Oracle Java SE 11

Developer

EXAM : 1Z0-819

DUMPS

1) public class Tester {

public static void main(String[ ] args) {

int sum =0,x=0;

for (; x<3 ;sum+= ++x);//line1

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

}

}

What is the result?

a) -0-1-3-6

b) -6

c) The compilation fails due to an error in line 1

d) -0-1-3

2) Public class Test {

Public void process(byte v) {

System.out.println(“Byte value” +v);

}

Public void process(short v) {

System.out.ptintln(“Short value” +v);

}

Public void process(Object v) {

System.out.println(“Object value”+v);

}

Public static void main (String [ ] args) {

byte x = 12;

short y = 13;

new Test ( ).process(x+y);//line1

}

}

What is the output?

a) Short value 25

b) Object value 25

c) Byte value 25

d) The compilation fails due to error in line 1

3) Given-.

public interface Builder {

public A build(String str);

```
}
```

and

```
public class BuilderImpl  implements Builder {
@Override
public B build (String  str) {
return new B(str);
}
}
```

Assume that this code compiles correctly , which three statements are true ?

a) B is a subtype of A.

b) B cannot be abstract B

c) B cannot be final

d) A cannot be abstract

e) A is a subtype of B

f) A cannot be final

4) Given the code fragment :

```
String name = " ";
If(/* insert code here */ ) { // line1
System.out.println("Name is required");
}
```

What should be inserted in line 1 so that the code fragment prints Name is required ?

a) name.isBlank();

b) name.trim( ) == " "

c) name.isEmpty( )

d) name.compareTo(" ") == 0

5) Given :

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

and

```
pubic abstract class B /*Position 1*/ {
/*position 2 */
public abstract void Z( );
}
```

and

```
public class C extends B implements A{
/*position 3*/
}
```

Which code , when inserted at one or more marked position , would allow classes B and C to compile ?

a) implemets A //position 1
@Override //position 2
b) public void z( ) { } //position 3

c) @Override //position3
void x ( ) { } //position 3
@Override //position 3
public void z() { } //potion 3

d) @Override //position 2
public void z () { } //position 3

6) Which module defines the foundational API's of the Java SE platform ?

a) java.object

b) java.base

c) java.se

d) java.lang

7) Given the declaration :

@Target{{TYPE, METHOD}}

@interface Resource { }

/*Loc1*/ class Manager extends /*Loc2*/ Person {

/*Loc3 */ Manager( ) {.....}

/*Loc4*/ String getRepartmentName( ) {.....}

/*Loc5* String departmentName;

}

In which two locations is it legal to apply the @Resource annotation ?

a) Loc1

b) Loc3

c) Loc4

d) Loc2

e) Loc5

8) 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("courses.txt");

Try {

/*line 1*/

} catch (IOException ex) {

System.out.println("File IO Exception is thrown ", ex);

}

Which code fragment at line 1 prints the line that contains Java from the

courses.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).cantains("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"));
```

9) 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) Implement serialization for long-term data storage

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

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

e) Implement secure serialization by generating secure object hash or using encryption

10) 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 already exists in the destination directory

Which code inserted on line 1 will accomplish this ?

a) try (FileChannel in = new

FileInputStream(source).getChannel( );

FileChannel out = new FileOutputStream(destination).getChannel( ) ;

{

In.transferTo(0, in.size( ), out);

b) try {

Files.copy(Paths.get(source), Paths.get(destination),

StandardOpenOption.CREATE_NEW);

Files.delete(Paths.get(source));

c) try {

Files.move(Paths.get(source),Paths.get(destination),

StandardCopyOption.REPLACE_EXISTING);

d) try {

Files.move(Paths.get(source),Paths.get(detination));

}

11) Which set of commands is necessary to create and run a custom image from Java source file ?

a) Jar , jlink

b) Javac, jlink

c) Java , jdeps

d) Javac, jar

12) 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);
for( int c =0 ; c < lst.size(); c++) {
display(lst.get (c)));
}
}
public static void display (var c) {
System.out.print(“>” + c);
}
```

What is the result ?

a) > 1 > 2.0 >4.0

>1 > 2.0 > 3 > 4.0

b) A compile time error occurs at line n2.

c) > 1 >2.0 > 4.0

>1 > 2.0 > 4.0

d) An exception is thrown at line n1.

