1Z0-804 Q&As

Java SE 7 Programmer II

Pass Oracle 1Z0-804 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass4itsure.com/1Z0-804.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Oracle Official Exam Center

QUESTION 1

Given the code fragment:

```
try {
conn.setAutoCommit(false);
stmt.executeUpdate("insert into employees values(1,\\'Sam\\')");
Savepoint save1 = conn.setSavepoint("point1");
stmt.executeUpdate("insert into employees values(2,\\'Jane\\')");
conn.rollback();
stmt.executeUpdate("insert into employees values(3,\\'John\\')");
conn.setAutoCommit(true);
stmt.executeUpdate("insert into employees values(4,\\'Jack\\')");
ResultSet rs = stmt.executeQuery("select * from employees");
while (rs.next()) {
System.out.println(rs.getString(1) + " " + rs.getString(2));
}
} catch(Exception e) {
System.out.print(e.getMessage());
}
```

What is the result of the employees table has no records before the code executed?

A. 1 Sam

B. 4 Jack

C. 3 John 4 Jack

D. 1 Sam 3 John 4 Jack

Correct Answer: C

Autocommit is set to false. The two following statements will be within the same transaction.

```
stmt.executeUpdate("insert into employees values(1,\\'Sam\\')");
stmt.executeUpdate("insert into employees values(2,\\'Jane\\')");
```

These two statements are rolled back through (the savepoint is ignored -the savepoint must be specified in the rollback

if you want to rollback to the savepoint):

conn.rollback();

The next two insert statements are executed fine. Their result will be in the output.

QUESTION 2

Given the code fragment:

try {

String query = "SELECT * FROM Employee WHERE ID=110";

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(query);

System.out.println("Employee ID: " + rs.getInt("ID"));

} catch (Exception se) {

System.out.println("Error");

}

Assume that the SQL query matches one record. What is the result of compiling and executing this code?

A. The code prints Error.

B. The code prints the employee ID.

C. Compilation fails due to an error at line 13.

D. Compilation fails due to an error at line 14.

Correct Answer: B

The code compiles fine. The code will rune fine.

public int getInt(String columnName) throws SQLException Retrieves the value of the designated column in the current row of this ResultSet object as an int in the Java programming language

QUESTION 3

You have been asked to create a ResourceBundle file to localize an application.

Which code example specifies valid keys menu1 and manu2 with values of File Menu and View Menu?

A. File Menu View Menu

B. menu1File Menu menu1View Menu

C. menu1m File menu, menu2, view menu

D. menu1 = File Menu menu2 = View Menu

Correct Answer: D

A properties file is a simple text file. You can create and maintain a properties file with just about any text editor.

You should always create a default properties file. The name of this file begins with the base name of your ResourceBundle and ends with the .properties suffix. In the PropertiesDemo program the base name is LabelsBundle. Therefore the default properties file is called LabelsBundle.properties. The following example file contains the following lines:

This is the default LabelsBundle.properties file s1 = computer s2 = disk s3 = monitor s4 = keyboard

Note that in the preceding file the comment lines begin with a pound sign (#). The other lines contain key-value pairs. The key is on the left side of the equal sign and the value is on the right. For instance, s2 is the key that corresponds to the value disk. The key is arbitrary. We could have called s2 something else, like msg5 or diskID. Once defined, however, the key should not change because it is referenced in the source code. The values may be changed. In fact, when your localizers create new properties files to accommodate additional languages, they will translate the values into various languages.

QUESTION 4

Given:

```
import java.util.*;

public class SearchText {

public static void main(String[] args) {

Object[] array1 = new Object[3];

array1[0] = "foo";

array1[0] = 1;

array1[0] = \\'a\\';

int index = Arrays.binarySearch(array1, "bar");

System.out.println(index);

}

}
```

What is the result?

A. ?1

B. 0

C. 2

D. Compilation fails

E. An exception is thrown at runtime

Correct Answer: E

The code compiles fine.

