs.isDirectory();

};

int depth = 1;

try(var stream = Files.find(Paths.get("/continent") , depth, pred)){

    stream.forEach(System.out::println);

} catch(IOException e){}

```

What is the result?

- a. /continent/
/continent/country
/continent/country/state
/continent/country/state/country
- b. /continent
/continent/country
- c. /continent/country/state
- d. /continent/country/state/country



194: class Test{

```

private final int x = 1;

static final int y;

public Test() {
    System.out.println(x);
}

```

```
        System.out.println(y);  
    }  
  
}  
  
public class Main {  
  
    public static void main(String[] args) {  
        new Test();  
    }  
}
```

What is the result?

- a. The compilation fails at line 9.
- b. 1
- c. The compilation fails at line 13.
- d. The compilation fails at line 16.
- e. 10

//Acutal answer compilation error occurs at line 10.



195:
int i = 3;

int j = 25;

```
System.out.println(i > 2 ? i > 10 ? i * (j+10) : i*j +5 : i);
```

What is the result?

- a. 25
- b. 395
- c. The compilation fails.

d. 3

e. 80

196:

Why would you choose to use a peek operation instead of a forEach operation on a Stream?

- a. To process the current item and return void.
- b. To remove an item from the end of the stream.
- c. To process the current item and return a stream.
- d. To remove an item from the beginning of the stream.

197: Which two can be considered good practices for serializing Java objects?

- a. Ensure that the class definition used is the same as the class definition used by Java runtime at the time when the object was serialized.
- b. Always override the readObject/writeObject methods from the java.io.Serializable interface.
- c. Implement serialization for long-term data storage.
- d. Assign null value by default while serializing and deserializing a transient variable.
- e. Implement secure serialization by generating secure object hash or using encryption.

198:

```
List<String> fruits = List.of("banana", "orange", "apple", "lemon");

Stream<String> s1 = fruits.stream();

Stream<String> s2 = s1.peek(i -> System.out.print(i + " "));

System.out.println("-----");

Stream<String> s3 = s2.sorted();

Stream s4 = s3.peek(i -> System.out.print(i + " "));

System.out.println("-----");
```

```
String strFruits = (String) s4.collect(Collectors.joining(", "));
```

What is the output?

- a. banana orange apple lemon apple banana lemon orange

- b. banana orange apple lemon

apple banana lemon orange

- c. -----

banana orange apple lemon apple banana lemon orange

- d. -----
banana orange apple lemon
-----apple banana lemon orange

199: question is completely blur

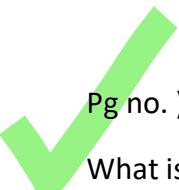
200:

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

Mark for Review You want to implement the java.io.Serializable interface to the MyPersistenceData class.

Which method should be overridden?

- a. The readExternal and writeExternal method
- b. Nothing
- c. The readExternal method
- d. The writeExternal method



Pg no.) 201, question-18)

What is the result?

```
package com.ltim.another;

import java.util.Comparator;

import java.util.List;

public class Color {

    String hue;
    int value;

    public Color (String hue, int value) {
        this.hue = hue;
        this.value = value;
    }

    public String tostring () {
        return this.hue + ":"+ this.value;
    }

    public static void main (String[] arga) {
        List clrs = List.of(new Color ("Red", 100),
                           new Color("Yellow", 50),
                           new Color("Red", 75),
                           new Color ("Yellow", 75));
        Comparator hueSrtr = (h1, h2) -> h1.hue.compareTo(h2.hue);
    }
}
```

```

Comparator valuesrtr = (h1, h2) ->{
    if (h1.value>h2.value) {
        return 1;
    }
    else{
        return -1;
    }
};

clrs.sort (hueSrtr.thenComparing (valueSrtr));
System.out.println(clrs);
}
}

```

Options :

- A) [Red:100, Red:75, Yellow:75, Yellow:50]
- B) [Red:75, Red:100, Yellow:50, Yellow:75]\
- C) [Yellow:50, Yellow:75, Red:75, Red:100]
- D) An Exception is thrown at runtime.**
- E) [Yellow:75, Yellow:50, Red:100, Red:75]

Pg no. 202 ,

Output of the Code?

package com.ltim.another;

class Person {

```

        private String name = "Green";
        public void setName(String name) {
            String title = "Mr.";
            name = title + name;
        }
    }

```

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

Options:

- A) Green
- B) Mr. Green
- C) Mr. Blue
- D) An Exception occurs at runtime

Pg no) 203

Question)

var h = new HashMap();

```
String[] k = { "1", "2", null, "3" };  
String[] v = { "a", "b", "c", null };
```

```
for (int i = 0; i < 4; i++) {
```

```
    h.put(k[i], v[i]);  
    System.out.println(h.get(k[i]) + " ");  
}
```

Options)

What is the result?

- A) a b c
- B) a b c null**
- C) a b followed by an exception
- D) a b c followed by an exception

Pg) 204

Question)

```
String name = " ";  
if( //insert code here// n1 ) {  
    System.out.println("Name is required");  
}
```

What should be inserted at line n1 so that the code fragment prints Name is required?

Options)

- A) name.isBlank()**
- B) name.compareTo("*")=0
- C) name.isEmpty()
- D) name.trim == " "

B) name.compareTo("*") == 0:

This code is invalid syntax. The argument "*" is not meaningful in the context, and compareTo is used for lexicographical comparison, not for checking blank strings.

C) name.isEmpty():

The isEmpty() method checks if the string's length is 0 (i.e., an empty string ""). Since name contains one space (" "), this condition returns false, and "Name is required" would not be printed.

D) name.trim() == " ":

The trim() method removes leading and trailing whitespace from the string. After trimming, name becomes an empty string (""). Comparing it to "" would return false, so the message would not be printed.

Pg no) 205

Question)

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

What is the result?

Options)

- A) at once 1
- B) hat at store 4
- C) at once 0
- D) An IndexOutOfBoundsException is thrown at runtime.
- E) hat at store 1**

Pg 206)

Question)

```
public class Tester {  
  
    public static void main(String[] args) {  
  
        int x = 0;  
  
        while (x < 10) {  
  
            System.out.print(x++);  
        }  
  
    }
```

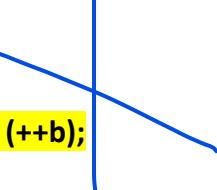
}

Which "for" loop produces the same output?

Options)

A)

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



B) for (int d = 0; d <10;){

```
    System.out.print (d);  
    ++d;  
}
```

C) for (a; a < 10; a++){

```
    System.out.print (a);  
}
```

D) for(int c=0 ;; c++){

```
    System.out.print(c);  
    if(c==10){  
        break;  
    }
```

Pg no. 207)

package com.ltim.another;



```
public interface Converter {  
  
    public static final double POUNDS_PER_KILOGRAM = 2.20462; //Line 1  
    public double tare();  
    public double net();  
    public default double gross() {// line 2  
        return tare () + net ();  
    }  
  
    public default double tare(String units) {  
        return toUnit (tare (), units);  
    }  
  
    public default double net (String units) {  
        return toUnit (net(), units);  
    }  
  
    public default double gross (String units) {  
        return toUnit (gross(), units);  
    }  
  
    private static double toUnit (double kilograms, String unit) { // line 3  
        switch (unit) {  
            case "KILO": return kilograms;  
            case "POUND": return kilograms * POUNDS_PER_KILOGRAM;
```

```
        default: throw new IllegalArgumentException();  
    }  
}  
}  
}
```

Which is true?

- A) It compiles without errors.**
- B) Line 3 is the first line to cause a compilation error.
- C) Line 2 is the first line to cause a compilation error.
- D) Line 1 is the first line to cause a compilation error.

Pg 208)
Questions)

```
import java.sql.Timestamp;  
  
public class Tester {  
  
    public static void main(String[] args) {  
  
        Timestamp ts = new Timestamp(1);  
  
    }  
}
```

and run these commands

```
javac Test.java  
jdeps -summary Test.class
```

What is the result on execution of these commands?

Options)



A) Test.class -> java.base Test.class -> java.sql

B) On execution, the jdeps command displays an error.

C) Test.class -> java.sql -> java.base

D) Test.class -> java.base Test.class -> java.sql java.sql > java.base

Pg 210)

```
package com.ltim.another;

public class A {
    int a = 0;
    int b = 0;
    int c = 0;
    public void foo (int i) {
        a += b * i;
        c -= b * i;
    }
    public void setB(int i) {
        b = i;
    }
}
```

Which makes class A thread safe?

- A) Make foo and setB synchronized.**
- B) Class A is thread safe.
- C) Make setB synchronized.
- D) Make foo synchronized.
- E) Make A synchronized.



Pg 211)

19. Given the code fragment:

```
Stream<Integer> data = IntStream.range(1, 10000).boxed():
```

```
Integer sum = data.mapToInt(a -> a).sum(); //line 1
```

Ques) Which two code fragments, independently, replace line 1 to implement the equivalent reduce operation?

A) OptionalInt value = data.mapToInt (a -> a).parallel().reduce(0, (a, b) -> a+b);

```
Integer sum = value.getAsInt();
```

B) Integer sum = data.map (a -> a).reduce((a+b)-> a+b);

|

C) OptionalInt value = data.mapToInt(a->a).parallel().reduce((a, b) -> a+b);

```
Integer sum = value.getAsInt()
```

D) int s=0;

```
Integer sum = data map(a -> a).reduce(0,(a-> a+9));
```

E) Integer sum = data.mapToint(a -> a).reduce(0, (a,b)->a-b);

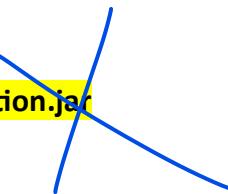


Pg 212)

13. When running jdeps, which three ways include dependent nonmodular jar files?

- A) jdeps -classpath lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar
- B) jdeps -cp lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar**
- C) jdeps --module-path lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar
- D) jdeps lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar
- E) jdeps --class-path lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar**

F) jdeps --upgrade-module-path
lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar



F) jdeps application.jar

Pg no 213)

Question)

public static void main(String[] args) {
 var lst = List.of(1, 2.0f, "4.0");
 for (var e : lst) {
 System.out.print ("> "+ e);
 }
 System.out.println();
 lst.add(2, 3); // line n1
 for (int c= 0; c < lst.size(); c++) {
 display(lst.get(c));
 }
}

public static void display(var c) { // line n2

```
        System.out.print(">" + c);  
    }
```

What is the result?

Options

- A) > 1> 2.0>4.0
- B) > 1> 2.0> 3> 4.0
- C) A compile time error occurs at line n2.
- D) An exception is thrown at line n1.**
- E) 1> 2.0> 4.0

Pg no 214)

Question)

```
Path p1 = Paths.get("/scratch/exam/topsecret/answers");  
Path p2 = Paths.get("/scratch/exam/answers/temp.txt");  
Path p3 = Paths.get("/scratch/answers/topsecret");
```

Which two statements print ..\..\..\answers\topsecret?

- A) System.out.print (p2.relativize(p1));**
- B) System.out.print (p1.relativize(p2));
- C) System.out.print(p3.relativize(p2));**
- D) System.out.print(p2.relativize(p3));
- E) System.out.print (p1.relativize(p3));**
- F) System.out.print (p3.relativize(p1));

 Pg 215)

Question)

```
public class Employee {  
    private String locality;
```

```
private int salary:  
// the setter and getter methods go here  
8. List roster = createEmployeeList();  
9. double average = roster  
10. .stream();  
11. /* insert code here */  
12. System.out.println(average);
```

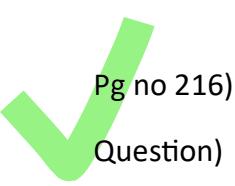
Which code fragment Inserted on line 11 prints the average salary of all employees from the Bay Area?

A) .filter(e -> e.getLocality().equals("Bay Area"))
.average (Employee: :getSalary)
.getAsDouble();

B) .filter(e -> e.getLocality().equals("Bay Area"))
.filter(s -> s.getSalary())
. average()
.getAsDouble();

C) .filter(e -> e.getLocality() .equals ("Bay Area"))
.mapToInt (Employee: :getSalary)
. average()
.getAsDouble();

D) .collect (Collectors.groupingBy (Employee :: getLocality,
Collectors.averagingDouble (Employee :: getSalary)));



Pg no 216)

Question)

Given

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

Compiling TripleThis.java gives this compiler warning:

Note: TripleThis.java uses unchecked or unsafe operations.

Which two replacements remove this compiler warning and prints 12?

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



PAGE 219

```
public class DNASynth {  
    int aCount;  
    int tCount;  
    int cCount;  
    int gCount;  
  
    DNASynth(int a, int tCount, int c, int g) {  
        this.aCount = a;  
        this.tCount = tCount;  
        this.cCount = setCCount(c);  
        setGCount(g);  
    }  
  
    int setCCount(int c) {  
        return c;  
    }  
  
    void setGCount(int gCount) {  
        this.gCount = gCount;  
    }  
  
    public static void main(String[] args) {  
        DNASynth dnaSynth = new DNASynth(5, 10, 15, 20);  
        System.out.println("A Count: " + dnaSynth.aCount);  
    }  
}
```

```
    System.out.println("T Count: " + dnaSynth.tCount);
    System.out.println("C Count: " + dnaSynth.cCount);
    System.out.println("G Count: " + dnaSynth.gCount);
}
}
```

Which two lines of code when inserted in line 1 correctly modifies instance variables?

tCount = tCount;

setGCount (g);

|

cCount = setCCount (c);

setCCount(c)=cCount;

aCount = a;

PAGE 220

```
import java.util.ArrayList;
import java.util.Collection;
import java.util.HashMap;
```

```
import java.util.Map;

public class X {
    private Collection collection;

    public void set(Collection collection) {
        this.collection = collection;
    }
}

public class Y extends X {
    public void set(Map<String, String> map) {
        super.set(map.values()); // option 1
        // or
        set(map.values()); // option 2
    }
}

public static void main(String[] args) {
    Y y = new Y();
    Map<String, String> map = new HashMap<>();
    map.put("key1", "value1");
    map.put("key2", "value2");

    y.set(map);
}

// Print the collection to verify
```

```
// Assuming the collection is stored in X, but private  
// We can't directly access it in Y, so no actual print here  
System.out.println("The map values have been set in the collection");  
}  
}
```

Which two lines can replace line 1 so that the Y class compiles?

super.set (List map)
super.set (map.values ());
set (map.values ());
map.forEach((k, v) -> set(v));
set (map)

PAGE 221

List original - new

```
ArrayList<>(ArrayList (1,2, 3,4,5));
```

Which two code fragments remove the elements from the original list?

```
List s1 = Collections.synchronizedList (original);  
for (Integer w : s1)  
s1.remove (w);
```

```
List cwa = new CopyOnwriteArrayList<(original);  
for (Integer w : cwa)
```

```
cwa.remove(w);
```

```
List al = new Arraylist(original);
```

```
for (Integer w: al)
```

```
al.remove(w)
```

```
Queue clq =new concurrentLinkedQueue<>(original);
```

```
for (Integer w: clq)
```

```
clq.remove(w);
```

PAGE 223

```
import java.nio.file.Files;
```

```
import java.nio.file.Path;
```

```
import java.nio.file.Paths;
```

```
import java.io.IOException;
```

```
import java.util.stream.Stream;
```

```
public class CourseFilter {
```

```
    public static void main(String[] args) {
```

```
        Path filePath = Paths.get("course.txt");
```

```
        try (Stream<String> lines = Files.lines(filePath)) {
```

```
            lines.filter(s -> s.contains("Java"))
```

```
                .forEach(System.out::println);
```

```
        } catch (IOException ex) {
```

```
        System.out.format("File IO Exception is thrown: %s", ex);
    }
}
}
```

Which code fragment at line 1 prints the lines that contain Java from the course.txt file?

```
List<String> lines1 =
Files.readAllLines (filePath) .contains ("Java");
for (String line : lines2) [
System.out.println(line); 1

Files.lines (filePath) . filter(s ->
s.contains ("Java") ) . forEach(System.out :: println);
```

```
List<string> lines2 = Files.readAllLines(filePath) .filter(s ->
s.contains ("Java"));
for (String line : lines2) (
System.out.println(line);
System.out.println(Files.readString(filePath).contains("Java"));

Files.lines (filePath) .map(s ->
s.contains("Java") ) . forEach (System.out :: println);
```

```
import java.util.function.*;  
public class TripleThis {  
    public static void main(String[] args) {  
        Function<Integer, Integer> tripler = x -> { return x * 3; };  
        TripleThis.printValue(tripler, 4);  
    }  
    public static <T extends Number> void printValue(Function<T, Integer> f, T num) {  
        System.out.println(f.apply(num));  
    }  
}
```

Which two replacements remove this compiler warning and prints 12?

Replace line 9 with Function tripler = x -> { return x * 3; } 

Replace line 12 with public static void printValue (Function f, Integer num) {

Replace line 12 with public static void printValue (Function f, int num) (

Replace line 12 with public static void printValue (Function f, T num) (

Replace line 9 with Function tripler = x -> (return x * 3;)

```
public abstract class ExampleAbstractClass {  
    static String origin = "Abstract Class";  
    abstract void exampleMethod (String first, String second);
```

Please click next to go to the next test page. Click Summary to see which questions you have answered.

Before submitting the test. Click Finish Test if you are ready to submit your test.

Time Remaining 00:15:04

10, Given:

```
public interface ExampleInterface {  
    static String origin = "Interface";  
    void exampleMethod (String first);
```

```
public class ExampleClass extends ExampleAbstractClass implements  
ExampleInterface {  
    public void exampleMethod (String first) {}  
    public void exampleMethod (String first, String second) {}  
    public static void main (String[] args) {  
        ExampleInterface theInstance = new ExampleClass ();  
        //line n1
```

Which two, when inserted at line n1 independently, will cause a compilation error?

the Instance. exampleMethod ("France");

((Exampleclass) theInstance) .exampleMethod("Japan", "Mexico");

```
theInstance.exampleMethod (ExampleAbstractClass.origin,  
ExampleInterface.origin);
```

```
theInstance. exampleMethod (ExampleAbstractClass.origin);
```

```
theInstance.exampleMethod (origin);
```

PAGE 226

Test: 819 - Java SE 11 Developer

Answer the question(s) on this page, and click Next to go to the next test page. Click Summary to see

answer before submitting the test. Click Finish Test if you are ready to submit your test.

Time Remaining 00:15:47

 1. Given the code fragment:

```
public class FileHandler{  
    public static void main (String[] args) {  
        try (FileInputStream in = new FileInputStream("foo.txt")) {}  
        catch (FileNotFoundException e) {}  
    }  
}
```

Which two actions, independently, enable the code to compile?

Inserting:

```
finally ( in.close () : )
```

Replacing the catch block with:

```
catch (Exception e) ( )
```

Adding throws FileNotFoundException declaration at the main () method

Adding throws IOException declaration at the main () method

Replacing the catch block with:

```
catch (FileNotFoundException | Exception e) ( )
```

```
finally ( in.close(); )
```

Replacing the catch block with:

```
catch (Exception | IOException e) ( )
```

autoclosable th

IOException

PAGE 227

Time Remaining 00:14:51

18. Given:



```
public class Employee {
    private String locality;
    private int salary;
    // the setter and getter methods go here
```

```
.atream ()  
/* insert code here */
```

and

8. List roater - createEmployeeList () /
9. double average - roster
- 10.
- 11.
12. System.out.println (average);

Which code fragment inserted on line 11 prints the average salary of all employees from the Bay Area?