13) Which two commands are used to identify class and module dependencies ?

a) java Hello.java

b) jar –show-module-resolution

c) jdeps -—list-deps

d) java -—show-module-resolution

e) jmod describe

14) Which two expressions create a valid Java Path instance?

a) Paths.get(URI.create("file:///domains/oracle/test.txt"))

b) Paths.get("foo")

c) Paths.getPath("too") new Path("foo")

d) Path.get(new URI ("file:///domains/oracle/test.txt"))

15) Given the code fragment:

```
Locale locale = new Locale("en", "US");
LocalDate today = LocalDate.of(2018, 12, 17);

String mToday=
today.format(DateTimeFormatter.ofLocalizedDate(FormalStyle.MEDIU
 M). withLocale(locale));

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

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

System.out.println(sToday);

What is the result?

Dec 17, 2018
12/17/18

16) 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) 42

b) def 42

c) 41 def 42

d) Compilation fails

17) 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 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<>();
itemsl.add(new Item (1, "Pen"));
itemsl.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(bl); out.flush(); out.close();
 }
catch (Exception e) {
System.out.println("Unable to Serialize");
}
 }
```

unable to serialize
if item implements serializable it will be serializable.

18) Given:

```
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 line 1 to use ParallelStream. Which one is correct?

a) The code may produce a different result.

b) The code will produce the same result.

c) The compilation fails.

d) A NoSuchElementException is thrown at runtime.

19) Which code fragment represents a valid Comparator implementation?

a) public class comps implements Comparator {
public int compare (String str1, String str2) {
return str1.length() str2.length();
}
}

b) public class Comps implements Comparator {
public boolean compare (Object obj1, Object obj2) {
return obj1.equals(obj2);
}
}

c) new Comparator () {
public int compareTo (String str1, String str2) {
return str1.compareTo(str2);
}
}

d) new Comparator () {

```
public int compare (String str1, String str2) {
return str1.compareTo(str2);
}
}
```

20)
```
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("X is "+x);
 }
catch (NumberFormatException e) {
System.out.println("Error parsing value of "+x); // line 2
}
}
}
```

What is the result?

a) X is 10.

b) The compilation fails due to an error in line 1.

c) X is 2.

d) Error parsing value 0

e) The compilation fails due to an error in line 2.

21) 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) Hello World!
Hello World!

d) Bonjour le monde!
Bonjour le monde!

22) 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!

b) Hello World!

Bonjour le monde

c) Hello World!

Hello World!

d) Bonjour le monde!

Bonjour le monde!

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 = s1.peak(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?

banana orange apple lemon

apple banana lemon orange

banana orange apple lemon apple banana lemon orange

banana orange apple lemon
apple banana lemon orange ...

banana orange apple lemon apple banana lemon orange

banana orange apple lemon apple banana lemon orange

24) Given:

```
package a;

abstract class A {

void print() {

System.out.print("Base class");

}
}
```

and

```
package ar;

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?

a) Derived class

b) The compilation fails.

c) An exception is thrown at runtime.

d) Base class

25) 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 fails.

d) An Illegal ThreadStateException is thrown at runtime.

26) Public class DNASynth (

int account;

int tcount;

int ccount;

int gCount;

DNASynth (int a, int tCount, int c, int g) {

// line 1

}

int setCCount(int c) {

```
return c;
}

void setGCount (int gCount) {

this.gCount = gCount;

}
}
```

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

a) aCount = a;

b) tCount=tCount;

c) cCount = setCCount(c);

d) setCCount(c) = CCount;

e) setGCount(g);

27) Given:

```
public class Tester {

public static void main(String[] args) {

Strings "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) C at once 1

c) at once 0

d) An IndexOutOfBoundsException is thrown at runtime.

e) hat at store 1

28) Which two statements are true about a class that is marked @Deprecated?

a) Using the class is guaranteed to cause errors at runtime.

b) The class cannot be extended.

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

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

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

29) Given :

```
public Interface Converter [

public static final double POUNDS_PER_KILOGRAM = 2.264627;

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

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) It compiles without errors.

