



1Z0-805^{Q&As}

Upgrade to Java SE 7 Programmer

Pass Oracle 1Z0-805 Exam with 100% Guarantee

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

<https://www.passapply.com/1Z0-805.html>

100% Passing Guarantee
100% Money Back Assurance

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

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers





QUESTION 1

Which statement is true about the take method defined in the WatchService interface?

- A. Retrieves and removes the next watch key, or returns null if none are present.
- B. Retrieves and removes the next watch key. If a queued key is not immediately available, the program waits for the specified wait time.
- C. Retrieves and removes the next watch key: waits if no key is yet present.
- D. Retrieves and removes all pending events for the watch key, returning a list of the events that were retrieved.

Correct Answer: C

The WatchKey take() method retrieves and removes next watch key, waiting if none are yet present.

Note: A watch service that watches registered objects for changes and events. For example a file manager may use a watch service to monitor a directory for changes so that it can update its display of the list of files when files are created or deleted. A Watchable object is registered with a watch service by invoking its register method, returning a WatchKey to represent the registration. When an event for an object is detected the key is signalled, and if not currently signalled, it is queued to the watch service so that it can be retrieved by consumers that invoke the poll or take methods to retrieve keys and process events. Once the events have been processed the consumer invokes the key's reset method to reset the key which allows the key to be signalled and re-queued with further events.

Reference: Interface WatchService

QUESTION 2

Given:

```
ConcurrentMap partList = new ConcurrentHashMap ();
```

Which fragment puts a key/value pair in partList without the possibility of overwriting an existing key?

- A. `partList.put (key,"Blue Shirt");`
- B. `partList.putAbsent(key,"Blu Shirt")`
- C. `partList.putIfNotLocked (key,"Blue Shirt");`
- D. `partList.putAtomic (key,"Blue Shirt");`
- E. `if (!partlist.containsKey(key)) partList.put(key,"Blue Shirt");`

Correct Answer: E

The containsKey method returns true if this map maps one or more keys to the specified value.

So this statement adds a new key if they key is not present.

Reference: Class ConcurrentHashMap



QUESTION 3

Given:

```
import java.io.*;

public class SampleClass {

public static void main(String[] args) throws IOException {

try {

String dirName = args[0];

File dir = new File(dirName);

File.createTempFile("temp", "log", dir);

} catch (NullPointerException | IOException e) {

e = new IOException("Error while creating temp file");

throw e;

}

}

}
```

What is the result?

- A. Compilation fails.
- B. An IOException with the default message is thrown at runtime.
- C. An IOException with the message Error while creating temp file is thrown at runtime.
- D. A temporary file is created in the specified directory.

Correct Answer: A

The multi-catch parameter e may not be assigned. The compilation will fail at line: e = new IOException("Error while creating temp file");

QUESTION 4

Given the code fragment:

```
SimpleDateFormat sdf;
```

Which code fragment displays the three-character month abbreviation?



- A. `sdf = new SimpleDateFormat ("mm", Locale.UK); System.out.println ("Result:" + sdf.format(new Date()));`
- B. `sdf = new SimpleDateFormat ("MM", Locale.UK); System.out.println ("Result:" + sdf.format(new Date()));`
- C. `sdf = new SimpleDateFormat ("MMM", Locale.UK); System.out.println ("Result:" + sdf.format(new Date()));`
- D. `sdf = new SimpleDateFormat ("MMMM", Locale.UK); System.out.println ("Result:" + sdf.format(new Date()));`

Correct Answer: C

C: Output example: Apr

Note:SimpleDateFormat is a concrete class for formatting and parsing dates in a locale-sensitive manner. It allows for formatting (date -> text), parsing (text -> date), and normalization. SimpleDateFormat allows you to start by choosing any user-defined patterns for date-time formatting. However, you are encouraged to create a date-time formatter with either `getTimeInstance`, `getDateInstance`, or `getTimeInstance` in `DateFormat`. Each of these class methods can return a date/time formatter initialized with a default format pattern. You may modify the format pattern using the `applyPattern` methods as desired.

QUESTION 5

The current working directory is named finance.

Which two code fragments allow you to write the salary.dat file if it does not exist under "finance\payroll"?