An exception is thrown at runtime due to data type comparison mismatch:

Exception in thread "main" java.lang.ClassCastException: java.lang.String cannot be cast to java.lang.Integer

at java.lang.Integer.compareTo(Integer.java:52)

at java.util.Arrays.binarySearch0(Arrays.java:1481)

at java.util.Arrays.binarySearch(Arrays.java:1423)

at searchtext.SearchText.main(SearchText.java:22)

Note: binarySearch

public static int binarySearch(char[] a, char key)

Searches the specified array of chars for the specified value using the binary search algorithm. The array must be sorted (as by the sort method, above) prior to making this call. If it is not sorted, the results are undefined. If the array contains

multiple elements with the specified value, there is no guarantee which one will be found.

Parameters:

a - the array to be searched.

key - the value to be searched for.

Returns:

index of the search key, if it is contained in the list; otherwise, (-(insertion point) - 1). The insertion point is defined as the point at which the key would be inserted into the list: the index of the first element greater than the key, or list.size(), if all

elements in the list are less than the specified key. Note that this guarantees that the return value will be >= 0 if and only if the key is found.

QUESTION 5

Given the integer implements comparable:

import java.util.*;

public class SortAndSearch2 {

static final Comparator IntegerComparator =

```
new Comparator() {
public int compare (Integer n1, Integer n2) {
return n2.compareTo(n1);
}
};
public static void main(String args[]) {
ArrayList list = new ArrayList();
list.add (4);
list.add (1);
list.add (3);
list.add (2);
Collections.sort(list, null);
System.out.println(Collections.binarySearch(list, 3));
Collections.sort(list,IntegerComparator);
System.out.println(Collections.binarySearch(list, 3));
}
}
```

What is the result?

A. 2

B. 1

C. 2

D. 2

E. 3

Correct Answer: A

QUESTION 6

Given the code fragment:

```
public class Test {
```

```
public static void main(String[] args) {
Path dir = Paths.get("D:\\company");
//insert code here. Line ***
for (Path entry: stream) {
System.out.println(entry.getFileName());
}
} catch (IOException e) {
System.err.println("Caught IOException: " + e.getMessage());
}
}
```

Which two try statements, when inserted at line ***, enable you to print files with the extensions.java, .htm, and .jar.

A. try (DirectoryStream stream = Files.newDirectoryStream(dir, "*. {java, htm, jar}")) {

B. try (DirectoryStream stream = Files.newDirectoryStream(dir, "*. [java, htm, jar]")) {

C. try (DirectoryStream stream = Files.newDirectoryStream(dir, "*. {java*, htm*, jar*}")) {

D. try (DirectoryStream stream = Files.newDirectoryStream(dir, "**. {java, htm, jar}")) {

Correct Answer: AD

"*. {java, htm, jar} and "**. {java, htm, jar} will match any file with file endings java, htm, or jar.

QUESTION 7

Given:

```
class Deeper {
public Number getDepth() {
return 10;
}
}
```

Which two classes correctly override the getDepth method?

A. public class deep extends Deeper { protected integer getDepth(){ return 5; } }

B. public class deep extends Deeper { public double getDepth() { return "5"; } }

C. public class deep extends Deeper {

public String getDepth () {

}

}

D. public class deep extends Deeper { public Long getDepth (int d) { return 5L; } }

E. public class deep extends Deeper { public short getDepth () { return 5; } }

Correct Answer: AE

Note: The abstract class Number is the superclass of classes Byte, Double, Float, Integer, Long, and Short.

Subclasses of Number must provide methods to convert the represented numeric value to byte, double, float, int, long, and short.

When class C extends B, we say that C is a "subclass" of B, and B is the "superclass" of C. This is called inheritence, because C inherited from B.

QUESTION 8

Given:

StringBuffer b = new StringBuffer("3");

System.out.print(5+4+b+2+1);

What is the result?

A. 54321

B. 9321

C. 5433

D. 933

E. Output is Similar to: 9java.lang.StringBuffer@100490121.

F. Compilation fails.

Correct Answer: F

The code will not compile.

The print function cannot handle the mixture of integers and strings.

Exception in thread "main" java.lang.RuntimeException: Uncompilable source code - Erroneous tree type

QUESTION 9

Which two are valid initialization statements?