30) Given :

```
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.test(number)) { predicate

b) if(p.accept(number)) {

c) if(p.get(number)) (

d) if(p.apply(number)) (

31) Given :

```
public class Employee {

private String name;

private LocalDate birthday;

private int salary;

/* the constructors, getters, and setters methods go here */
```

and

```
List roster = new ArrayList<>();

Predicate p = e -> e.getSalary() > 25;

LocalDate d = IsoChronology.INSTANCE. date (1989, 1, 1);

long youngAndRich = roster.stream()

// Line 1
```

Which code fragment, when inserted on line 1, gives the number of employees who were born after January 1, 1989 and have a salary greater than 25?

```
.filter(p && e.getBirthday().isAfter(d))
.count();

collect (Collectors.partitioningBy(p))
.get(true)
.stream()

.collect (Collectors.partitioningBy(e-> e.getBirthday().isAfter(d)))
.get(true)
.count();
.filter(p)


.filter(e->e.getBirthday().isAfter(d))
.count();
```

```
.filter (p)

.collect (Collectors.partitioningBy(e-> e.getBirthday().isAfter(d)))
.get(true)
.count();
```

32) Public class Main{

```
public static void main(String[] args) {
IntStream.range(1, 4 )
 .peek(System.out::print)
 .peek(i -> {  if (i == 3) throw new RuntimeException("Exception
thrown")
 .forEach(i -> { } );
}
};
```

What is the result ?

a) The program prints : 123 and the RuntimeException is thrown

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

c) The program prints nothing

d) The program prints: 12 and the RuntimeException is thrown

33) Which statement is true?

a) PrintWriter outputs characters and automatically flushes the stream.

b) Console.readPassword () method encrypts the text entered.

c) PrintStream outputs only bytes.

d) System.exit() invokes the close() method for the

InputStream/OutputStres resources.

34) Which module-info.java file would work for the new library version

clients-10.3jar7

a) module.com.company.clients{

requires com.company.clients;

}

b) module com.company.clients {

uses com.company.clients;

}

c) module com.company.clients {

exports com.company.clients.Client;

}

d) module com.company.clients {

exports com.company.clients;

}

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

a) Always override the readObject/writeObject methods from the

Java.io.Serializable interface.

b) Assign null value by default while serializing and deserializing a

transient variable.

c) 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.

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

e) Implement serialization for long-term data storage.

36) Given:

```
int i =31;
int j= 25;
System.out.println(i > 2 ? 1 > 10 ? 1 * (+10) : 1*1 + 5 : 1);
```

What is the result? 6

a) 385

b) 80

c) The compilation fails.

d) 25

a) 3

37) Given the code fragment :

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

public class FileHandler {
try (FileInputStream in = new FileInputStream (“foo.txt” )) { }
catch (FileNotFoundException e) { }
 }
 }
```

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

a) Replacing the catch block with:

catch (Exception | IOException e) { }

b) Replacing the catch block with:

catch (FileNotFoundException | Exception e) (1

finally { in.close(); }

c) Inserting:

finally { in.close( ); }

Adding throws IOException declaration at the main () method

d) Replacing the catch block with:

catch (Exception e) { }

Adding throws FileNotFoundException declaration at the main()

method

38) 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 I?

a) states.forEach((String s) -> {return System.out.println(s);});

b) states.forEach((var s) -> System.out.println(s));

c) states.forEach (var s -> (System.out.println(s)});

d) states.forEach((s) -> System.out.println(s););

39) Given :

public class Foo {

public static String ALPHA = "alpha";

protected String beta = "beta";

private final String delta;

public Foo (String d) { delta ALPHA + d; }

public String foo() { return beta delta; }

Which change would make Foo more secure?

a) private String delta;

b) public String beta = "beta";

c) public static final String ALPHA = "alpha";

d) protected final String beta = "beta";

40) Given :

```
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?

a) Nothing

b) Compilation fails

c) Changed to Java

d) Changed to Test