- A.

```
public static void setFileContent (String[] s) throws IOException { path p=paths.get("payroll\salary.dat"); File file=p.toAbsolutePath().toFile(); try (BufferWriter br = new BufferWriter (new FileWriter(File))) { br.write ("experience new features of java"); } }
```
- B.

```
public static void setFileContent (String[] s) throws IOException { path p=paths.get ("payroll\salary.dat"); File file=p.toAbsolutePath(LinkOption.NOFOLLOW_LINKS).toFile(); try (BufferWriter br = new BufferWriter (new FileWriter(File))) { br.write ("experience new features of java"); } }
```
- C.

```
public static void setFileContent (String[] s) throws IOException { File file= new file ("payroll\salary.dat").getCanonicalFile(); try (BufferWriter br = new BufferWriter (new FileWriter(File))) { br.write ("experience new features of java"); } }
```
- D.

```
public static void setFileContent (String[] s) throws IOException { File file= new File ("payroll\salary.dat").getCanonicalFile(); try (BufferWriter br = new BufferWriter (new FileWriter(File))) { br.write ("experience new features of java"); } }
```
- E.

```
public static void setFileContent (String[] s) throws IOException { File file=new File ("payroll\salary.dat").getAbsolutePath(); try (BufferWriter br = new BufferWriter (new FileWriter(File))) { br.write ("experience new features of java"); } }
```

Correct Answer: BD

The problem in this scenario is how to construct a system-dependent filename from the string "payroll\salary.dat".

Regarding File-paths:

1- A file can have many relative paths.2- Canonical paths are absolute paths.3- An absolute path is not necessarily a canonical path! This holds true especially under Unix, which support symbolic links. Under Windows, an absolute path is usually a canonical path.



B: The absolute path can include symbolic links. Here we ignore them with NOFOLLOW_LINKS option.

D: The File.getCanonicalFile Method creates a new instance of a File object representing the file located at the absolute path of the current File object. All \\.\ and \\..\ references will be resolved.

QUESTION 6

What are two benefits of a Factory design pattern?

- A. Eliminates direct constructor calls in favor of invoking a method
- B. Provides a mechanism to monitor objects for changes
- C. Eliminates the need to overload constructors in a class implementation
- D. Prevents the compile from complaining about abstract method signatures
- E. Prevents tight coupling between your application and a class implementation

Correct Answer: AE

Factory methods are static methods that return an instance of the native class. Factory methods :

*

have names, unlike constructors, which can clarify code.

*

do not need to create a new object upon each invocation - objects can be cached and reused, if necessary.

*

can return a subtype of their return type - in particular, can return an object whose implementation class is unknown to the caller. This is a very valuable and widely used feature in many frameworks which use interfaces as the return type of static factory methods.

Note: The factory pattern (also known as the factory method pattern) is a creational design pattern. A factory is a JavaSW class that is used to encapsulate object creation code. A factory class instantiates and returns a particular type of

object based on data passed to the factory. The different types of objects that are returned from a factory typically are subclasses of a common parent class.

The data passed from the calling code to the factory can be passed either when the factory is created or when the method on the factory is called to create an object. This creational method is often called something such as getInstance or

getClass .

QUESTION 7

Given:



```
class Fibonacci extends RecursiveTask {
```

```
    final int n;
```

```
    Fibonacci (int n) { this.n = n }
```

```
    Integer compute () {
```

```
        if (n
```



To Read the [Whole Q&As](#), please purchase the [Complete Version](#) from [Our website](#).

Try our product !

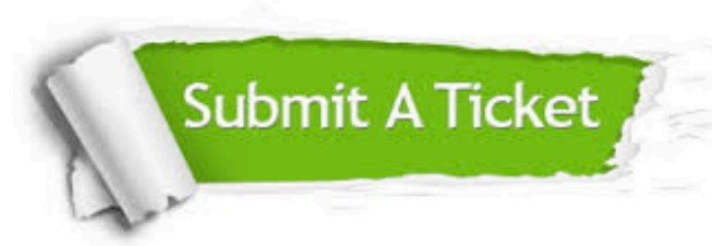
- 100% Guaranteed Success
- 100% Money Back Guarantee
- 365 Days Free Update
- Instant Download After Purchase
- 24x7 Customer Support
- Average 99.9% Success Rate
- More than 800,000 Satisfied Customers Worldwide
- Multi-Platform capabilities - [Windows](#), [Mac](#), [Android](#), [iPhone](#), [iPod](#), [iPad](#), [Kindle](#)

We provide exam PDF and VCE of Cisco, Microsoft, IBM, CompTIA, Oracle and other IT Certifications. You can view Vendor list of All Certification Exams offered:

<https://www.passapply.com/allproducts>

Need Help

Please provide as much detail as possible so we can best assist you.
To update a previously submitted ticket:



 <p>One Year Free Update Free update is available within One Year after your purchase. After One Year, you will get 50% discounts for updating. And we are proud to boast a 24/7 efficient Customer Support system via Email.</p>	 <p>Money Back Guarantee To ensure that you are spending on quality products, we provide 100% money back guarantee for 30 days from the date of purchase.</p>	 <p>Security & Privacy We respect customer privacy. We use McAfee's security service to provide you with utmost security for your personal information & peace of mind.</p>
---	---	--

Any charges made through this site will appear as Global Simulators Limited.
All trademarks are the property of their respective owners.
Copyright © passapply, All Rights Reserved.