```
.filter (e -> e.getLocality ().equals ("Bay Area"))  
.average (Employee::getSalary)  
.getAsDouble ();  
  
.filter (e -> e.getLocality ().equals ("Bay Arca"))  
.filter (s -> s.getSalary ())  
.average ()  
.getAsDouble ();
```

```
.filter (e -> e.getLocality () .equals ("Bay Area") )  
.mapToInt (Employee: :getSalary)  
. average ()  
.getAsDouble ();  
  
.collect (Collectors.groupingBy (Employee :: getLocality,  
Collectors.averagingDouble (Employee: :getSalary) ) );
```

PAGE 223

Time Remaining 00:14:27

24. Given:

```
Path p1 - Paths.get("/scratch/exam/topsecret/anawers");  
Path p2 - Paths.get("/scratch/exam/answers/temp.txt");  
Path p3 - Patha.get("/acratch/answers/topsecret");
```

Which two statements print .. \ .. \answers\topsecret?

```
System.out.print (p2.relativize(p1));
```

```
System.out.print (p1.relativize(p2));
```

```
System.out.print (p3.relativize(p2));
```

```
System.out.print (p2.relativize(p3));
```

```
System.out.print (p1.relativize (p3));
```

```
System.out.print (p3.relativize(p1));
```

PAGE 230

26. Assuming the bodies are correct, which will result in a compilation error?

```
public <T> BiFunction<T, . Bookens> predicate (Function<T, T> transform) ( ... )
```

```
class FOo<T> {  
    public Foo(BiFunction<T, T, T> op) ( ... )
```

```
    public BiFunction<String, String, String> foo;
```

```
    public void foo (BiFunction<int, int, boolean> predicate) [ ... ]
```

PAGE 231

List original = new

```
ArrayList<>(Arrays.asList(1,2,3,4,5));
```

Which two code fragments remove the elements from the original list?

```
List s1= Collections.synchronizedList (original);
```

```
for (Integer w : s1)
```

```
sl.remove (w) ;
```

```
List cwa = new CopyOnWriteArrayList<>(original);  
for (Integer w : cwa)  
    cwa.remove (w);
```

```
List al = new ArrayList<>(original);  
for (Integer w : al)  
    al.remove (w);
```

```
Queue clq = new ConcurrentLinkedQueue<>(original);  
for (Integer w : clq)  
    clq.remove (w);
```



PAGE 232

46. Given the content from the couraes. txt file:

Given the code fragment:

```
Path filePath - Patha.get("course.txt");  
try (  
    /* line 1 */  
) catch (IOException ex) {  
    System.out.format ("File IO Exception is thrown.", ex);
```

Which code fragment at line 1 prints the lines that contain Java from the course.txt file?

```
List<String> lines1 =  
    Files.readAllLines (filepath) -contains ("Java");  
    for (String line : lines2)  
        System.out.println (line) : )  
  
    Files.lines (filePath).filter (s ->  
        s.contains ("Java") ) .forEach (System.out :: println);
```

```
List<String> lines2 = Files.readAllLines(filePath).filter (s->  
    s.contains ("Java") );  
    for (string line : lines2) {  
        System.out.println(line);  
  
        System.out.println(Files.readString (filePath).contains("Java"));  
  
    Files.lines (filePath) .map(s ->  
        s.contains ("Java") ) .forEach(System.out: :println);
```

PAGE 233

19. Given the code fragment:

```
Stream<Integer> data - IntStream. range (1, 10000) . boxed () /  
    Integer sum - data.mapToInt(a -> a) .sum(); //line 1
```

Which two code fragments, independently, replace line 1 to implement the equivalent reduce operation?

OptionalInt value - data.mapToInt(a -> a).parallel().reduce(0, (a, h) ->a+b);

Integer sum = value.getAsInt (#

Integer sum = data.map(a -> a).reduce((a, b)-> a+b);

OptionalInt value = data.mapToInt(a -> a).parallel().reduce((a,b) ->a+b);

Integer sum = value.getAsInt ();

int s = 0;

Integer sum = data.map(a -> a).reduce (0, (a-> a + s));

Integer sum =data.mapToInt (a -> a).reduce(0, (a,b)->a+b);

PAGE 234

19. Given the code fragment:

Stream<Integer> data - IntStream.range(1, 10000).boxed () :

Integer aum - data.mapToInt(a -> a) .sum() ; //line 1

Which two code fragments, independently, replace line 1 to implement the equivalent reduce operation?

```
OptionalInt value = data.mapToInt(a -> a).parallel().reduce(0, (a, b) -> a + b);
```

```
Integer sum = value.getAsInt(); Integer num = data.map(a -> a).reduce(0, (a, b) -> a+b);
```

```
OptionalInt value = data.mapToInt(a -> a).parallel().reduce((a, b) -> a + b);
```

```
Integer sum = value.getAsInt();
```

```
int a = 0;
```

```
Integer num = data.map(a -> a).reduce(0, (a -> a + 5));
```

```
Integer sum = data.mapToInt(a -> a).reduce(0, (a, b) -> a + b);
```

PAGE 235



1. Given the code fragment:

```
public class FileHandler {  
    public static void main (String[] args) {  
        try (FileInputStream in = new FileInputStream("foo.txt")) {}  
        catch (FileNotFoundException e) {}  
    }  
}
```

Which two actions, independently, enable the code to compile?

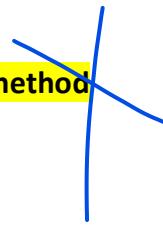
Inserting:

```
finally { in.close(); }
```

Replacing the catch block with:

```
catch (Exception e) {}
```

Adding throws FileNotFoundException declaration at the main () method



Adding throws IOException declaration at the main () method

Replacing the catch block with:

```
catch (FileNotFoundException | Exception e) {}  
finally { in.close (); }
```

Replacing the catch block with:

```
catch (Exception | IOException e) ()
```

Question 46:

Given the content from the courses.txt file:

123:java:1
124:MySQL:2
125:Java Server Pages: 3

Given the code fragment:

```
Path filePath = Paths.get("course.txt");  
try {  
    /*line 1*/  
}  
catch (IOException ex) {  
    System.out.format("File IO Exception is thrown.", ex);  
}
```

Which code fragment at line 1 prints the lines that contain Java from the course.txt file?

- A. List<String> lines1 = Files.readAllLines (filePath).contains("Java");
 - a. for (String line : lines2) {
 - i. System.out.println (line);

- b. }
- B. Files.lines(filePath).filter(s ->
 a. s.contains ("Java")) . forEach (System.out::println);
- C. List<String> lines2 = Files.readAllLines(filePath).filter(s -> s.contains("Java"));
 for (String line : lines2) {
 a. System.out.println(line);
 }
- D. System.out.println(Files.readString(filePath).contains("Java"));
- E. Files.lines (filePath).map (s ->
 a. s.contains ("Java")).forEach (System.out :: println);



Question 29:

```
package com.ltim.dumps;
import java.util.function.*;

public class TripleThis {
    public static void main(String[] args) {
        Function tripler = x -> {
            return (Integer) x * 3;
        };
        TripleThis.printValue(tripler, 4);
    }

    public static void printValue (Function f, T num) {
        System.out.println(f.apply(num));
    }
}
```

Given TripleThis.java:

Compiling TripleThis.java gives this compiler warning:

Note: TripleThis.java uses unchecked or unsafe operations.

Which two replacements remove this compiler warning and prints 12?

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

Question 48:

Given:

```
List original = new ArrayList<>(Arrays.asList(1,2,3,4,5));
```

Which two code fragments remove the elements from the original list?

- A. List sl = Collections.synchronizedList(original);
for (Integer w : sl)
sl.remove (w);
- B. List cwa = new CopyOnWriteArrayList<>(original);
for (Integer w : cwa)
cwa. remove (w); //cannot convert from object to integer so have to write
List<Integer>**
- C. List al = new ArrayList<>(original);
for (Integer w : al)
al.remove (w);
- D. Queue clq = new ConcurrentLinkedQueue<>(original);
for(Integer w : clq)
clq. remove (w) ;**

Question 35:

```

public class Foo {
    public void foo(Collection arg) {
        System.out.println("Bonjour le monde!");
    }
}
And
public class Bar extends Foo {
    public void foo(List arg) {
        System.out.println("Hello world!");
    }
}

public static void main(String... args) {
    List<String> li = new ArrayList<>();
    Collection<String> co = li;
    Bar b = new Bar();
    b.foo(li);
    b.foo(co);
}
}

```

What is the output?

A. Bonjour le monde!
Hello world!

B. Hello world!
Bonjour le monde!

C. Hello world!
Hello world!

D. Bonjour le monde!
Bonjour le monde!

Question 31:



```

public class Main {
    static String prefix = "Mondial:";
    private String name = "domainmodel";

    public static String getName() {
        return new Main().name;
    }
}

```

```
public static void main(String[] args) {  
    Main m = new Main();  
    System.out.println(/* Insert code here */);  
}  
}
```

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

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

Question 24:

Given:

```
Path p1 = Paths.get("/scratch/exam/topsecret/answers");  
Path p2 = Paths.get ("/scratch/exam/answers/temp.txt");  
Path p3 = Paths.get ("/scratch/answers/topsecret");
```

Which two statements print ..\..\..\answers\topsecret?

- A. System.out.print (p2.relativize(p1));
- B. System.out.print (p1.relativize(p2));
- C. System.out.print (p3.relativize(p2));
- D. System.out.print(p2.relativize(p3));
- E. System.out.print(p1.relativize(p3));
- F. System.out.print (p3.relativize(p1));



Question 33:

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

What is the result?

A. The compilation fails.

final variable
must be
initialized in all
the constructor of
the class.

B. 1.99,2.99,0

C. 1.99,2.99,0.0

D. 1.99,2.99



Question 34:

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

Which "for" loop produces the same output?

- A.

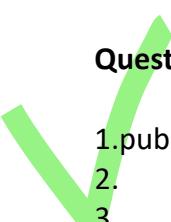
```
for(a; a < 10; a++)
System.out.print (a);
```
- B.

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

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

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

Question 32:



```
1. public class Test {
2.     private static class Greet {
3.         private void print() {
4.             System.out.println("Hello World");
5.         }
6.     }
7.
8.     public static void main(String[] args) {
9.         Test.Greet i = new Greet();
10.        i.print();
11.    }
12.}
```

What is the result?

- A. Hello World
- B. The compilation fails at line 2.
- C. The compilation fails at line 9.
- D. The compilation fails at line 8.

 **Question 39:**

Given the code fragment:

Integer i = 11;

Which two statements compile?

- A. Double a = i;
- B. Double c = (Double) i;
- C. Double b = Double.valueOf (i);
- D. double d = i;
- E. double e = Double.parseDouble (i);

 **Question 38:**

Which three initialization statements are correct?

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

G. boolean false = (4 != 4);



Question 40:

```
public class Tester {  
    public static void main(String[] args) {  
        float x = 2, y = 4, z = 4;  
        float a = y / x, b = y / z;  
        if (a > b) {  
            System.out.println(a + b);  
        }  
    }  
}
```

What is the result?

A. An exception is thrown at runtime.

B. 1.0

C. 2.0

D. 3.0

E. The program prints nothing.



Question 37:

Given:

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

```
public void foo(List arg) {  
    System.out.println("Olá Mundo!");  
}  
}  
and  
Foo f1 = new Foo ();  
Foo f2 = new Bar () ;  
Bar b1 = new Bar () ;  
Collection<String> c=new ArrayList<>();
```

Which three are true?

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

Question 36:

Given:

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

and

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

```
// line 1  
}  
}  
}
```

Which two lines can replace line 1 so that the Y class compiles?

- A. super.set (List map)
- B. super.set (map.values());
- C. set (map.values());
- D. map.forEach((k, v) -> set(v));
- E. set (map) This one is also correct , if the question asks for three options.

Question 44:

Given:

```
public class Tester {  
    public static void main(String args[]) {  
        String s = "10";  
        try {  
            int x = 0;  
            x = Integer.parseInt(s, 2); // line 1  
            System.out.println("% is " + x);  
        } catch (NumberFormatException e) {  
            System.out.println("Error parsing value of " + x); // line 2  
        }  
    }  
}
```

What is the result?

- A. Error parsing value 0
- B. The compilation fails due to an error in line 2.
- C. X is 10.
- D. X is 2.

E. The compilation falls due to an error in line 1.

 **Question 45:**

Given:

```
class Scope {  
    static int myint = 666;  
  
    public static void main(String[] args) {  
        int myint = myint compilation error  
        System.out.println(myint);  
    }  
}
```

Which is true?

- A. It prints 666.
- B. The code compiles and runs successfully but with a wrong answer (i.e., a bug).
- C. **The code does not compile successfully.**
- D. Code compiles but throws a runtime exception when run.

local variable myint inside the main method "hides" or shadows the static myint. The int myint on the left-hand side is the new local variable being declared.

The myint on the right-hand side refers to the same local variable, not the static myint. Since local variables are not automatically initialized in Java, the compiler cannot resolve the reference, and it throws the error: "variable myint might not have been initialized."

Question: 5

Given the content from the courses.txt file:

123:Java:1

124:MySQL:2

125:Java Server Pages:3

 **Given the code fragment:**

```
Path filePath = Paths.get("course.txt");
```

```
try {
```

```
/* line 1 */  
} catch (IOException ex) {  
    System.out.format("File 10 Exception is thrown.", ex);  
}
```

Which code fragment at line 1 prints the lines that contain Java from the course.

A. List<String> lines2 = Files.readAllLines(filePath).filter(s ->

```
    s.contains("Java"));  
    for(String line : lines2) {  
        System.out.println(line);  
    }
```

B. List<String> lines1 =

```
    Files.readAllLines (filePath).contains ("Java");  
    for (String line : lines2) {  
        System.out.println(line);  
    }
```

C. System.out.println(Files.readString(filePath).contains("Java"));

D. Files.lines (filePath).filter(s ->

```
    s.contains("Java")).forEach (System.out::println);
```

E. Files.lines(filePath).map(s ->

```
    s.contains("Java")).forEach(System.out::println);
```



Question: 26

Given the code fragment:

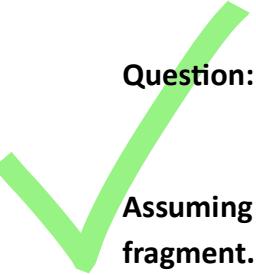
```
module citizen {  
    exports com.name to greeting;  
}
```

and

```
module greeting {  
}
```

Which statement is true?

- A. public members in the com.name package are accessible to all modules.
- B. All members in the com. name package are accessible only to the greeting module.
- C. Inserting "requires citizen;" at greeting's module-info.java, enables com.name members accessible to the greeting module.
- D. All members of com.name are accessible only to the citizen and greeting modules.
- E. public members in the com. name package are accessible only to the greeting module.



Question: 31

Assuming the Book class contains the get Price() method and given the code fragment.

```
List books = List.of(new Book ("Goodbye to Childhood", 19),  
                    new Book ("Farewell to the Land", 35),  
                    new Book ("City Life", 17));
```

Which two statements, independently, print the books with price less than 20?

A. Stream bookStream = books.stream();

```
bookStream.map((Book a) -> a.getPrice() <20)  
.forEach (System.out::println);
```

Wrong syntax and no need to use map in this case

B. books.stream().filter(a -> a.getPrice() <20)

```
.forEach (System.out::println);
```

C. Stream bookStream = books.stream();

```
bookStream.filter(a-> ((Book)a).getPrice() <20)  
.forEach (System.out::println);
```

D. Stream bookStream = books.stream();

```
bookStream.filter(a -> a.get.Price() < 20)      we must cast a to Book  
.peek(System.out::println);
```

E. Stream bookStream = books.stream();

```
bookStream.peek(a -> a.getPrice() <20)  
.forEach (System.out::println);
```

Here also, we must cast a to Book.
The peek() method is intended for debugging or intermediate operations in a stream pipeline. It does not filter or transform the stream

 Question: 25

Which two statements are correct about modules in Java?

A. By default, modules can access each other as long as they run in the same folder.

- B. **java.base** exports all of the Java platforms core packages.
- C. **module-info.java** cannot be empty.
- D. **module** must be declared in **module-info.java** file.
- E. **module-info.java** can be placed in any folder inside module-path.

Question: 16

Given the code fragment:

```
/* line n1 */  
A() {  
    super ("The Mandatory Criteria Yet to Meet");  
}  
}  
15. public class TestCE {  
16.     public static void main(String[] args) throws A {  
17.         int a 10, b = 13;  
18.         try {  
19.             if (ab) {  
20.                 throw new A();  
21.             }  
22.         }  
23.         catch (Exception e) { System.out.println(e); }  
24.         System.out.println("Continue...");  
25.     }  
26. }
```

Throwable is the parent class for both Exception and Error. It does not automatically fall into the Exception hierarchy.

Since catch (Exception e) does not handle Throwable directly unless explicitly mentioned, this will not be caught by the catch block.

The program will terminate when A is thrown at line 20. This satisfies the requirement.

You must define the A exception class. The program execution must be terminated if true and an A exception is thrown at line 20.

Which code fragment at line n1 defines A as per the requirement?

- A. class A extends ArithmeticException {
- B. class A extends Exception {
- C. class A extends Throwable {
- D. class A extends RuntimeException {

Question: 42

Given the code fragment from Box.java:

```
public class Box implements Serializable {  
    private int boxId;  
    private String size;  
    private List items;  
}
```

Given the code fragment from Item.java:

```
public class Item {  
    private int id;  
    private String name;  
}
```

Given the information:

The classes Box and Item are encapsulated with getters and setters methods.

The classes Box and Item contains required constructors source code.

and the code fragment:

```
public static void main (String[] args) throws IOException {  
    List items1 = new ArrayList<>();  
    items1.add(new Item(1, "Pen"));  
    items1.add(new Item(2, "Ruler"));  
    Box b1 = new Box (123, "a", items1);  
    try ( FileOutputStream fout = new FileOutputStream("boxser.txt")) {  
        ObjectOutputStream out = new ObjectOutputStream(fout);)  
        out.writeObject(b1);  
        out.flush ();  
        out.close ();  
    } catch (Exception e) {  
        System.out.println ("Unable to Serialize");  
    }  
}
```

Which action serializes the b1 object?

- A. Override readObject () and writeObject () methods in the Book class.
- B. Handle NotSerializableException in the try clause or throw in the main () method definiton.
- C. Add serialVersionUID to the Box and Item class.
- D. Implement the Serializable interface in the Item class.**
- E. Remove out.flush () method invocation.

Question: 44

Given:

```
package test.t1;  
public class A {
```

```
    public int x = 42;  
    protected A() {}      // line 1
```

```
}
```

and

```
package test.t2;  
import test.t1.*;  
public class B extends A {
```

```
    int x = 17;          // line 2  
    public B() { super(); } // line 3
```

```
}
```

and

```
package test;  
import test.t1.*;  
import test.t2.*;  
public class Tester {
```

```
    public static void main(String[] args) {  
        A obj = new B();           //line 4  
        System.out.println(obj.x); // line 5  
    }  
}
```

What is the result?

- A. The compilation fails due to an error in line 1.
- B. 42**
- C. The compilation fails due to an error in line 3.
- D. The compilation fails due to an error in line 5.**
- E. The compilation fails due to an error in line 2.
- F. The compilation fails due to an error in line 4.



Question: 30

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

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

What does the `transitive` modifier mean?

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



Question: 9

Given the data of the EMP table:

ID	NAME	DEPT
101	SMITH	HR
102	JONES	ENG
103	WEAVER	HR

Assuming that jdbcURL, username, and password are declared and initialised.

```
try (Connection conn = DriverManager.getConnection (jdbcURL, username,
password)

    PreparedStatement query = conn.prepareStatement ("SELECT ID, NAME FROM
EMP ?");

    PreparedStatement update = conn.prepareStatement ("INSERT INTO RECRUITING
VALUES (?,?)"))