A. Map m = new SortedMap();

B. Collection m = new TreeMap();

C. HashMap m = new SortedMap();

D. SortedMap m = new TreeMap ();

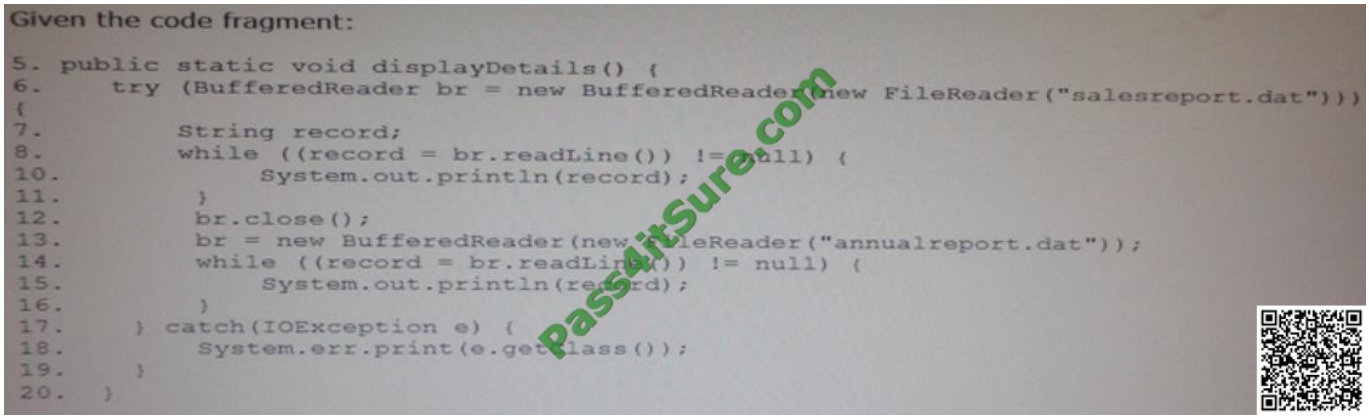
E. Hashtable m = new HashMap();

F. Map m = new Hashtable();

Correct Answer: DF

QUESTION 10

Given the code fragment:



What is the result, if the file salesreport.dat does not exist?

A. Compilation fails only at line 6

B. Compilation fails only at line 13

C. Compilation fails at line 6 and 13

D. Class java.io.IOException

E. Class java.io.FileNotFoundException

Correct Answer: D

Compilation works fine.

There will be a runtime error at line 6 (as the file salesreport.dat) does not exist.

This will be caught as an IOException at line 17.

QUESTION 11

A valid reason to declare a class as abstract is to:

A. define methods within a parent class, which may not be overridden in a child class

B. define common method signatures in a class, while forcing child classes to contain unique method implementations

C. prevent instance variables from being accessed

D. prevent a class from being extended

E. define a class that prevents variable state from being stored when object Instances are serialized

F. define a class with methods that cannot be concurrently called by multiple threads

Correct Answer: B

Note: An abstract method in Java is something like a pure virtual function in C++ (i.e., a virtual function that is declared = 0). In C++, a class that contains a pure virtual function is called an abstract class and cannot be instantiated. The same

is true of Java classes that contain abstract methods.

Any class with an abstract method is automatically abstract itself and must be declared as such.

An abstract class cannot be instantiated.

A subclass of an abstract class can be instantiated only if it overrides each of the abstract methods of its superclass and provides an implementation (i.e., a method body) for all of them. Such a class is often called a concrete subclass, to

emphasize the fact that it is not abstract.

If a subclass of an abstract class does not implement all the abstract methods it inherits, that subclass is itself abstract.

static, private, and final methods cannot be abstract, since these types of methods cannot be overridden by a subclass. Similarly, a final class cannot contain any abstract methods.

A class can be declared abstract even if it does not actually have any abstract methods.

Declaring such a class abstract indicates that the implementation is somehow incomplete and is meant to serve as a superclass for one or more subclasses that will complete the implementation. Such a class cannot be instantiated.

QUESTION 12

Given the code fragment:

```
public class Employee {

String name;

transient String companyName;

}
```

public class Manager extends Employee implements Serializable {

String mgrId;

public static void main(String s[]) throws Exception {

Manager mgr = new Manager();

What is the result?

A. M001, ,

B. M001, null, null

C. M001, Sam,

D. M001, Sam, null

E. M001, Sam, ABC Inc

F. Compilation fails

G. A NullPointerException is thrown at runtime

Correct Answer: E