{
    query.setString(1, "HR");

    ResultSet rs = query.executeQuery();

    while (rs.next()) {

        update.setObject(1, rs.getObject (1, Integer.class), JDBCType.INTEGER);

        update.setObject (2, rs.getObject (2, String.class), JDBCType.VARCHAR);

        update.execute();

    }

}
```

Which two happen upon execution?

- A. Memory leaks because **Connection**, **PreparedStatement**, and **ResultSet** are not closed.
- B. Three SQL statements are executed.**
- C. A **SQLException** is thrown because the **ResultSet** is not closed.
- D. Three **PreparedStatement** objects are created.
- ~~E. Two SQL statements are executed.~~
- F. Two **PreparedStatement** objects are created.**

 **Question: 50 : ALREADY DONE**

Question: 20

Given:

```
public enum Season {  
    WINTER ('W'), SPRING('s'), SUMMER('h'), FALL('f');  
    char c;  
    private Season (char c) {  
        this.c = c;  
    }  
}
```

and the code fragment:

```
public static void main(String[] args) {  
    Season[] sA = Season.values();  
    // Line n1  
}
```

Which three code fragments, at line n1, prints SPRING?

- A. System.out.println(sA[0]);
- B. System.out.println(sA[1]);**
- C. System.out.println(Season.valueOf("SERING"));**
- D. System.out.println(Season.valueOf('s'));
- E. System.out.println(Season.valueOf("SPRING").ordinal());
- F. System.out.println(season.SPRING);**
- G. System.out.println(Season.values(1));

Question: 47

Given:



```
public class Tester {  
    public static void main(String[] args) {  
        int x = 0, y = 6;  
        for(; x<y; x++, y--) { // line 1  
            if (x%2 == 0) {  
                continue;  
            }  
            System.out.println(x+"-"+y);  
        }  
    }  
}
```

What is the result?

- A. 1-5
- B. 0-6
- 2-4
- C. The compilation fails due to an error in line 1.
- D. 1-5
- 2-4
- E. 2-4
- F. 0-6
- G. 0-6
- 1-5
- 2-4

 Question: 15

Given the declaration:

```
@Target ({TYPE, METHOD})
```

```
@interface Resource {}
```

This is a class declaration. Since @Resource is allowed on types (with @Target(TYPE)), it can be applied here.

```
/* Loc1 */ class Manager extends /* Loc2 */ Person {
```

```
    /* Loc3 */ Manager() {...} This is a constructor. Since @Resource is not targeted at constructors (CONSTRUCTOR target is missing), it cannot be applied here.
```

```
    /* Loc4 */ String getDepartmentName() {...}
```

```
    /* Loc5 */ String departmentName; This is a method. Since @Resource includes @Target (METHOD), it can be applied here.
```

```
}
```

This is a class being extended in the extends clause. Annotations cannot be applied to the class being extended, as it's not a type declaration or method.

In which two locations is it legal to apply the @Resource annotation?

A.Loc2

B.Loc5



C.Loc3

D.Loc1

E.Loc4

Question: 12

Given:

```
public class Main {  
    public static void main(String[] args) {  
        Thread t1 = new Thread(new MyThread());  
        Thread t2 = new Thread(new MyThread());  
        Thread t3 = new Thread (new MyThread());  
  
        t1.start();  
        t2.run()  
        t3.start();  
  
        t1.start();  
    }  
}
```

```
class MyThread implements Runnable {  
    public void run() {  
        System.out.println("Running.");  
    }  
}
```

Which one is correct?

- A. Three threads are created.
- B. Four threads are created.
- C. The compilation falls.
- D. An **IllegalThreadStateException** is thrown at runtime.



Question: 13

Given:

```
public class A {  
    int a = 0;  
    int b = 0;  
    int c = 0;  
    public void foo(int i) {  
        a += b*i;  
        c -= b*i;  
    }  
    public void setB(int i) {  
        b = i;  
    }
```

```
    }  
}
```

Which makes class A thread safe?

- A. Make **foo** synchronized.
- B. **Class A** is thread safe.
- C. Make **setB** synchronized.
- D. Make **A** synchronized.
- E. Make **foo** and **setB** synchronized.



Question 1:

```
package test.t1;

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

and

```
package test.t2;
import test.t1.*;

public class B extends A{
    int x = 17;           //line 2
    public B () { super(); } //line 3
}
```

and

```
package tester;
import test.t1.*;
import test.t2.*;

public class Tester {
```

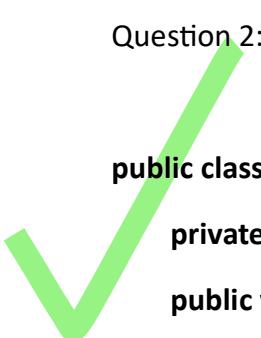
```
public static void main(String[] args) {  
    A obj = new B();           //line 4  
    System.out.println(obj.x); //line 5  
}  
}
```

What is the result?

1. The compilation fails due to an error in line 3.
2. The compilation fails due to an error in line 5.
3. The compilation fails due to an error in line 3.
4. 17
5. 42
6. The compilation fails due to an error in line 4.

Answer: 5. 42

Question 2:



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

```
 }  
 }
```

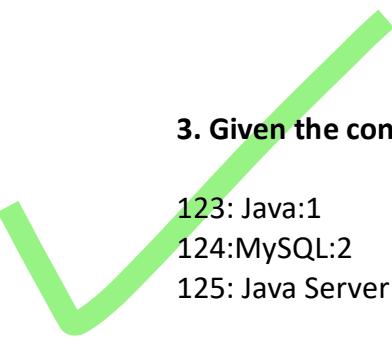
and

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

What is the result?

1. Mr. Green
2. Green
3. Mr. Blue
4. An exception is thrown at runtime.

Answer: 3. Mr.Blue



3. Given the content from the courses.txt file:

123: Java:1
124:MySQL:2
125: Java Server Pages: 3

Given the code fragment:

```
Path filePath = Paths.get("course.txt");
try {
/* line 1 */
} catch (IOException ex) {
System.out.format ("File 10 Exception is thrown.", ex);
}
```

Which code fragment at line 1 prints the lines that contain Java from the course.txt file?

- a) Files.lines (filePath).map(s ->
s.contains("Java")).forEach (System.out::println);
- b) List<String> lines2 = Files.readAllLines (filePath).filter(s ->
s.contains("Java"));
for (String line: lines2) {
System.out.println(line);
}
- c) List<String> lines1 =
Files.readAllLines (filePath).contains ("Java");
for (String line: lines2) {
System.out.println(line); }
- d) **Files.lines (filePath).filter(s ->
s.contains("Java")).forEach (System.out::println);**
- e) System.out.println (Files.readString (filePath) .contains ("Java"));



4. Given:

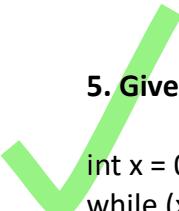
```
public class Person {  
    private String name = "Green";  
    public void setName (String name) {  
        String title = "Mr. ";  
        this.name = title + name;  
    }  
    public String toString() {  
        return name;  
    }  
}
```

And

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

What is the result?

- a) Mr. Green
- b) Green
- c) Mr. Blue
- d) An exception is thrown at runtime.



5. Given the code fragment:

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

Which "for" loop produces the same output?

- a)

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

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

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

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

7. A company has an existing Java app that includes two Java 8 jar files, sales-8.10.jar and clients- 10.2.jar.

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

They have decided to modularize clients-10.2.jar.

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

Options:

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

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

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

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

Ans: a.

Explanation: This option exports the com.company.clients package, making it available for other modules, such as sales-8.10.jar, to use. The other options either incorrectly specify dependencies or do not export the required packages.



8. Given:

```
public class A {  
    int a = 0;  
    int b = 0;  
    int c = 0;  
    public void foo (int i) {  
        a += b * i;  
        c -= b * i;  
    }  
    public void setB (int i) {  
        b = i;  
    }  
}
```

Which makes class A thread safe?

a. Make foo and setB synchronized.

- b. Class A is thread safe.
- c. Make A synchronized.
- d. Make foo synchronized.
- e. Make setB synchronized.

Ans: a.

Explanation: This ensures that both methods foo and setB are synchronized, preventing concurrent access to the shared state and thus making the class thread-safe.



9. Given:

```
5. IntStream str = IntStream.of(2, 3, 4);  
6. IntFunction<Integer> func = x -> y -> x * y;  
7. str.map(func.apply(10)).forEach(System.out::println);
```

Which action will enable the code to compile?

- a. Replace line 6 with Function<UnaryOperator> func = x -> y -> x * y;
- b. Replace line 6 with IntFunction<UnaryOperator> func= x -> y -> x * y;

- c. Replace line 6 with `IntFunction<IntUnaryOperator> func = x -> y -> x * y`
- d. Replace line 6 with `BiFunction<Integer> func = x -> y -> x * y;`

Ans: c

Explanation: This option correctly uses `IntFunction` to produce an `IntUnaryOperator`, which matches the lambda expression's type.

(PAGE NO. 10)

Q.45

45. Given:

```
public class Main {  
    public static void main (String[] args) {  
        List<Player> players = List.of (new Player ("Scott", 115),  
                                         new Player ("John", 70),  
                                         new Player ("Jelly", 105) );  
  
        double average = // line 1  
  
        System.out.println ("The average is: " + average);  
    }  
}  
  
class Player {  
    public String name;  
    public int score;  
    public Player (String name, int score) {  
        this.name = name;  
        this.score = score;  
    }  
}
```

You want to calculate the average of the Player's score.

Which statement inserted on line 1 will accomplish this?

- a) players.stream() .mapToInt (a -> a.score) .average().orElse (0.0);
- b) players.stream() .mapToDouble (a -> a.score) .average () ;
- c) players.stream() .map(a -> a.score).average ();
- d) players.stream().average ().orElse (0.0);

(PAGE NO.11)

Q.8 Given:

```
class MyType<T> {  
    private T value;  
    public T getValue () {  
        return value;  
    }  
    public void setValue (T value) {  
        this. value = value;  
    }  
}
```

and

```
public class Test {  
    public static void main (String ... args) {  
        MyType<String> strType = new MyType<>() ;  
        MyType <? extends Number> type = new MyType<>() ;  
        strType.setValue ("test") ;  
        type. setValue (null) ;  
        System.out.println (strType.getValue() + ":" + type.getValue())  
    }  
}
```

What is the result ?

- a) test: null
- b) test: 0
- c) An Exception is thrown at runtime.
- d) The compilation fails.

e) null: null

(PAGE NO.12)

Q49 – Given:

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

Which is true?

- a) The compilation fails due to an error in line 4.
- b) The compilation fails due to an error in line 10.
- c) The compilation fails due to an error in line 9.
- d) The compilation fails due to an error in line 6.
- e) The compilation succeeds.
- f) The compilation fails due to an error in line 7.
- g) The compilation fails due to an error in line 2.

(PAGE NO.13)

Q50 - Given:

```
public enum Status (
```

BRONZE (5) , SILVER (10) , GOLD (15);

```
private int rate;  
  
private Status (int rate) {  
  
    this.rate = rate;  
  
}  
  
public int getRate() { return rate; }  
  
public Status addStatus (int rate) {  
  
    return new Status (20) ;  
  
}  
  
}
```

and

```
public class Test {  
  
    public static void main (String [] args) {  
  
        Status silver = Status. SILVER;  
  
        System.out.println(silver+silver.getRate ()) ;  
  
        Status platinum = Status. addStatus (20) ;  
  
        System.out.println(platinum+platinum.getRate()) ;  
  
    }  
}
```

What is the result?

- a) SILVER10 platinum20
- b) SILVER10 PLATINUM20
- c) The compilation fails.
- d) An exception is thrown at runtime.
- e) SILVER10 20

QUESTION :-

Given the code fragment:

Integer i = 11;

Which two statements compile? (Choose two.)

- A. Double b = Double.valueOf(i);
- B. double e = Double.parseDouble(i);
- C. Double a = i;
- D. double d = i;
- E. Double c = (Double) i;

QUESTION :-

Given the code fragment :

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

17. }

18. }

What is the result?

A. The compilation fails at line 16.

B. 10

C. The compilation fails at line 9.

D. The compilation fails at line 13.

E. 1

17)

Given the code fragment:

```
Integer i = 11;
```

Which two statements compile?

- a) Double b = Double.valueOf(i);
- b) **double e = Double.parseDouble(i);**
- c) Double a = j;
- d) **double d = i;**
- e) Double c = (Double) i;

18)

Your organization provides a cloud server to your customer to run their Java code. You are reviewing the changes for the next release and you see this change in one of the config files:

```
JAVA_OPTS="$JAVA_OPTS-Xms8g-Xmx8g -noverify"
```

```
JAVA_OPTS="$JAVA_OPTS -Xms8g"
```

Which is correct?

1. You reject the change because -Xms8g -Xmx8g uses too much system memory.
2. You accept the change because -noverify is a standard option that has been supported since Java 1.0.
- 3. You reject the change because -noverify is a critical security risk.**
4. You accept the change because -noverify is necessary for your code to run with the latest version of Java.

19)

```
interface MyInterface1 {  
    public int method() throws Exception;  
    private void pMethod() { /* an implementation of pMethod */ }
```

```
}

interface MyInterface2 {
    public static void sMethod() { /* an implementation of sMethod */ }
    public boolean equals();
}

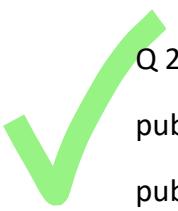
interface MyInterface3 {
    public void method();
    public void method(String str);
}

interface MyInterface4 {
    public void dMethod() { /* an implementation of dMethod */ }
    public void method();
}

interface MyInterface5 {
    public static void sMethod();
    public void method(String str);
}
```

Which two interfaces can be used in lambda expressions?

- MyInterface1
- **MyInterface2**
- MyInterface3
- MyInterface4
- **MyInterface5**



Q 20

```
public class Tester {  
    public static void main (String[] args) {  
        float x=2, y= 4, z= 4;  
        float a = y / x, b = y / z;  
        if (a > b) {  
            System.out.println (a + b);  
        }  
    }  
}
```

What is the result?

The program prints nothing.

An exception is thrown at runtime.

3.0

2.0

1.0



Q 21

```
package pratice;
```

```
public interface Converter {  
    public static final double POUNDS_PER_KILOGRAM = 2.20462;
```

```
    public double tare();
```

```
    public double net();
```

```
public default double gross() {  
    return tare() + net();  
}  
  
public default double tare(String units) {  
    return toUnit(tare(), units);  
}  
  
public default double net(String units) {  
    return toUnit(net(), units);  
}  
  
public default double gross(String units) {  
    return toUnit(gross(), units);  
}  
  
private static double toUnit(double kilograms, String unit) {  
    switch (unit) {  
        case "KILO":  
            return kilograms;  
        case "POUND":  
            return kilograms * POUNDS_PER_KILOGRAM;  
        default:  
            throw new IllegalArgumentException();  
    }  
}
```

```
 }  
 }
```

Which is true?

Line 3 is the first line to cause a compilation error.

Line 2 is the first line to cause a compilation error.

Line 1 is the first line to cause a compilation error.

It compiles without errors.

Q22

Given:

```
Path p1 = Paths.get("/scratch/exam/topsecret/answers");
```

```
Path p2 = Paths.get("/scratch/exam/answers/temp.txt");
```

```
Path p3 = Paths.get("/scratch/answers/topsecret");
```

Which two statements print ..\..\..\answers\topsecret? (Choose two.)

- A. System.out.print(p3.relativize(p1));
- B. System.out.print(p2.relativize(p3));
- C. System.out.print(p1.relativize(p3));
- D. System.out.print(p3.relativize(p2));
- E. System.out.print(p1.relativize(p2));
- F. System.out.print(p2.relativize(p1));



23) Given the code fragment:

```
public class Main {  
  
    public static void main(String[] args) {  
  
        List<String> fruits = List.of ("banana", "orange", "apple", "lemon"); Stream<String> s1  
        =fruits.stream();  
  
        Stream<String> s2 =a1.peek (i-> System.out.print (i+" "));  
  
        System.out.println("----");  
  
        Stream<String> s3 = s2.sorted();  
  
        Stream s4 = s3.peek (i-> System.out.print (i+" ")); System.out.println("----");  
  
        String strFruits = s4.collect(Collectors.joining (","));  
  
    }  
  
}
```

What is the output?

a) -----

b) banana orange apple lemon

apple banana lemon orange

c) -----

banana orange apple lemon apple banana lemon orange

d) -----

banana orange apple lemon
----- apple banana lemon orange

e) banana orange apple lemon apple banana lemon orange

24)

Given:

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

You want to make the reduction operation parallelized.

Which two modifications will accomplish this?

a) Replace line 1 with int sum = numbers.parallel().stream().reduce(0, (n, m) -> n + m);

b) Replace line 1 with int sum = numbers.parallelStream().reduce(0, (n, m) -> n+m);

c) Replace line 1 with int sum = numbers.stream().iterate(0, a -> a+1).reduce(0, (n, m) -> n + m);

d) Replace line 1 with int sum = numbers.stream().parallel ().reduce (0, (n, m) -> n + m);

e) Replace line 1 with int sum = numbers.stream().flatMap (a -> a) . reduce (0, (n, m) -> n + m);

25)

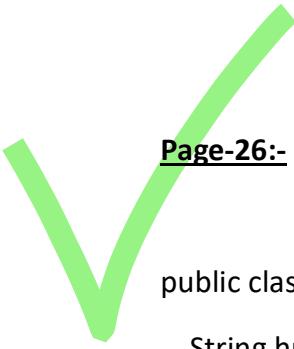
Given the code fragment:

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

What is the result?

- a) The compilation fails at line 16.
- b) 10

- c) The compilation fails at line 9.
- d) The compilation fails at line 13.
- e) 1



Page-26:-

```
public class Color {  
    String hue;  
    int value;  
  
    public Color(String hue, int value) {  
        this.hue = hue;  
        this.value = value;  
    }  
  
    public String toString() {  
        return this.hue + ":" + this.value;  
    }  
  
    public static void main(String[] args) {  
        List<Color> clrs = List.of(  
            new Color("Red", 100),  
            new Color("Yellow", 50),  
            new Color("Red", 75),  
            new Color("Yellow", 75)  
        );  
  
        Comparator<Color> hueSrtr = (h1, h2) -> h1.hue.compareTo(h2.hue);  
        Comparator<Color> valueSrtr = (h1, h2) -> {  
            if (h1.value >= h2.value) {  
                return 1;  
            } else {  
                return -1;  
            }  
        };  
    }  
}
```

```
        return 1;  
    } else {  
        return -1;  
    }  
};  
  
clrs.sort(hueSrtr.thenComparing(valueSrtr));  
System.out.println(clrs);  
}  
}
```

What is the result?

- A. [Red:75, Red:100, Yellow:50, Yellow:75]
- B. [Yellow:75, Yellow:50, Red:100, Red:75]
- C. [Yellow:50, Yellow:75, Red:75, Red:100]
- D. An Exception is thrown at runtime.
- E. [Red:100, Red:75, Yellow:75, Yellow:50]



Page-27:-

Given:

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

and

```
public class Bar extends Foo {  
    public void foo(List arg) {  
        System.out.println("Hello world!");  
    }  
    public static void main(String... args) {  
        List<String> li = new ArrayList<>();  
        Collection<String> co = li;  
        Bar b = new Bar();  
        b.foo(li);  
        b.foo(co);  
    }  
}
```

What is the output?

- A. Bonjour le monde!
Hello world!

- B. Hello world!
Bonjour le monde!

- C. Bonjour le monde!
Bonjour le monde !

- D. Hello world!
Hello world!



Page-28:-

Given:

```
Path p1 = Paths.get("/scratch/exam/topsecret/answers");
Path p2 = Paths.get("/scratch/exam/answers/temp.txt");
Path p3 = Paths.get("/scratch/answers/topsecret");
```

Which two statements print ..\..\..\answers\topsecret?

- A. System.out.print(p1.relativize(p3));
- B. System.out.print(p3.relativize(p1));
- C. System.out.print(p2.relativize(p3));
- D. System.out.print(p1.relativize(p2));
- E. System.out.print(p2.relativize(p1));
- F. System.out.print(p3.relativize(p2));



Question 29).- (Doubt in this question)

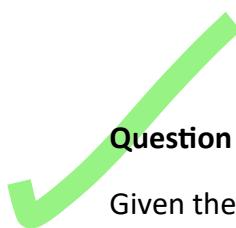
```
module citizen {  
    exports com.name to greeting;  
}
```

AND

```
module greeting {  
}
```

Which statement is true?

- a.) public members in the com.name package are accessible only to the greeting module.
- b.) All members of com.name are accessible only to the citizen and greeting modules.
- c.) All members in the com.name package are accessible only to the greeting module.
- d.) Inserting "requires citizen;" at greeting's module-info.java, enables com.name members accessible to the greeting module.
- e.) public members in the com.name package are accessible to all modules.



Question 30).-

Given the code fragment:

```
StringBuilder txt1 = new StringBuilder("PPQRRRSTT");  
int i=0;  
a:  
while (i < txt1.length()) {  
    char x = txt1.charAt(i);  
    int j = 0;
```

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

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

```
continue a;  
i--;  
j--;  
continue b;  
break b;  
break a;
```

 Question 31).-

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

You want to make the reduction operation parallelized.

Which two modifications will accomplish this?

Replace line 1 with int sum = numbers.parallel().stream().reduce(0, (n, m) -> n + m);

Replace line 1 with int sum = numbers.parallelStream().reduce(0, (n, m)-> n+m);

Replace line 1 with int sum = numbers.stream().iterate(0, a -> a+1).reduce(0, (n, m) -> n+m);

Replace line 1 with int sum = numbers.stream().parallel ().reduce(0, (n, m) -> n + m);

Replace line 1 with int sum = numbers.stream().flatMap(a -> a).reduce(0, (n, m) -> n +m);

```

public class App {
// line 1
    public static void main(String[] args){
        new App().new Greeting().greet ("Joe");
    }
}

```



```

public class App {
    class Greeting {
        private void greet(String name) {
            System.out.println("Hello " + name);
        }
    }
    public static void main(String[] args) {
        new App().new Greeting().greet("Joe");
    }
}

```

<terminated> App [Java App]
Hello Joe

Which code fragment added to line 1 enables the code to compile and print Hello Joe ?

1. class Greeting {
private void greet (String name) {
System.out.println("Hello " + name);
}
}

The method greet cannot be declared static; static methods can only be declared in a static or top level type. i.e, we have to make class Greeting as static, in order to run this static greet method.

2. class Greeting {
public static void greet (String s) { System.out.println("Hello "+ s);
}
}

3. static class Greeting
public void greet (String name) {System.out.println("Hello" + name);
}
}

4. interface Greeting {
public default void greet (String name) { System.out.println(greet+name);
}
}

```

1 package com.lti.experiments;
2
3 public class Tester{
4     private int x;
5     private static int y;
6
7     public static void main(String[] args) {
8         Tester t1 = new Tester();
9         t1.x = 2;
10        Tester.y = 3;
11        Tester t2 = new Tester();
12        t2.x = 4;
13        t2.y = 5;
14        System.out.println(t1.x + ", " + t1.y);
15        System.out.println(t2.x + ", " + Tester.y);
16        System.out.println(t2.x + ", " + t1.y);
17    }
18 }
19
20

```



```

1 package com.lti.experiments;
2
3 public class Tester{
4     private int x;
5     private static int y;
6
7     public static void main(String[] args) {
8         Tester t1 = new Tester();
9         t1.x = 2;
10        Tester.y = 3;
11        Tester t2 = new Tester();
12        t2.x = 4;
13        t2.y = 5;
14        System.out.println(t1.x + ", " + t1.y);
15        System.out.println(t2.x + ", " + Tester.y);
16        System.out.println(t2.x + ", " + t1.y);
17    }
18 }
19
20

```

<terminated>
2, 5
4, 5
4, 5



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

ANSWER :

2, 5
4, 5
4, 5



```
import java.util.function.*;  
public class TripleThis {  
    public static void main(String[] args) {  
        Function tripler = x -> {return (Integer) x * 3;};  
        TripleThis.printValue(tripler, 4);  
    }  
    public static void printValue (Function f, T num) {  
        System.out.println(f.apply(num));  
    }  
}
```

Given TripleThis.java

Compiling TripleThis.jav gives this compiler warning:

Note: Triplethis.java uses unchacked or unsafe operation

A. Replace line 9 with Function tripler = x -> { return x * 3; }

B. Replace line 12 with public static void printValue (Function f, int num) {

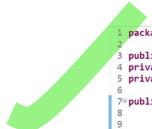
C. Replace line 12 with public static void printValue (Function f, T num) {

D. Replace line 12 with public static void printValue (Function f, Integer num) {

E. Replace line 9 with Function tripler = x -> { return x*3}

F. Replace line 9 with Function tripler = x-> { return (Integer)x*3}

35))



```
1 package com.liti.experiments;
2 public class Tester{
3     private int x;
4     private static int y;
5
6     public static void main(String[] args) {
7         Tester t1 = new Tester();
8         t1.x = 2;
9         Tester.y = 3;
10        Tester t2 = new Tester();
11        t2.x = 4;
12        t2.y = 5;
13        System.out.println(t1.x + ", " + t1.y);
14        System.out.println(t2.x + ", " + Tester.y);
15        System.out.println(t2.x + ", " + t1.y);
16    }
17}
18}
```

public class Tester{
private int x;
private static int y;

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

ANAWER :

2, 5

4, 5

4, 5

36))



```
public class ResourceTest {  
    public static void main(String[] args) {
```

```
final MyResource res1 = new MyResource();
MyResource res2 = new MyResource();
try(res1; res2) {
    // do something
} catch (Exception e) {}
}
static class MyResource implements AutoCloseable {
    public void close() throws Exception { }
}
```

//Which statement is true?
//The code fails to compile as res2 should be declared as final.
//The code fails to compile as MyResource must implement Closeable.
//The code compiles successfully.
//The code fails to compile as try-with-resource needs a variable declaration such as
MyResource r1 = res1; MyResource r2 = res2;

37))



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

What is the result?

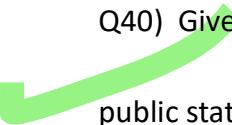
hat at store 1

at once 1

An IndexOutOfBoundsException is thrown at runtime.

at once 0

hat at store 4

 Q40) Given the code fragment

```
public static void main(String[] args) {  
    var lst = List.of(1, 2.0f, "4.0");  
    for (var c : lst) {  
        System.out.print("> " + c);  
    }  
    System.out.println();  
    lst.add(2, 3);      // line n1  
    for (int c = 0; c < lst.size(); c++) {  
        display(lst.get(c));  
    }  
}
```

public static void display(var c) { // line n2

```
    System.out.print("> " + c);  
}
```

What is the result?

Options:

A)

> 1> 2.0> 4.0
> 1> 2.0> 4.0

B) (Ans)

A compile time error occurs at line n2.

C)

> 1> 2.0> 3> 4.0

D)

An exception is thrown at line n1.

=====

Q39) Given the code fragment

Locale locale = Locale.US;
// line 1
double currency = 1_00.00;
System.out.println(formatter.format(currency));

You need to display the value of currency as \$100.00.

Which code inserted in line 1 will accomplish this?

Options:

A)

NumberFormat formatter = NumberFormat.getInstance(locale);

B) (Answer)

NumberFormat formatter = NumberFormat.getCurrencyInstance(locale);

C)

NumberFormat formatter = NumberFormat.getCurrency(locale);

D)

NumberFormat formatter = NumberFormat.getInstance(locale).getCurrency();

=====

Q38) Given :

public class Main {

```
public static void main(String[] args) {  
    Thread t1 = new Thread(new MyThread());  
    Thread t2 = new Thread(new MyThread());  
    Thread t3 = new Thread(new MyThread());  
  
    t1.start();  
    t2.run();  
    t3.start();  
  
    t1.start();  
}  
}
```

```
class MyThread implements Runnable {  
    public void run() {  
        System.out.println("Running.");  
    }  
}
```

Which one is correct?

Options

A)

Four threads are created.

B)

The compilation fails.

C)(Answer)

An `IllegalThreadStateException` is thrown at runtime.

D)

Three threads are created.

Page 41



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

what would be the output?

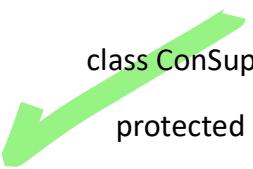
An exception is thrown at run time

3

5

4(Answer)

Page 42



```
class ConSuper {  
    protected ConSuper() {  
        this(2); // Call the parameterized constructor  
        System.out.print("3");  
    }  
}
```

```
protected ConSuper(int a) {  
    System.out.print(a);  
}  
}  
  
public class ConSub extends ConSuper {  
    public ConSub() {  
        this(4); // Call the parameterized constructor of ConSub  
        System.out.print("1");  
    }  
  
    public ConSub(int a) {  
        super(); // Call the no-argument constructor of ConSuper  
        System.out.print(a);  
    }  
  
    public static void main(String[] args) {  
        new ConSub(4); // Create an instance of ConSub with the parameterized  
        constructor  
    }  
}
```

What will be the result:

2134

234

2431

214



Page 43

```
class Employee {  
    private String name;  
    private String neighborhood;  
    private int salary;  
    // Constructors, getters, and setters go here.  
}
```

```
List<Employee> roster = new ArrayList<>();  
  
// Example employees  
roster.add(new Employee("Alice", "Downtown", 40));  
|  
roster.add(new Employee("Bob", "Uptown", 25));  
roster.add(new Employee("Charlie", "Midtown", 50));
```

```
Predicate<Employee> p = e -> e.getSalary() > 30;  
Function<Employee, Optional<String>> f = e ->  
Optional.ofNullable(e.getNeighborhood());
```

Select the correct Option:

1. Map<String, List<Employee>> r2 = roster.stream()
.filter(p)
.collect(Collectors.groupingBy(f, Employee::getNeighborhood));

2. Map<Optional<String>, List<Employee>> r5 = roster.stream()

```
.collect(Collectors.groupingBy(  
    Employee::getNeighborhood,  
    Collectors.filtering(p, Collectors.toList())  
)); // Valid
```

3. Map<Optional<String>, List<Employee>> r4 = roster.stream()

```
.collect(Collectors.groupingBy(  
    f,  
    Collectors.filtering(p, Collectors.toList())  
)); // Valid
```

4. Map<String, List<Employee>> r1 = roster.stream()

```
.collect(Collectors.groupingBy(  
    Employee::getNeighborhood,  
    Collectors.filtering(p, Collectors.toList())  
)); // Valid
```

5. Map<Optional<String>, List<Employee>> r3 = roster.stream()

```
.filter(p)  
.collect(Collectors.groupingBy(p)); // Invalid
```

Page 44

Question



```
public class Test {  
    public static void main(String... args) {  
        int number = 0;  
        Predicate<Integer> p = a -> a % 2 != 0;  
        // line 1  
        if(/* code needed here */){  
            System.out.println(number + " is odd.");  
        } else {  
            System.out.println(number + " is even.");  
        }  
    }  
}
```

Which statement on line 1 enables the code to compile?

Options

```
if(p.test(number)) {  
  
if(p.accept(number)) {  
  
if(p.get(number)) {  
  
if(p.apply(number)) {
```

Page 45



```
public class Main {  
    public static void main(String[] args) {  
        Thread t1 = new Thread(new MyThread());  
        Thread t2 = new Thread(new MyThread());  
        Thread t3 = new Thread(new MyThread());  
  
        t1.start();  
        t2.run();
```

```
t3.start();
```

```

        t1.start();
    }
}

class MyThread implements Runnable {
    public void run() {
        System.out.println("Running.");
    }
}

```

Which one is correct?

Options

Four threads are created.

The compilation fails.

An `IllegalThreadStateException` is thrown at runtime.

Three threads are created.

Page 46

Given `TripleThis.java`



```

import java.util.function.*;
public class TripleThis {
    public static void main(String[] args) {
        Function tripler = x -> (Integer) (Integer) x * 3;
        TripleThis.printValue(tripler);
    }
    public static void printValue(Function f, T num) {
        System.out.println(f.apply(num));
    }
}

```

Compiling `TripleThis.java` gives this compiler warning :

Note : TripleThis.java uses unchecked or unsafe operations.

Which two replacements remove this compiler warning and prints 12 ?

Options

Replace line 9 with Function tripler = x -> { return x * 3; }

Replace line 12 with public static void printValue(Function f, int num) {

Replace line 12 with public static void printValue(Function f, T num) {

Replace line 9 with Function tripler = x -> { return x * 3; }

Replace line 9 with Function tripler = x -> { return (Integer)x * 3; } 

```
public class ResourceTest {  
    public static void main(String[] args) {  
        final MyResource res1 =new MyResource();  
        MyResource res2= new MyResource();  
        try(res1;res2) {  
            // do something  
        } catch (Exception e) {}  
    }  
  
    static class MyResource implements AutoCloseable {  
        public void close() throws Exception {}  
    }  
}
```

Which statement is true?

- The code fails to compile as res2 should be declared as final.
- The code fails to compile as MyResource must implement Closeable.
- **The code compiles successfully.**
- **The code fails to compile as try-with-resource needs a variable declaration such as MyResource r1 =res1; MyResource r2=res2;**

PAGE NO:48



```
public interface Converter {  
    public static final double POUNDS_PER_KILOGRAM = 2.204627;      // LINE 1  
    public double tare();  
    public double net();  
    public default double gross() {           // LINE 2  
        return tare() + net();  
    }  
    public default double tare(String units) {  
        return toUnit(tare(), units);  
    }  
    public default double net(String units) {  
        return toUnit(net(), units);  
    }  
    public default double gross(String units) {  
        return toUnit(gross(), units);  
    }  
}  
  
private static double toUnit(double kilograms, String unit) {      // LINE 3  
    switch (unit) {  
        case "KILO":return kilograms;  
        case "POUND":return kilograms * POUNDS_PER_KILOGRAM;  
        default:throw new IllegalArgumentException();  
    }  
}
```

Which is true?

- Line 3 is the first line to cause a compilation error,
- Line 2 is the first line to cause a compilation error.

- Line 1 is the first line to cause a compilation error.
- It compiles without errors.

PAGE NO:49



```

import java.util.List;
public class Main {
    public static void main(String[] args) {
        var lst = List.of(1, 2.0f, "4.0");
        for (var c : lst) {
            System.out.print(">" + c);
        }
        System.out.println();
        lst.add(2, 3); //n1
    }
}

for (int c = 0; c < lst.size(); c++) {
    display(lst.get(c));
}
}

public static void display(var c) { //n2
    System.out.print(">" + c);
}
}

```

var cannot be used as parameter. If in place of var we write Object then output will be >1>2.0>4.0 followed by an exception

What is the result?

- >1> 2.0> 4.0

> 1> 2.0> 4.0

- A compile time error occurs at line n2.
- >1> 2.0> 4.0 >
1> 2.0> 3> 4.0
- An exception is thrown at line n1.



39. Given the code fragment:

```
Locale locale = Locale.US;
```

```
//line1
```

```
Double currency = 1_00.00;
```

```
System.out.println(formatter.format(currency));
```

You want to display the value if currency as \$100.00.

Which code inserted on line 1 will accomplish this?

a. NumberFormat formatter = NumberFormat.getInstance(locale);

b. NumberFormat formatter = NumberFormat.getCurrencyInstance (locale);

c. NumberFormat formatter = NumberFormat.getCurrency (locale);

d. NumberFormat formatter =

```
NumberFormat.getInstance(locale).getCurrency();
```



40. Given the code fragment:

```
public static void main (String[] args){  
    Var lst = List.of(1,2.0f,"4.0");  
    for(var c:lst){  
        System.out.println(">" +c);  
    }  
    System.out.println();  
    lst.add(2,3); //line n1  
    for(int c=0; c<lst.size();c++) {  
        display(lst.get(c )));  
    }  
}  
public static void display(var c){ //line n2  
    System.out.println(">" +c);  
}
```

What is the result?

a. >1>2.0>4.0

> 1>2.0>4.0

b. A compile time error occurs at line n2.

c. > 1> 2.0> 4.0

> 1> 2.0> 3> 4.0

d. An exception is thrown at line n1.

41. Which declaration of an annotation type is legal?

- a. @interface Author {
 String name();
 String date;
}
- b. @interface Author {
 String name();
 String date default "";
}
- c. @interface Author extends Serializable {
 String name() default "";
 String date();
}
- d. @interface Author {
 String name() default "";
 String date();
}
- e. @interface Author {
 String name() default null;
 String date();
}



Qn:56

```
import java.io.FileNotFoundException;

public class ExSuper extends Exception {
private final int eCode;
public ExSuper (int eCode, Throwable cause) {
super (cause);
this.eCode = eCode;
}

public ExSuper (int eCode, String msg, Throwable cause) {
super (msg, cause);
this.eCode = eCode;
}

public String getMessage() {
    return this.eCode+": "+ super.getMessage()+"-"+this.getCause ().getMessage();
}
}

public class ExSub extends ExSuper {
public ExSub (int eCode, String msg, Throwable cause){ super(eCode, msg, cause); }
}

try {
String param1 = "Oracle";
if (param1.equalsIgnoreCase ("oracle")) {
throw new ExSub (9001, "APPLICATION ERROR-9001", new
FileNotFoundException("MyFile.txt"));
}
throw new ExSuper (9001, new FileNotFoundException ("MyFile.txt")); // Line 1 .
}
catch (ExSuper ex) {
System.out.println(ex.getMessage());
}
```

Ans:

9001: APPLICATION ERROR-9001-MyFile.txt

Qn:57



```
import java.util.List;

public class Two {

    public static void main(String[] args) {
        var lst = List.of(1, 2.0f, "4.0");
        for (var c : lst) {
            System.out.print("> " + c);
        }

        System.out.println();
        lst.add(2, 3); //line n1
        for(int c = 0; c < lst.size(); c++)
        {
            display(lst.get(c));

        }
    }

    public static void display(var c){ //line n2
        System.out.print("> " + c );
    }
}
```

What is the result?

1. > 1> 2.0> 4.0
 > 1> 2.0> 4.0
2. A compile time error occurs at line n2.
3. > 1> 2.0> 4.0
 > 1> 2.0> 3> 4.0
4. An exception is thrown at line n1.

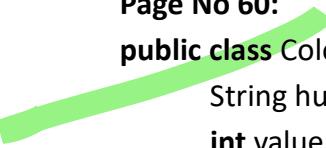


Page No 59 :

```
public class ResourceTest {  
    public static void main (String [] args) {  
        final MyResource res1 = new MyResource ();  
        MyResource res2 = new MyResource ();  
        try (res1; res2) {  
            // do something  
        } catch (Exception e) {}  
    }  
    static class MyResource implements AutoCloseable {  
        public void close () throws Exception {}  
    }  
}
```

Which statement is true?

- A. The code fails to compile as res2 should be declared as final.
- B. The code fails to compile as MyResource must implement Closeable.
- C. **The code compiles successfully.**
- D. The code fails to compile as try-with-resource needs a variable declaration such as MyResource r1 = res1; MyResource r2 = res2;



Page No 60:

```
public class Color {  
    String hue;  
    int value;  
    public Color (String hue, int value) {  
        this.hue = hue;  
        this.value = value;  
    }  
    public String toString() {  
        return this.hue+":"+this.value;  
    }
```

```

public static void main(String[] args) {

    List clrs = List.of (new Color ("Red", 100),
    new Color ("Yellow", 50),
    new Color ("Red", 75),
    new Color ("Yellow", 75));

    Comparator hueSrtr = ( h1, h2 ) -> h1.hue.compareTo(h2.hue);
    Comparator valueSrtr = (h1, h2) -> {

        if (h1.value >= h2.value) {

            return 1;

        }

        else {

            return -1;

        }

    };

    clrs.sort (hueSrtr.thenComparing (valueSrtr));

    System.out.println (clrs);

}

}

```

What is the result?

- A. [Red: 75, Red: 100, Yellow: 50, Yellow: 75]
- B. [Yellow: 75, Yellow:50, Red: 100, Red:75]
- C. [Yellow:50, Yellow: 75, Red: 75, Red:100]
- D. An Exception is thrown at runtime.
- E. [Red: 100, Red: 75, Yellow: 75, Yellow:501]

DUMPs 2

Question: 1

1. When running jdeps, which three ways include dependent nonmodular jar files?

jdeps -- module-path lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar

jdeps -classpath lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar

jdeps -- upgrade-module-path

lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar

jdeps application.jar

jdeps lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar

jdeps -cp lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar

jdeps -- class-path lib/file1.jar:lib/file2.jar:lib/file3.jar application.jar

Question: 2

Your organization provides a cloud server to your customer to run their Java code.

You are reviewing the

changes for the next release and you see this change in one of the config files:

old: JAVA_OPTS="\$JAVA_OPTS -Xms8g -Xmx8g"

new: JAVA_OPTS="\$JAVA_OPTS -Xms8g -Xmx8g -noverify"

Which is correct?

You reject the change because -Xms8g -Xmx8g uses too much system memory.

You reject the change because -noverify is a critical security risk.

You accept the change because -noverify is a standard option that has been supported since

Java 1.0.

You accept the change because -noverify is necessary for your code to run with the latest

version of Java.

Question: 3

Which module defines the foundational APIs of the Java SE Platform?

java.lang

java.se

java.object

java.base

Question: 4

Given the code fragment:

Supplier supplier = () -> "Hello World";

// line 1

Which statement on line 1 is calling the method of the supplier object correctly

System.out.println(supplier.accept());

System.out.println(supplier.get());

System.out.println(supplier.test());

System.out.println(supplier.apply())

Question: 5

Which two statements are true about running code on the class path and the module path?

A modular JAR placed on the -classpath results in an automatic module.

A modular JAR placed on the -classpath results in a named application module.

A non-modular JAR placed on the -classpath results in an unnamed module.

A modular JAR placed on the -- module-path results in a named application module.

A non-modular JAR placed on the -- module-path results in a named application module.

Question: 6

Which declaration of an annotation type is legal?

@interface Author {

String name() default "";

```
String date ();  
@interface Author {  
    String name ();  
    String date default "";  
}  
@interface Author extends Serializable {  
    String name () default "";  
    String date ();  
}  
@interface Author {  
    String name ();  
    String date;  
}  
@interface Author {  
    String name () default null;  
    String date ();
```

Question: 6

Which two statements are correct about modules in Java?

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

Question: 6

Given the data of the BNF table:

ID NAME DEPT

101 SMITH HR

102 JONES ENG

103 WEAVER HR

Assuming that jdbcURL, username, and password are declared and initialised.

```
try (Connection conn = DriverManager.getConnection(jdbcURL, username,
password);

PreparedStatement query conn.prepareStatement ("SELECT ID, NAME FROM EMP
WHERE DEPT=?");

PreparedStatement update = conn.prepareStatement("INSERT INTO RECRUITING (ID,
NAME) VALUES (?, ?)");

query.setString(1, "HR");

ResultSet rs query.executeQuery();

while (rs.next()) {

update.setObject(1, rs.getObject (1, Integer.class), JDBCType. INTEGER);

update.setObject (2, rs.getObject (2, String.class), JDBCType.VARCHAR);

update.execute();

}

}
```

Which two happen upon execution?

- a. A SQLException is thrown because the Resultset is not closed.
- b. Three PreparedStatement objects are created
- c. Three SQL statements are executed

Question: 6



```
public class X {

private Collection collection:

public void set (Collection collection) {

this.collection collection;

}

}

and
```

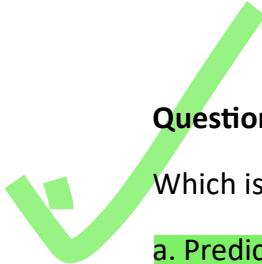
```
public class Y extends X {  
    public void set (Map map) {  
        super.set(map); // line 1  
    }  
}
```

Which two lines can replace line 1 so that the Y class compiles?

- map.forEach((k, v) -> set (v));
- **set (map.values());**
- super.set (List map)
- **super.set(map.values());**
- **set (map)**

Corrected code -

```
import java.util.Collection;  
import java.util.Map;  
  
public class X {  
    private Collection collection;  
  
    public void set(Collection collection) {  
        this.collection = collection;  
    }  
}  
  
public class Y extends X {  
    public void set(Map map) {  
        super.set(map.values()); // Correct replacement for compilation  
        // set(map.values()); // Also correct if you want to call inherited set method  
    }  
}
```



Question : 7

Which is a valid statement?

- a. Predicate testEquality = (var, var y) -> (x.equals(y));
- b. BiPredicate testEquality= (var x,y)-> (x.equals(y));
- c. BiPredicate testEquality= var x, var y-> (x.equals());
- d. BiPredicate testEquality= (var x, Integer y)-> (x.equals(y));

Question: 8

```
public class Person {  
    private String name;  
  
    public Person(String name) {  
        this.name = name;  
    }  
  
    public String toString() {  
        return name;  
    }  
}  
  
public class Tester {  
    static Person p = null;  
  
    public static void main(String[] args) {  
        p = checkPerson(p);  
        System.out.println(p);  
  
        Person p1 = new Person("Joe");  
        p1 = checkPerson(p1);  
        System.out.println(p1);  
    }  
  
    public static Person checkPerson(Person p) {
```

```
if (p == null) {  
    p = new Person("Mary");  
}  
return p;  
}  
}
```

Ans: Mary Joe

Question: 10

Given:

```
public class Resource implements AutoCloseable {  
    public Resource() {  
        System.out.print("A");  
    }  
    @Override  
    public void close() {  
        System.out.print("B");  
    }  
    public void printResource() {  
        System.out.print("C");  
    }  
}
```

and

```
try (Resourcer = new Resource()){  
    r.printResource();  
} finally {  
    System.out.print("D");  
}
```

What is the result?

- a) ADBC
- b) ACD
- c) ACH
- d) ACAD

Correct - ACBD

Question:12

Which module defines the foundational APIs of the Java SE Platform?

- a. java.lang
- b. java.se
- c. **java.base**
- d. java.object

Question: 13

Given the code fragment:

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

What is the result?

- a) **1 3 5 7 9 11**
- b) 13 5 7 9
- c) 2468
- d) 2 4 6 8 10

Question: 14

Given the code fragment:

```
var i = 10;  
var j = 5;  
i += (j * 5+i) / j - 2;
```

`System.out.println (i) ;`

What is the result?

a) 21

b) 15

c) 23

d) 11

e) 5

Question: 15

Given the code fragment:

```
public class Test {  
    private final int x = 1;  
    static final int y;  
    public Test () {  
        System.out.print(x);  
        System.out.print(y);  
    }  
    public static void main (String args[]) {  
        new Test () ;  
    }
```

What is the result?

The compilation fails at line 16.

The compilation fails at line 9.

1

The compilation fails at line 13.

10

Question: 16



Assuming the bodies are correct, which will result in a compilation error?

- a. class Foo{
 - a. public foo(BiFunction op) {...}
 - }
- b. public void foo {BiFunction predicate) {...}}
- c. public BiFunctionct predicate (Function transform){...}
- d. public BiFunction foo;`

Question: 17

Given:

```
public class Main {  
    public static void main(String[] args) {  
        Optional value = createValue();  
        String str= value.orElse ("Duke");  
        System.out.println(str);  
    }  
  
    static Optional createValue() {  
        String s = null;  
        return Optional.ofNullable(s);  
    }  
}
```

What is the output?

- a) A NullPointerException is thrown at run time.
- b) A NoSuchElementException Exception is thrown at run time.
- c) Duke
- d) null

Question: 18

Given,



```
var c = new CopyOnWriteArrayList<>(List.of("1", "2", "3", "4"));

Runnable r = () -> {

try {

Thread.sleep (150) ;

}

catch (InterruptedException e) {

System.out.println (e) ;

}

c.set(3, "four") ;

System.out.print(c + " ");

};

Thread t = new Thread(r);

t.start();

for (var s: c) {

System.out.print(s + " ");

Thread.sleep(100);

}

}
```

What is the output?

- a) 1 2 [1, 2, 3, 4] 3 four
- b) 1 2 [1, 2, 3, four] 3 four
- c) 1 2 [1, 2, 3, four] 3 4
- d) 12 [1, 2, 3, 4] 3 4

Question: 19



Which two are valid statements?

- a) BiPredicate test = (**final** Integer x, var y) -> (x.equals(y)) ;
- b) BiPredicate test = (**final** var x, y) -> (x.equals(y)) ;

- c) BiPredicate test = (var x, **final** var y) -> (x.equals(y)) ;
- d) BiPredicate test = (Integer x, **final** var y) -> (x.equals(y));
- e) BiPredicate test = (Integer x, **final** Integer y) -> (x.equals(y));

Question: 21

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

```
module java.se {
```

```
...
```

```
    requires transitive java.xml;
```

```
...
```

```
}
```

What does the transitive modifier mean?

- a) Only a module that requires the java.se module is permitted to require the java.xml module.
- b) Any module that requires the java.xml module does not need to require the java.se module.
- c) Any module that attempts to require the java.se module actually requires the java.xml module instead.
- d) Any module that requires the java.se module does not need to require the java.xml module.

Question: 28

Given the code fragment:

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

Which would cause s to be AQCD?

- a. s.replace(s.indexOf("B"), s.indexOf("B"), "Q");
- b. s.replace(s.indexOf("B"), s.indexOf("C"), "Q");
- c. s.replace(s.indexOf("A"), s.indexOf("C"), "Q");
- d. s.replace(s.indexOf("A"), s.indexOf("B"), "Q");

Question:29

Which three initialization statements are valid?

- a. var loc = List.of("UK", "US");
- b. var loc = Set.of("UK", "US");
- c. var loc = Set.of ("UK", "US", "UK"); **duplicate values**
- d. var loc = List.of("UK", null, "US"); **null not allowed, it throws exception at runtime**
- e. var loc = Arrays.of("UK", "US", "ES"); **Arrays.of and ArrayList.of**
- f. var loc = ArrayList.of("UK", "US"); **methods do not exists**
- g. var loc = Map.of ("UK", 1, "US", 2);

Question:30

Given:

```
public class Main {  
    public static void main(String... args) {  
        var list new ArrayList( List.of("Coffee", "Cappuccino", "Latte"));  
        list.forEach((item) -> (  
            list.remove(item);  
        ));  
        System.out.println(list);  
    }  
}
```

What is the result?

- a. [Coffee, Cappuccino, Latte]
- b. A java.lang.NullPointerException is thrown.
- c. It prints []
- d. A java.util.concurrentModificationException is thrown.
- e. It prints null

Question:31

```
@Target ((TYPE, METHOD))  
 @interface Resource ()  
 /*Loc1*/ class Manager extends /* Loc2 */ Person {
```

```
/* Loc3 */ Manager() {...}  
/* Loc4 */ String getDepartmentName() {...}  
/* Loc5 */String departmentName;  
}
```

In which two locations is it legal to apply the Resource annotation? (multiple choice)

- a. Loc3
- b. Loc2
- c. Loc 4
- d. Loc 1
- e. Loc 5

Question:33

Given:



```
class Super {  
    final int num // line n1  
    public Super (int num){  
        this.num = num;  
    }  
    final void method(){  
        System.out.println("Output from Super");  
    }  
    class Sub extends Super{  
        int num; // Line n2  
        Sub (short num) { // line n3  
            super (num);  
        }  
        protected void method() { // line 14
```

```
System.out.println("Output from Sub");  
}
```

Which line of code results in a compilation error?

a. line n3

b. line n2

c. line n4 Cannot override final method from Super.

d. line n1

 **Question: 35**

Given:

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

```
b.foo();  
b.bar();  
}  
}  
  
a. Bonjour le monde!
```

Hello world!

Question:36

Given:

```
package a;  
abstract class A {  
    void print() {  
        System.out.print("Base class");  
    }  
}
```

and

```
package a;  
public class B extends A {  
    protected void print() {  
        System.out.print("Derived class");  
    }  
}  
public static void main(String args[]){  
    B b = new B();  
    ((A)b).print();  
}
```

What is the output?

Correct - Derived Class

- a. An exception is thrown at runtime.
- b. Base class
- c. The compilation fails.

Question: 37

Given:

```
Path v1 = Paths.get("./forest/.").resolve(Paths.get("tree.txt"));
```

```
Path v2 = new File("/forest/.water/../tree.txt").toPath();
```

```
System.out.print(Files.isSameFile(v1, v2));
```

```
System.out.print("'" + v1.equals(v2));
```

```
System.out.print("'" + v1.normalize().equals(v2.normalize())));
```

Assuming all referenced paths exist within the file system, what is the result?

true false true

false true true

true true true

false false true

Question: 36

Given the code fragment:

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

What is the result?

-3

2

1

-1

Question: 42

Given the code fragment:

```
ExecutorService es = Executors.newCachedThreadPool(); es.execute(()->
System.out.print ("Ping ")); // line 1

System.out.println(future.get()); // line 2

es.shutdown();
```

Which statement at line 1 will print Ping Pong?

- Future future = es.submit(() -> "Pong");
- Future future = new Callable () {

public String call() throws Exception { return "Pong";

}

.call();
- Future future = es.execute(() -> "Pong");
- Future future = es.invokeAny (new Callable() { public String call() throws
Exception {

return "Pong";

}

});

Question: 39

Given TripleThis.java:

```
6:import java.util.function.Function;
7:public class Tester {
8: public static void main(String[] args) {
```

```
9: Function tripler = x -> { return (Integer) x * 3; };  
10: Tester.printValue(tripler, 4);  
11: }  
12: public static void printValue(Function f, T num) {  
13: System.out.println(f.apply(num));  
14: }  
15: }
```

Compiling TripleThis.java gives this complier warning:

Note: TripleThis.java uses unchecked or unsafe operation.

Which two replacements remove this compiler warning and prints 12?

- a. Replace line 9 with Function tripler = x -> { return x* 3; }
- b. Replace line 12 with public static void printValue (Function f, Integer num)
- c. Replace line 9 with Function tripler = x -> { return x * 3; }
- d. Replace line 9 with Function tripler = x -> { return (Integer) x *3; }
- e. Replace line 12 with public static void printValue (Function f, int num)
- f. Replace line 12 with public static void printValue (Function f, T num)

Question:40

Path v1 = Paths.get("./forest/.").resolve(Paths.get("tree.txt"));

Path v2 = new File(*/forest/.water/..tree.txt).toPath();

System.out.print(Files.isSameFile(v1, v2));

System.out.print(" "+ v1.equals(v2));

System.out.print(""+ v1.normalize().equals(v2.normalize()));

Assuming all referenced paths exist within the file system, what is the result?

- a. true true true
- b. false true true
- c. true false true

d. false false true

Question:43

Your organization provides a cloud server to your customer to run their java code. You are reviewing the changes for the next release and you see this change in one of the config files:

old: JAVA_OPTS = “\$JAVA_OPTS -Xms8g -Xmx8g”

new: JAVA_OPTS= “\$JAVA_OPTS -Xms8g -Xmx8g -noverify”

which is correct?

- a. You reject the change because -noverify is a critical security risk.
- b. You reject the change because Xms8g -Xmx8g uses too much system memory.
- c. You accept the change because -noverify is necessary for your code to run with the latest version of java. is a standard option that has been supported since java 1.0
- d. You accept the change because -noverify is a standard option that has supported since java 1.0.

Question:45

Given these declarations:

```
String eName = "SMITH";
```

```
String empId = "42";
```

And these two code fragments:

Fragment 1:

```
Statement stmt = conn.createStatement();
String sql = "INSERT INT EMP VALUES ('" + eName + "','" + empId + "')";
stmt.executeUpdate(sql);
```

Fragmant 2:

```
String sql1 = "INSERT INTO EMP VALUES (?,?)";
PreparedStatement pstmt = conn.prepareStatement(sql1);
pstmt.setObject(1, eName, JDBCType.VARCHAR);
pstmt.setObject(2, empId, JDBCType.VARCHAR);
```

```
pstmt.executeUpdate();
```

Fragment 1 directly concatenates user inputs (eName and empld) into the SQL query. This approach is vulnerable to SQL injection attacks where malicious input could alter the SQL query and potentially compromise the database.

Which code fragment is preferred and why?

- a. Fragment 1 because it is shorter.
- b. Fragment 1 because it is more performant.
- c. Fragment 2 because it explicitly specifies the SQL types of the column values.
- d. Fragment 2 because it prevents SQL injection.**

Fragment 2 uses PreparedStatement with placeholders (?), ensuring user inputs are treated as data and not executable code. This makes it immune to SQL injection vulnerabilities.

Question:46



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

what is the result?

- a. 2,5 4,5 4,5**
- b. 2,3 4,5 4,5
- c. 2,3 4,5 4,3

d. 2,3 4,3 4,5

Question: 47

Given:

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

And the code fragment:

```
public class Tester {  
    public static void main(String[] args) {  
        var items = List.of(new Item("A", 10), new Item("B", 2), new Item("C", 12), new  
        Item("D", 5), new Item("E", 6));  
  
        //line 1 {  
        System.out.println(items);  
    }  
}
```

Which code fragment at line 1 will accomplish this?

- a. if(items.stream().anyMatch(i -> i.count < 0))
- b. if(items.stream().filter(i -> i.count < 0).findFirst())
- c. if(items.stream().allMatch(i -> i.count < 0))
- d. if(items.stream().filter(i -> i.count < 0).findAny())

Question 47:

Given the code fragment:

```
Stream data IntStream.range (1, 10000).boxed();
```

```
Integer sum = data.mapToInt (a -> a).sum();
```

//line 1

Which two code fragments, independently, replace line 1 to implement the equivalent reduce operation?

Which two code fragments, independently, replace line 1 to implement the equivalent reduce operation?

A) Optionallnt value Integer sum data.mapToInt (a -> a).parallel().reduce(0, (a, b) - a+b); value.getAsInt();

B) Integer sum data.mapToInt (a> a).reduce(0, (a,b)->a+b);

C) Optionallnt value data.mapToInt (a -> a).parallel().reduce((a, b) -> a+b);

Integer sum value.getAsInt();

D) int = 0;

Integer sum data.map(a -> a).reduce(0, (a-> a+ 3));

E) Integer sum data.map(a -> a).reduce((a, b) -> a+b);

Question: 48

Why would you choose to use a peek operation instead of a forEach operation on a Stream?

- a. To process the current item and return a stream.
- b. To remove an item from the end of the stream.
- c. To remove an item from the beginning of the stream.
- d. To process the current item and return word.

Question: 49

```
List list1 = new ArrayList<>();
```

```
list1.add("A");
```

```
list1.add("B");
```

```
List list2 = Collections.unmodifiableList(list1);
```

```
list1.add("C");
```

```
System.out.println(list1);
```

```
System.out.println(list2);
```

What is the result?

a. [A, B, C]

[A, B, C]

b. On line 9, an exception is thrown at run time.

c. [A, B, C]

[A, B]

d. [A, B, C]

Followed by an exception thrown on line 11.

Question:50

```
LocalDate d1 = LocalDate.now();
```

```
d1.plusDays(1);
```

```
d1.minusMonths(2);
```

```
LocalDate d2 = d1.plusWeeks(3);
```

```
d2.minusDays(4);
```

```
d2= null;
```

How many LocalDate objects are created in this example?

a. 4

b. 5

c. 3

d. 2

Question:51

Which two assignments create Locale instances?

locale = "en-USA";

locale = "fr_FR";

locale = Locale.getDefault();

locale = new Locale("en", "GB");--a

locale = Locale.getAvailableLocales();--a

Since list2 is just a view of list1, any changes to list1 (like adding "C") are visible in list2 as well.

An unmodifiable list created with Collections.unmodifiableList() does not create a completely independent, immutable copy of the original list. It simply prevents modifications through the unmodifiable list reference.

Changes made to the original list (list1) are still reflected in the unmodifiable view (list2).

LocalDate.now(): Creates 1 object.

d1.plusDays(1): Creates 1 new object (does not modify d1).

d1.minusMonths(2): Creates another new object (again, d1 remains unchanged).

d1.plusWeeks(3): Creates another new object (assigned to d2).

d2.minusDays(4): Creates yet another new object.

So, that's 5 LocalDate objects in total. The assignment d2 = null does not create or destroy objects—it just changes the reference d2.



Question:52

```
class ConSuper {  
protected ConSuper() {  
this(2);  
System.out.print("3");  
}  
protected ConSuper(int a) {  
System.out.print(a);  
}  
}  
and  
  
public class ConSub extends ConSuper {  
ConSub() {  
this(4);  
System.out.print("1");  
}  
ConSub(int a) {  
System.out.print(a);  
}  
public static void main(String[] args) {  
new ConSub(4);  
}  
}
```

What is the result?

- a)2134
- b)2341
- c)214
- d)234**

Question:53



Which two statements are true about running code on the class path and the module path?

- A modular JAR placed on the -classpath results in a named application module.
- A modular JAR placed on the --module-path results in a named application module.--a

A non-modular JAR placed on the -classpath results in an unnamed module.--a

A modular JAR placed on the -classpath results in an automatic module.

A non-modular JAR placed on the --module-path results in a named application module.

Question:54



```
public class A {  
    int a = 0;  
    int b = 0;  
    int c = 0;  
  
    public void foo(int i) {  
        a += b * i;  
        c -= b * i;  
    }  
  
    public void setB(int i) {  
        b = i;  
    }  
}
```

Which makes class A thread safe?

a.Make A synchronized.

b.Make setB synchronized.

c.Class A is thread safe.

d.Make foo synchronized.

e.Make foo and setB synchronized.

Question:55



```
@Target({TYPE, METHOD})  
@interface Resource {}  
  
/* Loc1 */ class Manager extends /* Loc2 */ Person {  
    /* Loc3 */ Manager() {...}  
    /* Loc4 */ String getDepartmentName() {...}  
    /* Loc5 */ String departmentName;  
}
```

In which two locations is it legal to apply the @Resource annotation?

- Loc5
- Loc4
- Loc3
- Loc1
- Loc2

Question:56



```
public class Tester {  
    public static void main(String args[]) {  
        String a = "10";  
        try {  
            int x = 0;  
            x = Integer.parseInt(a, 2); // line 1  
            System.out.println("X is " + x);  
        } catch (NumberFormatException e) {  
            System.out.println("Error parsing value of " + x); // line 2  
        }  
    }  
}
```

What is the result?

- a) X is 2.
- b) Error parsing value 0
- c) The compilation fails due to an error in line 2.
- d) The compilation fails due to an error in line 1.
- e) X is 10.

Question:57



```
public class Person {  
    private String name = "Green";  
    public void setName(String name) {  
        String title = "Mr. ";  
        this.name = title + name;  
    }  
}
```

```
public String toString() {  
    return name;  
}  
}
```

and

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

What is the result?

- a)Green
- b)An exception is thrown at runtime.
- c)Mr. Green
- d)Mr. Blue**

Question:58

Which three initialization statements are correct?

- **float x = 1f;**
- int[][][] e = {{1,1,1},{2,2,2}};
- boolean false = (4 != 4);
- short sh = (short)'A';**
- int x = 12_34;**
- byte b = 10;char c = b;
- String contact# = "(+2) (999) (232)";

Question:59

Given the code fragment:

```
var i = 10;
```

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

What is the result?

- 1. 11
- 2. 21
- 3. 15**
- 4. 23
- 5. 5

Question:60

Given:

```
int i = 3;  
int j= 25;  
System.out.println( i > 2 ? i > 10 ? i*(j+10): i*j + 5 : i);
```

What is the result?

- 1. 385
- 2. 25
- 3. 3
- 4. The compilation fails.

5. 80

Question:61

Given: class Employee {

String office; } and the code fragment:

- 5. public class HRApp {
- 6. **var employee = new ArrayList();**
- 7. **public var display() {**

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

Which two lines cause compilation errors?

1. line 12
 - 2. line 7**
 3. line 9
 - 4. line 6**
 5. line 8
-

Question:62

Given:

```
1.List fruits = List.of("banana", "orange", "apple", "lemon");  
fruits.sort(new Comparator() {  
    @Override  
    public int compare(String m, String n) {  
        return n.compareTo(m);  
    }  
});
```

Which statement will refactor line 2 to use a lambda expression?

1. fruits.sort((String d, String e)-> {e.compareTo(d);}); return keyword is missing

2. fruits.sort((a, b) -> {return b.compareTo(a);});

3. fruits.sort(o, p->p.compareTo(o));

wrong
syntax

4. fruits.sort((String x, y)-> {return y.compareTo(x)});

Question:63

Given the code fragment:

1. var list = List. of (1,2,3,4,5,6,7,8,9,10);

2. UnaryOperator u = i -> i *2;

3. list. replaceAll (u);

Which can replace line 2?

1. UnaryOperator u = (var i) ->(i *2);

2. UnaryOperator u = i -> { return i * 2};

3. UnaryOperator u = var i -> { return i *2;};

4. UnaryOperator u = (int i) -> i *2;

Question:64

interface AbilityA {
default void action () {
System. out. println ("a action");
}
}

and

interface AbilityB {
void action ();
}
and
public class Test implements AbilityA, AbilityB {

```
public void action () {  
    System.out.println ("ab action");  
}  
  
public static void main (String [] args) {  
    // line 2  
    AbilityB x = new Test ();  
    x.action ();  
}
```

What is the result?

An exception is thrown at run time.

1. a action
2. The compilation fails on line 1.
- 3. ab action**
4. The compilation fails on line 2.

Question:65



```
public interface ExampleInterface {  
    int one = 1;  
    static int two = 2;  
    static final int three = 3;  
}  
  
public class ExampleClass implements ExampleInterface {  
    public static void main (String [] args) {  
        ExampleInterface theInstance = new ExampleClass ();  
    }  
}
```

Which three statements cause a compiler error when inserted at line 1?

```
int a = one++;
int b = two;
int d = ExampleInterface. one;
int h = theInstance. two;
int e = ExampleInterface.two++;
int c = three;
int i = theInstance. three++;
int f = ExampleInterface.three;
int g = theInstance.one;
```

Question:66

✓

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

Which three classes successfully override showFirst () ?

1.

```
public class MainClass implements AdaptorFirst {
    public void showFirst () {
        System.out.println ("first");
    }
}
```
2.

```
public abstract class MainClass implements AdaptorFirst {
    public void showFirst () {
        System.out.println ("first");
    }
}
```
3.

```
public abstract class MainClass implements AdaptorFirst {
```

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

Question:67



```
class Test {  
    void display (int i) {  
        System.out.println ("one") ;  
    }  
    void display (long l) {  
        System.out.println("two") ;  
    }  
    public static void main (String [] args) {  
        new Test().display(0B1010_0101_1001_0110);  
    }  
}
```

What is the result ?

1. The compilation fails.

2. two

3. one

4. A NumberFormatException is thrown at runtime.

Question:68



```
class NoMatchException extends RuntimeException {}

public class Test{

    public static void main (String[] args) {

        try {

            if ("oracle". equals ("ORACLE".toLowerCase())){

                throw new NoMatchException () ;

            }

        }

        catch (NoMatchException | NullPointerException npe) {

            System.out.println ("Exception 1") ;

        }catch (RuntimeException e) {

            System.out.println ("Exception 2") ;

        }catch (Exception e) {

            System.out.println ( "Exception 3") ;

        }finally{

            System.out.println ("Finally Block") ;

        }

    }

}
```

How many lines of text does this program print?

1. four

2. three

3. one

4. two

 **Question:69**

Given:

```
LocalDate d1 = LocalDate.now () ;  
d1.plusDays (1);  
d1 = d1. minusMonths (2);  
LocalDate d2 = d1.plusWeeks (3) ;  
d2.minusDays(4);  
d2 = null;
```

How many LocalDate objects are created in this example?

1. 5

2. 3

3. 2

4. 4



Question:70

```
class MyType {  
    private T value;  
    public T getValue () {  
        return value;  
    }  
    public void setValue (T value) {  
        this. value = value;  
    }  
}  
public class Test {
```

```
public static void main (String ... args) {  
    MyType strType = new MyType<> () ;  
    MyType type = new MyType<> () ;  
    strType.setValue ("test") ;  
    type. setValue (null) ;  
    System.out.println(strType.getValue() + ":" + type.getValue());  
}  
}
```

What is the result ?

1. test:null
2. An Exception Is thrown at runtime.
3. The compilation fails.
4. null:null
5. test : 0

Question: 71

Given:

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

You want to implement the java.io.Serializable interface to the MyPersistenceData class.

Which method should be overridden?

- a.The readExternal method
- b.The readExternal and writeExternal method

c.The writeExternal method

d.Nothing

Question: 72

Given:

```
public enum Season {  
    WINTER ('w'), SPRING ('s'), SUMMER ('h'), FALL('f');  
    char c;  
    private Season (char c) {  
        this.c = c;  
    }  
}
```

and the code fragments:

```
public static void main(String[] args) {  
    Season [] sA = Season.values();  
    // line nl  
}
```

Which three code fragments, at line nl, prints SPRING?

- a.System.out.println (Season.valueOf("SPRING").ordinal());
- b.System.out.println(sA[0]);
- c.System.out.println(sA[1]);
- d.System.out.println (Season. SPRING);**
- e.System.out.println (Season. values (1));
- f.System.out.println (Season.valueof ("SPRING"));**
- g.System.out.println (Season.valueOf('s'));

Question: 73

Which two statements are correct about modules in Java?

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

Question: 73

Which statement is true?

- a.System.exit () Invokes the close () method for the InputStream/OutputStream resources.
- b.PrintWriter outputs characters and automatically flushes the stream.
- c.PrintStream outputs only bytes.
- d.Console.readPassword () method encrypts the text entered.

Question: 82

package Q82;

```
public class StrBldr {  
    // Add your class content here  
    static StringBuilder abl= new StringBuilder("yo ");  
    static StringBuilder sb2= new StringBuilder("hi ");  
    public static void main(String[] args) {  
        sb1=sb1.append(new StrBldr().foo(new StringBuilder("hey")));  
        System.out.println(sb1);  
    }  
    StringBuilder food(StringBuilder builder) {  
        ab1=ab1.append(builder+" oh ");  
    }  
}
```

```
    return sb2;  
}  
}
```

//compile time error occured

- yo hi
- hey oh hi yo
- **yo hi hey oh**
- compile time error
- hey oh yo hi

Question: 84

```
package Q84;
```

```
public class ResourceTest {  
    // Add your class content here  
    public static void main(String[] args) {  
        final MyResource res1 = new MyResource();  
        MyResource res2 = new MyResource();  
        try(res1;res2){  
            //do something  
        }catch(Exception e) {  
            //catch exception  
        }  
    }  
    static class MyResource implements AutoCloseable{  
        @Override
```

```

    public void close() throws Exception {
        // TODO Auto-generated method stub
    }
}

```

- The code fails to compile as MyResource must implement closable
- The code fails to compile as try with resources needs a variable declaration such as (MyResource r1 = res1; MyResource r2 = res2;)
- The code fails to compile as res2 should be declared as final
- The code compiles successfully

Question: 87

```
package Q87;
```

```

final class X {
    // Add your class content here
    public static X createX(double amount) {
        return new X(amount);
    }

    public double amount;
    private X(double amount) {
        this.amount=amount;
    }
    public String toString() {

```

```

        return String.valueOf(amount);
    }

}

public final class Main{
    public static void main(String[] args) {
        X x = X.createX(100.0);
        x.amount=500.0;
        System.out.println(x);
    }
}

```

- 100.0
- a java.lang.IllegalAccessException is thrown
- 500.0
- The compilation fails

Question: Which three classes successfully override showFirst()?

- public abstract class Mainclass implements AdaptorFirst {
- ```

public string showFirst (){
 return "first";
}

```
- public class Mainclass implements AdaptorFirst {
- ```

public void showFirst () {
}

```

```

System.out.println("first");

}

}

• public abstract class MainClass implements AdaptorFirst {

public void showFirst () {

System.out.println("first");

}

• public class MainClass implements AdaptorFirst {

void showFirst();

}

• public abstract class MainClass implements AdaptorFirst {

public abstract void showFirst();

}

• public class MainClass implements AdaptorFirst {

private void showFirst () {

System.out.println("first");

}

}

}

```

Question: 37

Given:

Path v1 =

```

Paths.get("./forest/.").resolve (Paths.get("tree.txt"));

Path v2 = new File("/forest/.water/../tree.txt").toPath(); System.out.print
(Files.isSameFile (v1, v2)); System.out.print (" " +v1.equals (v2));

System.out.print (" " +v1.normalize().equals (v2.normalize ())));

```

Assuming all referenced paths exist within the file system, what is the result?

true false true

false true true

true true true

false false true

36. Given the code fragment:

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

What is the result?

-3

2

1

-1

Question

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

Which "for" loop produces the same output?

- for (int c = 0; ; c++) {

```
    System.out.print (c);  
  
    if (c ==10) {  
  
        break;
```

```
}

}

• for (a; a < 10; a++) {

System.out.print (a);

}

• for (int d = 0; d < 10; ) {

System.out.print (d);

++d;

}

• int b = 0;

for( b < 10; ) {

System.out.print(++b);

}
```

Question

```
public static void main(String... args) {

String filename = "/u01/work" + args[0];

// line n1

// ...

}
```

You want to validate a path name before reading the file. Before validation, all path names should be canonicalized. Which code inserted on line n1 will accomplish this?

```
• Path file = Paths.get(filename);

Path canonicalPath = file.toAbsolutePath().toString();

FileInputStream fis = new FileInputStream(canonicalPath);

• File file = new File(filename).getAbsoluteFile();

FileInputStream fis = new FileInputStream(file);

• File file = new File(filename);
```

```
String canonicalPath = file.getCanonicalPath();  
FileInputStream fis = new FileInputStream(canonicalPath);
```

- Path file = Paths.get(filename);

```
String canonicalPath = file.normalize().toString();
```

```
FileInputStream fis = new FileInputStream(canonicalPath);
```

Question: 98.

```
StringBuilder txt1 = new StringBuilder("PPQRRRSTT");
```

```
int i = 0;
```

```
a: while (i < txt1.length()) {
```

```
    char x = txt1.charAt(i);
```

```
    int j = 0;
```

```
    i++;
```

```
    b:
```

```
    while (j < txt1.length()) {
```

```
        char y = txt1.charAt(j);
```

```
        if (i != j && y == x) {
```

```
            txt1.deleteCharAt(j);
```

```
// line 1
```

```
}
```

```
j++;
```

```
}
```

```
}
```

```
System.out.println(txt1);
```

"Which two statements inserted independently at line 1 enable this code?"

The options provided are:

- break a;

- continue a;

- j--;

- i--;

Question 99

```
package com.ltim.dumps;

class Person {

    private String name;

    public Person(String name) {
        this.name = name;
    }

    @Override
    public String toString() {
        return name;
    }
}

public class Tester {

    static Person p = null;

    public static void main(String[] args) {
        p = checkPerson(p);
        System.out.println(p);
        Person pl = new Person("Joe");
        pl = checkPerson(pl);
        System.out.println(pl);
    }

    public static Person checkPerson(Person p) {
        if (p == null) {
```

```
p = new Person("Mary");
}

return p;
}
```

What is the result?

a) Joe

Joe

b) Mary

Mary

c) null

null

d) Mary

Joe

Question 100

```
package com.ltim.dumps;

class Thing {

    int x, y, z;

    public Thing() {
        this(2, 1);

        System.out.println(x + ", " + y + ", " + z);
    }

    public Thing(int x) {
        this.x = x;

        System.out.println(x + ", " + y + ", " + z);
    }
}
```

```
public Thing(int x, int y) {  
    this(x);  
    this.y = y;  
    System.out.println(x + ", " + y + ", " + z);  
}  
}  
  
public class Tester {  
    public static void main(String[] args) {  
        Thing t1 = new Thing();  
    }  
}
```

Answer:

2,0,0
2,1,0
2,1,0

0,0,0
2,1,0
2,0,0

0,0,0
1,0,0
2,1,0

1,0,0
1,1,0
0,0,0

Question: 101

```
package com.ltim.dumps;  
public class File {
```

```
String filename = "/u01/work" + args [0];  
//line n1...  
// ...  
}
```

You want to validate a path name before the read file. Before validation, all path names should canonicalized.

Which code inserted on line n1 will accomplish this?

a) Path file = Paths.get (filename) ;

```
Path canonicalPath = file.toAbsolutePath ().toString () ;
```

b) FileInputStream fis = new FileInputStream (canonicalPath) ;

```
File file = new File(filename) .getAbsoluteFile () ;
```

c) FileInputStream fis = new FileInputStream(file) ;

```
Path file = Paths.get (filename) ;
```

d) String canonicalPath = file.normalize() .toString () ;

```
FileInputStream fis = new FileInputStream (canonicalPath) ;
```

e) File file = new File (filename) ;

```
String canonicalPath = file.getCanonicalPath() ;
```

```
FileInputStream fis = new FileInputStream(f) ;
```

Question 102

```
public class Tester {  
    public static void main(String[] args) {  
        String s = "hat at store";  
        int x = s.indexOf("at");
```

```
s.substring(x + 3); // This line has no effect on the string 's'  
x = s.indexOf("at");  
System.out.println(s + " " + x);  
}  
}
```

What is the result?

- a) hat at store 1
- b) hat at store 4
- c) An IndexOutOfBoundsException is thrown at runtime.
- d) at once 1
- e) at once 0

Question 103

Given the code fragment:

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

What is the result?

- a) -1
- b) 2
- c) -3
- d) 1

Question 104

```
public class Strbldr {  
    static StringBuilder sb1 = new StringBuilder("yo ");  
    StringBuilder sb2 = new StringBuilder("hi ");  
    public static void main(String[] args) {  
        sb1 = sb1.append(new Strbldr().foo(new StringBuilder("hey")));
```

```
System.out.println(sb1);
}

StringBuilder foo(StringBuilder s) {
    System.out.print(s + " oh " + sb2);
    return new StringBuilder("ey");
}
}
```

What is the result?

- A. yo ey
- B. A compile time error occurs.
- C. oh hi hey
- D. hey oh hi ey
- E. hey oh hi yo ey
- F. hey oh hi

Question 105

```
class Scope {
    static int myint=666;
    public static void main (String[] args) {
        int myint = myint;
        System.out.println(myint) ;
    }
}
```

Which is true?

- A. Code compiles but throws a runtime exception when run.
- B. The code does not compile successfully.

C. It prints 666.

D. The code compiles and runs successfully but with a wrong answer (i.e., a bug).

Corrected Code

```
class Scope {  
    static int myint=666;  
    public static void main (String[] args) {  
        int myint = Scope.myint;  
        System.out.println(myint);  
    }  
}
```

Question 106

```
public class Person {  
    private String name = "Green";  
    public void setName (String name) {  
        String title = "Mr. ";  
        this.name = title + name;  
    }  
}
```

```
public String toString () {  
    return name;  
}  
}
```

and

```
public class Test {  
    public static void main (String args [ ]) {  
        Person p = new Person ();  
        p.setName("Blue");  
    }  
}
```

```
System.out.println(p);  
}  
}
```

What is the result?

- A. An exception is thrown at runtime.
- B. Mr. Green
- C. Mr. Blue
- D. Green

Question 107

```
public enum Season {  
    WINTER('w'), SPRING('s'), SUMMER ('h'), FALL ('f');  
    char c;  
  
    private Season (char c) {  
        this.c=c;  
    }  
}
```

and

```
public static void main (String[] args) {  
    Season[] sA = Season.values() ;  
    //line n1  
}
```

Which three code fragments, at line n1, prints SPRING

- A. System.out.println (Season.values (1));
- B. System.out.println (sA[1]) ;
- C. System.out.println (Season.valueOf("SPRING") .ordinal ()) ;
- D. System.out.println (Season.valueOf("SPRING")) ;

E. System.out.println(Season.valueOf('s'));

F. System.out.println(Season.SPRING);

G. System.out.println(sA[0]);

Question 108

Given code fragment

```
var i = 1;  
  
var result = IntStream.generate(() -> { return i; })  
    .limit(100) .sum();  
  
System.out.println(result);
```

Which statement prints the same value of result?

- A. System.out.println(IntStream.range(0, 99).count());
- B. System.out.println(IntStream.rangeClosed(0, 100).map(x =>x).count())
- C. System.out.println(IntStream.range(1, 100).count());
- D. System.out.println(IntStream.rangeClosed(1, 100).count());

Question 109

```
List list1 = new ArrayList<>();  
  
list1.add("A");  
  
list1.add("B");  
  
List list2= Collections.unmodifiableList(list1);  
  
list1.add("C");  
  
System.out.println(list1);  
  
System.out.println(list2);  
  
a. On line 9, an exception is thrown at run time.  
b. [A, B, C]  
[ A, B, C]  
c.[A, B, C]
```

followed by an exception thrown on line 11.

d.[A, B, C]

[A, B]

Question 110

```
public class practice {  
    public static void main(String[] args) {  
        var list = new ArrayList (  
            List.of("Coffee", "Cappuccino", "Latte") );  
        list.forEach ( (item) -> {  
            list.remove (item) ;  
        });  
        System.out.println (list) ;  
    }  
}
```

What is the result?

a.It prints null

b.A java.util.ConcurrentModificationException is thrown.

c.It prints []

d.[Coffee, Cappuccino, Latte]

e.A java.lang. NullPointerException is thrown.

Question 111

Given the content from the courses. txt file:

123: Java : 1

124 : MySQL : 2

125: Java Server Pages: 3

Given the code fragment:

```

Path filePath = Paths.get("course.txt") ;

try {
    /* line 1 */
} catch (IOException ex) {
    System.out.format("File IO Exception is thrown. ", ex);
}

```

Q Which code fragment at line 1 prints the lines that contain Java from below

- a). System.out.println(Files.readString(filePath) .contains ("Java"));
- b). Files.lines (filePath) .map (s ->
s.contains("Java")) .forEach (System. out: : println) ;
- c). Files.lines (filePath) . filter (s ->
s.contains("Java")) .forEach (System. out : : println) ;
- d). List lines2 = Files.readAllLines (filePath) . filter (s ->
s.contains ("Java")) ;

for (String line : lines2) {

 System.out. println (line) ;

}
- e). List lines1 =

Files. readAllLines (filePath) .contains ("Java") ;

for (String line : lines2) {

 System. out. println (line);

}

Question 112

```

public class A {

    int a = 0;

    int b = 0;

```

```
int c = 0;  
  
public void foo (int i) {  
  
    a += b * i;  
  
    c -= b * i;  
  
}  
  
public void setB (int i) {  
  
    b = 1;  
  
}  
  
}
```

Q Which makes class A thread safe?

- a. Make foo synchronized.
- b. Class A is thread safe.
- c. Make A synchronized.]
- d. Make foo and setB synchronized.e
- e. Make setB synchronized.

Question 113

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

What is the result?

- a. 0

- b. 0
- c. The program prints nothing.
- d. 1
- e. It prints 1 in infinite loop.

Question 114

```
public class Main{  
    public static void main (String[] args) throws IOException {  
        final Reader reader = new FileReader ("File1.txt");  
        try (reader) {  
            reader.read (); //line 1  
        } finally {  
            reader.read (); //line 2  
        }  
        reader.read (); //line 3  
    }  
}
```

If File1.txt does exist, what is the result?

- a.The program executes and prints nothing.
- b.The compilation fails.
- c.A java. io. IOException is thrown on line 2.
- d..A java. io. IOException is thrown on line 1.
- e.A java.io. IOException is thrown on line 3.

Question 115

```
public class Main {  
    public static void main(String[] args) {  
        List fruits = List.of("banana", "orange", "apple", "lemon");  
    }  
}
```

```
Stream s1 = fruits.stream();
Stream s2 = s1.peek(i -> System.out.print(i + " "));
System.out.println(" ----- ");
Stream s3 = s2.sorted();
Stream s4 = s3.peek(i -> System.out.print(i + " "));
System.out.println(" ----- ");
String strFruits = s4.collect(Collectors.joining(","));
}
}
```

What is the output?

a.-----

banana orange apple lemon apple banana lemon orange

b.-----

banana orange apple lemon

----- apple banana lemon orange

c.banana orange apple lemon apple banana lemon orange

d.banana orange apple lemon

apple banana lemon orange

e.-----

Question 116

```
Class.forName (JDBC_DRIVER_CLASS_NAME) ;
```

When is it necessary to execute this statement?

- a. It must be executed before each call to DriverManager to get a Connection using the named JDBC driver.
- b. It is no longer required to execute this method.
- c. It must be executed once and only before the first call to DriverManager to get a Connection using the named JDBC driver.
- d. It must be executed once and before accessing the named JDBC driver in any way.

Question 117

```
String [] words = {"am", "am", "first", "second", "mismatch"} ;  
Map map = Arrays.stream (words)  
.collect(Collectors  
.groupingBy (x -> x, Collectors.counting()) ) ;  
System.out.println(map) ;
```

Taking into account that the order of the elements is unpredictable, what is the output?

{am=2, first=1, mismatch=1, second=2}

{mismatch=1, am=2, first=1, second=1}

{1=mismatch, 2=am}

{mismatch=2, am=2, first=1, second=1}

Question 118

```
public class Test {  
private static class Greet{  
private void print(){  
System.out.println("Hello World");  
}  
}  
public static void main(String[] args) {  
Test.Greet i = new Greet();  
i.print();
```

```
 }  
 }
```

What is the result?

The compilation fails at line 9.

Hello World

The compilation fails at line 8.

The compilation fails at line 2.

Question 119

Which two expressions create a valid Java Path instance?

Paths.get (URI.create("file:///domains/oracle/test.txt"))

new Path ("foo")

Paths. get ("foo")

Paths.getPath ("too")

Path.get (new URI ("file:///domains/oracle/test.txt"))

Question 120

```
public class Tester {  
    static Person p = null;  
    public static void main(String[] args) {  
        p = checkPerson(p);  
        System.out.println(p);  
        Person p1 = new Person("Joe");  
        p1 = checkPerson(p);  
        System.out.println(p1);  
    }  
  
    public static Person checkPerson(Person p) {  
        if(p == null) {  
            p = new Person("Marry");  
        }  
        return p;  
    }  
}
```

What is the result?

Joe

Joe

Mary

Mary

null

null

Marry

Joe

Question 121

24.Given,

```
public class ExSuper extends Exception {  
    private final int eCode;  
    public ExSuper(int eCode, Throwable cause) {  
        super(cause);  
        this.eCode = eCode;  
    }  
    public ExSuper(int eCode, String msg, Throwable cause) {  
        super(msg, cause);  
        this.eCode = eCode;  
    }  
    public String getMessage() {  
        return this.eCode + ":" + super.getMessage() + "-" + this.getCause();  
    }  
}  
  
public class ExSub extends ExSuper {  
    public ExSub(int eCode, String msg, Throwable cause) {
```

```
        super(eCode, msg, cause);  
    }  
}  
}
```

And the following code fragment:

```
try {  
    String param1 = "Oracle";  
  
    if (param1.equalsIgnoreCase("oracle")) {  
  
        throw new ExSub(9001, "APPLICATION ERROR-9001", new  
FileNotFoundException("MyFile.txt"));  
    }  
  
    throw new ExSuper(9001, new FileNotFoundException("MyFile.txt"));  
} catch (ExSuper ex) {  
  
    System.out.println(ex.getMessage());  
}
```

What is the result?

```
9001: APPLICATION ERROR-9001-MyFile.txt  
Compilation fails at Line 1.  
9001: APPLICATION ERROR-9001-MyFile.txt  
9001: java.io.FileNotFoundException: MyFile.txt-MyFile.txt  
9001: java.io.FileNotFoundException: MyFile.txt-MyFile.txt
```

Question 122

Given, TripleThis. java:

```
import java.util.function.*;  
public class TripleThis {  
    public static void main (String[] args) {  
        Function tripler = x → ( return (Integer) x * 3; );  
        TripleThis. printValue (tripler, 4);  
        public static void printValue (Function f, l num) +
```

```
System.out.println(f.apply(num));  
}  
}
```

Compiling TripleThis.java gives this compiler warning:

Note: TripleThis.java uses unchecked or unsafe operations.

Which two replacements remove this compiler warning and prints 12?

- a) Replace line 9 with Function tripler = x > (return x - 3; 1)
- b) Replace line 12 with public static void printValue (Function f, Integer num)
- c) Replace line 9 with Function tripler = x > (return x - 3;)
- d) Replace line 9 with Function tripler = x - (return (Integer) x * 3;)
- e) Replace line 12 with public static void printValue (Function f, int, num)
- f) Replace line 12 with public static void printValue (Function f, T num)

Question 123

Given the code fragment:

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

What is the output?

The compilation fails.

A runtime exception is thrown.

- [-3, -2, -1]
- [-1, -2, -3]

Question 124

Which two statements are correct about modules in Java?

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

Question 125

Given the code fragment:

```
var i = 1:  
var result = IntStream.generate (() -> ( return i, ))  
.limit (100) .sum () :  
System.out.println (result);
```

Which statement prints the same value of result?

```
System.out.println (IntStream.rangeClosed (0, 100) .map (x -> 7) .count ()) ;  
System.out.println (IntStream.range (1, 100) .count () ;  
System.out.println (IntStream.rangeClosed (1, 100) .count ()) ;  
System.out.println (IntStream.range (0, 99) .count ()) ;
```

Question 126

```
public class Main {  
    public static void main (String[] args) {  
        Thread t1 = new Thread (new MyThread ()) ;  
        Thread t2 = new Thread (new MyThread ()) ;  
        Thread t3 = new Thread (new MyThread ()) ;  
        t1.start () ;  
        t2.run () ;  
        t3.start () ;  
        t1.start () ;  
    }  
    class MyThread implements Runnable {  
        public void run () {  
            System.out.println ("Running.");  
        }  
    }  
}
```

Which one is correct?

- A. An IllegalThreadStateException is thrown at run time.
- B. Three threads are created.
- C. The compilation fails.
- D. Four threads are created.

Question 127

Which two statements are true about running code on the class path and the module path?

- A modular JAR placed on the class path results in an automatic module.
- A non-modular JAR placed on the class path results in an unnamed module.
- A modular JAR placed on the module-path results in a named application module.
- A non-modular JAR placed on the module-path results in a named application module.
- A modular JAR placed on the class path results in a named application module.

Group 4 Dump 4

Question 1:

Given:

```
public class StrBldr {  
    static StringBuilder sb1 = new StringBuilder("yo ");  
    StringBuilder sb2 = new StringBuilder("hi ");  
    public static void main(String[] args) {  
        sb1 = sb1.append(new StrBldr().foo(new StringBuilder("hey")));  
        System.out.println(sb1);  
    }  
    StringBuilder foo(StringBuilder s) {  
        System.out.println(s + " oh " + sb2);  
        return new StringBuilder("ey");  
    }  
}
```

What is the result?

- A) oh hi hey
- B) Compile time error occurs.
- C) hey oh hi yo ey
- D) hey oh hi
- E) hey oh hi ey
- F) yo ey

Question 2:

Code:

```
import java.io.Serializable;  
  
public class MyPersistenceData implements Serializable{  
  
    String str;  
  
    private void methodA(){  
  
        System.out.println("methodA");  
  
    }  
  
}
```

You want to implement the java.io.Serializable interface to the MyPersistenceData class.

Which method should be overridden?

- A) The readExternal and writeExternal method.
- B) Nothing.**
- C) The writeExternal method.
- D) The readExternal method.

Question 3:

Code:

```
package q3;
```

```
import java.time.LocalDate;  
  
import java.util.ArrayList;  
  
import java.util.Comparator;  
  
import java.util.List;  
  
import java.util.Map;  
  
import java.util.Optional;  
  
import java.util.stream.Collectors;
```

```

public class Sc {

    public static void main(String[] args) {
        List<Employee> roster = new ArrayList<>();

        // Example employees
        roster.add(new Employee("John Doe", "Downtown", LocalDate.of(1990, 1, 15),
50000));
        roster.add(new Employee("Jane Smith", "Suburb", LocalDate.of(1985, 2, 20),
60000));
        roster.add(new Employee("Jim Brown", "Downtown", LocalDate.of(1992, 3, 25),
55000));
        roster.add(new Employee("Emily White", "Suburb", LocalDate.of(1980, 4, 30),
61000));

        Map<String, Optional<Employee>> highestPaidEmployeesByNeighborhood =
roster.stream()

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

        highestPaidEmployeesByNeighborhood.forEach((neighborhood, employee) -> {
            employee.ifPresent(e -> System.out.println(neighborhood + ":" + e.getName()
+ " with salary " + e.getSalary()));
        });
    }
}

Given:
public class Employee {

```

```

private String name;
private String neighborhood;
private LocalDate birthday;
private int salary;
}

and

List roster= new ArrayList<>();

Map m= roster.stream();

//line 1

```

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

- A) .collect(Collectors.groupingBy(Employee::getNeighborhood,
 Collectors.maxBy(Comparator.comparing(Employee::getSalary))
));
- B) .collect(Collectors.groupingBy(e->e.getNeighborhood(),
 Collectors.maxBy((x,y)->y.getSalary()-x.getSalary())
));
- C) .collect(Collectors.maxBy((x,y)->y.getSalary(x.getSalary()),
 Collectors.groupingBy(Employee::getNeighborhood)
));
- D) .collect(Collectors.maxBy(Employee::getSalary,
 Collectors.groupingBy(Comparator.comparing(e->e. getNeighborhood()))
));

Question 4:

Given:

```
public class StrBldr {
```

```

static StringBuilder sb1 = new StringBuilder("yo ");

StringBuilder sb2 = new StringBuilder("hi ");

public static void main(String[] args) {
    sb1.append(new StrBldr().foo(new StringBuilder("hey")));
    System.out.println(sb1);
}

StringBuilder foo(StringBuilder s) {
    System.out.println(s + " oh " + sb2);
    return new StringBuilder("ey");
}
}

```

What is the result?

- A) oh hi hey
- B) Compile time error occurs.
- C) hey oh hi yo ey**
- D) hey oh hi
- E) hey oh hi ey
- F) yo ey

Question 5:

Code:

```

public class Main {

    public static void main(String[] args) throws IOException{
        final Reader reader = new FileReader("File1.txt");
        try (reader) {
            reader.read(); // line 1
        } finally {
            reader.read(); // line 2
        }
    }
}

```

```
reader.read(); // line 3  
}  
}  
}
```

If File1.txt does exist, what is the result?

Options:

- A. A java.io.IOException is thrown on line 3.
- B. The program executes and prints nothing.
- C. The compilation fails.
- D. A java.io. IOException Lon is thrown on line 1.
- E. A java.io.IOException is thrown on line 2.

Question 6:

Given the code fragment:

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

You want to examine whether path is a directory.

Which code inserted on line 1 will accomplish this?

Options:

- A. BasicFileAttributes attributes=Files.readAttributes(path, "isDirectory");

- B. BasicFileAttributes attributes = Files.readAttributes(path,
BasicFileAttributes.class);**
- C. BasicFileAttributes attributes=Files.isDirectory(path);**
- D. BasicFileAttributes attributes = Files.readAttributes(path, FileAttributes.class);**

Question 7:

Code:

```
class.forName(JDBC_DRIVER_CLASS_NAME);
```

When is it necessary to execute this statement?

- A.It must be executed before each all to DriverManager to get a connection Using the named JDBC driver.**
- B.It must be executed once and only before the first call to DriverManager to get a Connection using the named JDBC driver.**
- C.It must be executed once and before accessing the named JDBC driver in any way.**
- D.It is no longer required to execute this method.**

Question 8:

Code:

```
StringBuilder txt1 = new StringBuilder("PPQRRRSTT");
```

```
int i=0;
```

a:

```
while(i<txt1.length()) {
```

```
char x = txt1.charAt(i);
```

```
int j=0;
```

```
i++;
```

b:

```
while(j<txt1.length()) {
```

```
char y=txt1.charAt(j);
```

```
if(i != j && y == x) {  
    txt1.deleteCharAt(j);  
    // line 1  
}  
j++;  
}  
}  
System.out.println(txt1);
```

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

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

Question 9:

Given the code fragment:

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

17. }

18. }

What is the result?

A. The compilation fails at line 16.

B. The compilation fails at line 9.

C. 10

D. The compilation fails at line 13.

E. 1

Question 10:

Given:

```
public class Main {  
    public static void main(String[] args) {  
        String source = "/u01/work/stage/message.txt";  
        String destination = "/u01/work/message.txt";  
        // line 1  
        }catch(IOException e){  
            e.printStackTrace();  
        }  
    }  
}
```

You want to move source.txt to the destination directory even if a file with the same name is in the destination directory.

Which code inserted on line 1 will accomplish this?

A. try{

 Files.move(Paths.get(source), Paths.get(destination));

B. try (FileChannel1 in = new

 FileInputStream(source).getChannel1());

```

FileChannel1 out = new
FileOutputStream(destination).getChannel(){}
in.transferTo(0, in.size(), out);
C. try {
    Files.move(Paths.get(source), Paths.get(destination),
    StandardCopyOption.REPLACE_EXISTING);
D. try {
    Files.copy(Paths.get(source), Paths.get(destination),
    StandardOperation.CREATE_NEW);
    Files.delete(Paths.get(source));

```

Question 11:

Which three initialization are correct?

- a) float x=1f;// since f is used for float it converts the value to float
- b) int [][] [] e= {{1,1,1},{2,2,2}};// for 3-D additional one more curly bracket required
- c) String contact#="(+2) (999) (232) ";//# cant be used for variable name
- d) short sh=(short)'A';//it type casts from character to short
- e)int x=12_34; //int can have _ in between numbers
- f) byte b=10;char c=b;// it has to type cast to character

Question 12:

Given the code fragment

Integer i=11;

- a)Double b=Double.valueOf(i);
- b) Double b=Double.parseDouble(i);// for parseDouble the argument can't be integer;
- c) Double a=i; // cant convert from Integer to Double
- d) Double c= (Double)i;// cannot cast from Integer to Double

e) double d=i;

Question 13:

Code:

```
public static void main(String args[]) {  
    String s = "10";  
    try {  
        int x = 0;  
        x = Integer.parseInt(s,2);  
        System.out.println("x is "+x);  
    }  
    catch(NumberFormatException e){  
        System.out.println("Error parsing value of "+x); //line 2  
    }  
}
```

What is the result?

A Error parsing value 0

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

C x is 2.

D x is 10.

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

Question 14:

Code:

```
public class Person {  
    private String name = "Green";  
    public void setName(String name) {  
        String title = "Mr. ";  
        this.name = title+name;
```

```
}

public String toString(){

return name;

}

}

and

public class Test{

public static void main(String[] args){

Person p = new Person();

p.setName("Blue");

System.out.println(p);

}

}
```

What is the result?

A An exception is thrown at runtime.

B Green

C Mr. Blue

D Mr. Green

Question 15:

Code:

```
public abstract class Automobile { // line 1

abstract void wheels();

}

public class Car extends Automobile { // line 2

void wheels(int i){ // line 3

System.out.print(4);

}
```

```
public static void main(String[] args) {  
    Automobile ob = new Car(); // line 4  
    ob.wheels();  
}  
}
```

What must you do so that the code prints 4?

- a) Replace the code in line 2 with Car ob = new Car();
- b) Remove the parameter from the wheels method in line 3.**
- c) Remove abstract keyword in line 1.
- d) Add @Override annotation at line 2.

Question 16:

Code:

```
public interface Worker {  
    public void doProcess();  
}  
  
public class HardWorker implements Worker {  
    public void doProcess() {  
        System.out.println("doing things");  
    }  
}  
  
public class Cheater implements Worker {  
    public void doProcess() {}  
}  
  
public class Main <T extends Worker> extends Thread { // Line 1  
    private List<T> processes = new ArrayList<>(); // Line 2  
    public void addProcess(HardWorker w) { // Line 3  
        processes.add(w);  
    }  
}
```

```

    }

    public void run() {
        processes.forEach((p) -> p.doProcess());
    }
}

```

What needs to change to make these classes compile and still handle all types of interface Worker?

Options:

- a) Replace Line 3 with **public void addProcess(Worker w){ }**
- b) Replace Line 3 with **public void addProcess(T w){ }****
- c) Replace Line 1 with **public class Main extends Thread { }**
- d) Replace Line 2 with **private List processes = new ArrayList<>();**

Question 17:

Code:

```

Locale l = new Locale("en", "US");

LocalDate today = LocalDate.of(2018, 12, 17);

String mToday =
today.format(DateTimeFormatter.ofLocalizedDate(FormatStyle.MEDIUM));

String sToday =
today.format(DateTimeFormatter.ofLocalizedDate(FormatStyle.SHORT));

System.out.println(mToday);

System.out.println(sToday);

```

What is the result?

O December 17, 2018

12/17/18

O Friday, December 17, 2018

December 17, 2018

O 12/17/18

Dec 17, 2018

● Dec 17, 2018

12/17/18

Question 18:

Code:

```
public class ResourceTest {  
    public static void main(String[] args){  
        final MyResource res1 = new MyResource();  
        MyResource res2 = new MyResource();  
        try(res1 ; res2){  
            // do something  
        } catch(Exception e) {}  
    }  
    static class MyResource implements AutoCloseable {  
        public void close() throws Exception {}  
    }  
}
```

Which statement is true?

- O The code fails to compile as MyResource must implement Closeable.
- O The code compiles successfully.
- O The code fails to compile as res2 should be declared as final.
- O The code fails to compile as try-with-resource needs a variable declaration such as MyResource r1 = res1; MyResource r2 = res2;**

Question 19:

Code:

Which two statements are true about a class that is marked @deprecated?

- A. There is always another class that can be used instead of the deprecated class.
- B. The author of the class wants to discourage people from using the class in any way.**
- C. Using the class is guaranteed to cause errors at runtime.
- D. Using the class will cause the Java compiler to give a warning.**
- E. The class cannot be extended.

Question 20:

Code:

```
interface MyInterface1 {  
    public int method() throws Exception;  
    private void pMethod() { /* an implementation of pMethod */ }  
}  
  
interface MyInterface2 {  
    public static void sMethod() { /* an implementation of sMethod */ }  
    public boolean equals();  
}  
  
interface MyInterface3 {  
    public void method();  
    public void method(String str);  
}  
  
interface MyInterface4 {  
    public void dMethod() { /* an implementation of dMethod */ }  
    public void method();  
}  
  
interface MyInterface5 {
```

```
public static void sMethod();  
public void method(String str);  
}
```

Which two interfaces can be used in lambda expressions ?

- A. MyInterface3
- B. MyInterface2**
- C. MyInterface5
- D. MyInterface4
- E. MyInterface1**

Question 21:

Code:

```
public class Person {  
    private String name;  
    private int age;  
    public Person(String name, int age){  
        this.name = name;  
        this.age = age;  
    }  
    public int getAge(){  
        return age;  
    }  
}  
  
var persons= Arrays.asList(new Person("Haz", 18).  
    new Person("Peter", 23).  
    new Person("Famela", 23).  
    new Person("David", 12));  
  
int num =persons.stream().
```

```
mapToInt(Person::getAge).
```

```
filter (p->p<20).
```

```
reduce(0, (a,b)->a+b);
```

```
System.out.println(num);
```

a) 30

b) 41

c) 46

d) 35

Question 22:

22. Which code fragment does a service use to load the service provider with a Print interface?

A. private Print print = com.service.Provider.getInstance();

B. private java.util.ServiceLoader loader = ServiceLoader.load (Print.class);

C. private java.util.ServiceLoader loader = new java.util.ServiceLoader<> ();

D. private Print print = new com.service.Provider.PrintImpl();

Question 23:

23. Which two statements are true about running code on the class path and the module path?

A. A non-modular JAR placed on the --classpath results in an unnamed module.

B. A modular JAR placed on the --module-path results in a named application module.

C. A non-modular JAR placed on the --module-path results in a named application module.

D. A modular JAR placed on the --classpath results in a named application module.

E. A modular JAR placed on the --classpath results in an automatic module..

Question 24:

Code:

```
. A() /* line n1 */ {  
    super("The Mandatory Criteria Yet to Meet");  
}  
20. public class TestCE {  
21.     public static void main(String[] args) throws A {  
22.         int a = 10, b = 13;  
23.         try {  
24.             if (a < b) {  
25.                 throw new A();  
26.             }  
27.         } catch (Exception e) {  
28.             System.out.println(e);  
29.         }  
30.         System.out.println("Continue...");  
31.     }  
32. }
```

"You must define the A exception class. The program execution must be terminated if the condition at line 24 fails and an exception is thrown at line 25.

Which statement at line n1 defines A as per the requirement?

- a) class A extends Exception {
- b) class A extends RuntimeException {
- c) class A extends Throwable {
- d) class A extends ArithmeticException {"

Question 25:

Code:

```
Supplier supplier = () -> "Hello World";
```

```
// line 1
```

Which statement on line 1 is calling the method of the supplier object correctly?

- a) System.out.println(supplier.accept());
- b) System.out.println(supplier.test());
- c) System.out.println(supplier.get());
- d) System.out.println(supplier.apply());

Question 26:

Code:

```
String s = "Oracle";  
Runnable r = () -> {  
    System.out.println(s);  
};  
s = "Java";  
Thread t = new Thread(r);  
t.start();
```

The multiple-choice question asks: "What is the result?" with the following options:

- a) Compilation error // Local variable s defined in an enclosing scope must be final or effectively final
- b) Oracle
- c) An exception is thrown at run time.
- d) Java

Question 27:

Code:

Given:

```
int i=0;  
do {
```

```
for(int j = i/2; j>0; j--) {  
    System.out.println(j + " ");  
}  
  
i=2;  
} while(i>0);
```

What is the result?

- A. 5
- B. Nothing**
- C. 5 4 3 2 1
- D. 5 4 3 2 1 4 3 2 1 3 2 1 2 1 1

Question 28:

Code:

Given the code fragment:

```
import java.io.File;  
  
import java.io.FileInputStream;  
  
import java.io.IOException;  
  
public class Hema {  
  
    public static void main(String[] args) throws IOException {  
  
        String filename = "/u01/work" + args[0];  
  
        // line 1  
  
        File file = new File(filename);  
  
        String canonicalPath = file.getCanonicalPath();  
  
        FileInputStream fis = new FileInputStream(file);  
  
        // ...  
  
    }  
}
```

You want to validate a path name before the read file. Before validation, all path names should be canonicalized.

Which code inserted on the line n1 will accomplish this?

A) `File file = new File(filename);`
`String canonicalPath = file.getCanonicalPath();`
`FileInputStream fis = new FileInputStream(file);`

B) `File file = new File(filename).getAbsoluteFile();`
`FileInputStream fis = new FileInputStream(file);`

C) `Path file = paths.get(filename);`
`String canonicalPath = file.normalize().substring();`
`FileInputStream fis = new FileInputStream(canonicalPath);`

D) `Path file = paths.get(filename);`
`Path canonicalPath = file.toAbsolutePath().toString();`
`FileInputStream fis = new FileInputStream(canonicalPath);`

Question 29:

Code:

```
import java.util.concurrent.ExecutionException;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.Future;
class Test {
    public static void main(String[] args) throws InterruptedException,
    ExecutionException {
        ExecutorService es = Executors.newCachedThreadPool();
        es.execute(() -> System.out.println("Ping "));
        Future<String> future = es.submit(() -> "Pong"); //added line
        System.out.println(future.get()); // line 2
        es.shutdown();
    }
}
```

```
}
```

```
}
```

Which statement at line 1 will print Ping Pong?

A) Future<String> future = es.submit(() -> "Pong");

B) Future<String> future = new Callable(){
public String call() throws Exception {
return "Pong";
}
.call

C) Future<String> future = es.invokeAny(new Callable<String> () {
public String call () throws Exception {
Return "Pong";
}
});

D) Future <String> future = es.execute (() -> "Pong");

Question 30:

Code:

```
public class A {  
    int a = 0;  
    int b = 0;  
    int c = 0;  
  
    public synchronized void foo(int i) {  
        a += b * 1;  
        c -= b * 1;  
        System.out.println("a: " + a + ", c: " + c);  
    }  
  
    public synchronized void setB(int i) {
```

```
b = i;  
}  
  
public static void main(String[] args) {  
    A instance = new A();  
    instance.setB(5);  
    instance.foo(0);  
}  
}
```

Which makes class A thread safe?

- A. Make sets synchronized.
- B. Make foo synchronized.
- C. Class A is thread safe.
- D. Make foo and sets synchronized.**
- E. Make A synchronized.

Question 31:

```
public interface APIInterface {  
    public default void process() {  
        System.out.println("Process() called 1.");  
    }  
}  
  
public abstract class AbstractAPI {  
    public abstract void process();  
}  
  
public class Apimpl extends AbstractAPI implements APIInterface {  
    @Override  
    public void process() {  
        System.out.println("Process() called: 2.");  
    }  
}
```

```
public static void main(String[] args) {  
    var impl = new Apimpl();  
    impl.process();  
}  
}
```

What is the result?

Options:

- A. The compilation fails.
- B. A java.lang.NoSuchMethodException is thrown.
- C. The program prints Process() called: 1.
- D. The program prints Process() called 2.**
- E. A java.lang.IllegalAccessException is thrown.

Question 32:

Code:

```
package pac;  
  
public class Hello {  
    public static void main(String[] args) {  
        Module module = Hello.class.getModule();  
        System.out.println("Module: "+module);  
        System.out.println("Name: "+module.getName());  
        System.out.println("Descriptor: "+module.getDescriptor());  
    }  
}
```

Given the directory structure:

\Test

 Hello.java

Given the commands to execute at the Test directory prompt:

Test>java -cp pac pac.Hello

Which statement is true?

A.Execute java --module-path pac pac.Hello instead of java -cp pac pac.Hello and on execution the program prints:

Module: pac @<<hash code>>

Name: pac.Test

Descriptor: null

B.On execution of the given commands,the program prints:

Module: unnamed module @<<hash code>>

Name: null

Descriptor: null

C.Create an empty module-info.java file in the Text directory and on execution of the given commands, the program prints:

Module: unnamed module @<<hash code>>

Name: null

Descriptor:module-info

D.On execution of the given commands the program prints:

Module: pac.Hello @<<hash code>>

Name: unnamed

Descriptor: null

Question 33:

Code:

Given:

```
class Employee{
```

```
    String office;
```

```
}
```

and the code fragment:

```
5. public class HRApp{
```

```
6.     var employee = new ArrayList();
```

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

Which two lines cause compilation errors?

- a) line 6
- b) line 12
- c) line 8
- d) line 7

Question 34:

Code:

```
Given: public class Sports {  
public double getRatings()  
{  
...  
}  
}  
public class Football extends Sports {  
public double getRatings()  
{  
...  
}
```

}

Which is the correct implementation of the getRatings method in the Football subclass?

- a)The subclass getRatings method implementation directly accesses the fields in the Sports superclass.
- b)The subclass getRatings method uses public getRatings() to call the base class method but uses its own named fields in the implementation
- c)The subclass getRatings method uses new.getRatings() to call the base class method but uses its own named fields in the implementation.
- d)The subclass getRatings() method uses super.getRatings() to call the base class method but uses its own named fields in the implementation.

Question 35:

Code:

```
public class Person {  
    private String name = "Joe Bloggs";  
    public Person(String name) {  
        this.name = name;  
    }  
    public String toString() {  
        return name;  
    }  
}  
and  
public class Tester{  
    public static void main(String[] args) {  
        Person p1 = new Person(); // line 1  
        System.out.println(p1);  
    }  
}
```

}

What is the result?

A p1

B null

C The Compilation fails due to an error in line 1.

D Joe Bloggs

Question 36:

Code:

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

and

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

What is the result?

a) p1

- b) null
- c) The compilation fails due to an error in line 1.
- d) Joe Bloggs

Question 37:

Code:

Which two can be considered good practices for serializing Java objects?

- O Implement secure serialization by generating secure object hash or using encryption.
- O Implement serialization for long-term data storage.
- O Assign null value by default while serializing and deserializing a transient variable.
- O Always override the readObject/writeObject methods from the java.io.Serializable interface.
- O Ensure that the class definition used is the same as the class definition used by Java runtime at the time when the object was serialized.

Question 38:

Code:

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

and

```
import java.util.List;

public class Test{

public static void main(String[] args){

var items = List.of(new Item("A", -2), new Item("B", 2),
new Item("C", 12), new Item("D", 5), new Item("E", 6));

if (items.stream().anyMatch(i -> i.count < 0)) {

System.out.println("There is an item for which the variable count is below zero.");

}

}

}
```

You want to examine the items list it contains an item for which the variable count is below zero. Which code fragment at line 1 accomplish this?

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

Question 39:

Code:

Why would you choose to use a peek operation instead of a forEach operation on a Stream?

- a) To remove an item from the end of the stream.
- b) To remove an item from the beginning of the stream.**
- c) To process the current item and return a stream.
- d) To process the current item and return void.

Question 40:

Code:

```
char[] characters = new char[100];  
FileReader reader = new FileReader("file_to_path");  
try {  
    //line 1//  
    System.out.println(String.valueOf(characters));  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

You want to read data through the reader object. Which statement inserted on line 1 will accomplish this?

- a) characters.read();
- b) reader.readLine();
- c) characters = reader.read();
- d) reader.read(characters);

Question 41:

Code:

```
import java.util.ArrayList;  
import java.util.Arrays;  
public class NewMain {  
    public static void main(String[] args) {  
        String[] catNames = { "abyssinian", "oxicat", "korat", "laperm", "bengal", "sphynx" };  
        var cats = new ArrayList<>(Arrays.asList(catNames));  
        cats.sort((var a, var b) -> a.compareTo(b));  
        cats.forEach(System.out::println);  
    }  
}
```

Answer:

A. abyssinian

oxicat

korat

laperm

bengal

sphynx

B.sphynx

oxicat

laperm

korat

bengal

abyssinian

C.abyssinian

bengal

korat

laperm

oxicat

sphynx

D. nothing

Question 42:

Code:

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

```
return name;  
}  
}  
  
public class Tester {  
    static Person p = null;  
    public static void main(String[] args) {  
        p = checkPerson(p);  
        System.out.println(p);  
        Person p1 = new Person("Joe");  
        p1 = checkPerson(p);  
        System.out.println(p1);  
    }  
    public static Person checkPerson(Person p) {  
        if(p==null) {  
            p = new Person("Mary");  
        }  
        return p;  
    }  
}
```

What is the result?

- A) Mary Mary
- B) Mary Joe
- C) null null
- D) Joe Joe

Question 43:

Code:

Given:

```
static void add (List l){  
    l.add(4);  
    l.add(3.14f);  
}  
  
public static void main(String[] args){  
    var x = new ArrayList();  
    x.add(3);  
    add(x);  
  
    for (Integer i: x) {  
        System.out.print(i+" ");  
    }  
}
```

What is the result?

- A) The program prints 3 4 and throws a classCastException.
- B) 3 4 3.14
- C) 3 4 3
- D) 3 4

Question 44:

Code:

Given this enum declaration:

- 1. enum Alphabet {
- 2. A, B, C;
- 3.
- 4.}

Examine this code:

```
System.out.println(Alphabet.getFirstLetter());
```

What code should be written at line 3 to make this code print A?

- A) static String getFirstLetter() { return Alphabet.values()[1].toString(); }
- B) final String getFirstletter() { return A.toString(); }
- C) static String getFirstletter() { return A.toString(); }
- D) String getFirstLetter() { return A.toString(); }

Question 45:

Code:

```
public class Over {

    public void analyze(Object[] o) {
        System.out.println("I am an object array");
    }

    public void analyze(long[] l) {
        System.out.println("I am an array");
    }

    public void analyze(Object o) {
        System.out.println("I am an object");
    }

    public static void main(String[] args) {
        int[] nums = new int[10];
        new Over().analyze(nums); // This will invoke the analyze(Object o) method
    }
}
```

what is the output?

a) The compilation fails due to an error in line 1.

b) I am an object

c) I am an array

d) I am an object array

Question 46:

Code:

```
package dumps;  
  
public class DoClass {  
  
    static String s;  
  
    public static void main(String[] args){  
  
        switch(s){  
  
            case "41": s += "41";  
  
            default: s+= "def ";  
  
            case "42": s += "42";  
  
        }  
  
        System.out.println(s);  
  
    }  
  
}
```

What is the output?

A)41 def 42

B)def 42

C)An exception is thrown at runtime.

D)null

Question 47:

Code:

```
import java.util.*;  
  
public class Foo {  
    public void foo(Collection ag) {  
        System.out.println("Bonjour le monde!");  
    }  
}  
  
class Bar extends Foo {  
    public void foo(Collection arg) {  
        System.out.println("Hello world!");  
    }  
  
    public void too(List arg) {  
        System.out.println("Hola Mundo!");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        ArrayList li = new ArrayList(); // Creating an ArrayList  
        Bar b = new Bar();           // Creating Bar instance  
        Foo f = b;                 // Upcasting Bar to Foo  
  
        b.foo(li);  
        f.foo(li);  
    }  
}
```

What is the output?

A) Bonjor le monde!

Hello World

B) Hello World!

Hello World!

C) Hello World!

Bonjor le monde!

D) Bonjour le monde!

Bonjour le monde!

Question 48:

Code:

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

What is the result?

- A) -1
- B) -3
- C) 2
- D) 1**

Question 49:

Code:

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

}

and

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

and

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

Which three are correct? (Choose three.)

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

Question 50:

Code:

Given:

```
public interface InterfaceOne {  
    public void methodA();  
    public void methodB();  
}
```

and

```
public interface InterfaceTwo extends AbstractClass {}
```

and

```
public abstract class AbstractClass implements InterfaceOne {  
    public String origin "Abstract Class";  
    public void methodA(){  
        System.out.println("A");  
    }  
    public abstract void methodC();  
}
```

and

```
public class ConcreteClass extends AbstractClass {  
    public void methodC(String c){  
        System.out.println(c);  
    }  
}
```

Which three changes make this code compile?

- a) Remove methodA() from AbstractClass
- b) Implement methodC() in ConcreteClass
- c) Implement methodB() in ConcreteClass
- d) Add the keyword abstract to the methodA() and methodB() declarations in InterfaceOne
- e) Remove methodA() from InterfaceOne
- f) InterfaceTwo should no longer extend AbstractClass
- g) Implement methodA() in ConcreteClass

```
public class Employee {  
    private String name;  
    private String neighborhood;  
    // the constructors, setters, and getter methods go here  
}
```

and

```
List<Employee> roster = List.of(new Employee("John", "West town"),  
    new Employee("Ray", "South town"),  
    new Employee("Tom"),  
    new Employee("Kenny", "West town"));
```

A) Map<String, List<Employee>> e3 =

```
roster.stream()  
.collect(Collectors.groupingBy( e -> Optional.ofNullable (e.getNeighborhood())  
.get())
```

```
));

B) Map<String, List<Employee>> e3 =  
    roster.stream()  
        .collect(Collectors.groupingBy(  
            e->Optional.ofNullable(e.getNeighborhood())  
        .get()  
    ));  
  
C) Map<String, List<Employee>> e1 =  
    roster.stream()  
        .collect(Collectors.groupingBy(  
            (e -> Optional.ofNullable(e.getNeighborhood()))  
        ));  
  
D) Map<Object, List<Employee>> e2 =  
    roster.stream()  
        .collect(Collectors.groupingBy(  
            e -> Optional.ofNullable (e.getNeighborhood())  
        ));
```

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

