Java Certified #2

A question lead guide to prepare Java certification

Working with Streams and Lambda expressions

Given:

```
StringBuffer us = new StringBuffer( "US" );
StringBuffer uk = new StringBuffer( "UK" );
Stream<StringBuffer> stream = Stream.of( us, uk );
String output = stream.collect( Collectors.joining( "-", "=", "" ) );
System.out.println( output );
```

What is the given code fragment's output?

- US-UK
- US=UK
- -US=UK
- =US-UK
- An exception is thrown
- Compilation fails

=US-UK

There's nothing wrong with the given code, implying no compile-time or runtime errors occur.

Among the three parameters of the `Collectors.joining` method, the first indicates the delimiter, while the second and the last represent the prefix and the suffix, respectively, of the joined string. At this point, it should be clear that option `=US-UK` is the correct answer.

https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/stream/Collectors.html#joining(java.lang.CharSequence,java.lang.CharSequence,java.lang.CharSequence)

Using Java I/O API

Classpath has Person.properties with name=James and

a Person extends ListResourceBundle returning { "name", "Jeanne" }.

ResourceBundle.getBundle("Person").getString("name") is printed.

What does it print?

➔ James

➔ Jeanne

➔ JamesJeanne

➔ JeanneJames

➔ MissingResourceException

➔ Compilation fails

Jeanne

The `getBundle` method first looks for a class file that matches the base name and the given locale. If it can;t find a class file, it then checks for properties files.

https://docs.oracle.com/javase/tutorial/i18n/resbundle/propfile.html

Handling Exceptions

What is the output, given:

```
class MyResource implements AutoCloseable {
    public void close() throws Exception {
        throw new IOException(); }}
```

Then I do:

```
try ( MyResource resource = new MyResource() ) {
    throw new RuntimeException();
} catch ( Exception e ) { System.out.println( e ); }
```

- ➔ java.lang.Exception
- ➔ java.lang.RuntimeException
- ➔ java.io.IOException
- ➔ Nothing
- ➔ Compilation fails

java.lang.RuntimeException

When dealing with a `try-with-resources` statement, if an exception is thrown from the `try` block as well as when closing a resource, the latter will be suppressed. Consequently, the exception caught by the `catch` block is `RuntimeException` .

In fact, this way of processing exceptions makes sense. After all, the real issue behind a failure is more important than problems with closing a resource.

https://docs.oracle.com/javase/tutorial/essential/exceptions/tryResourceClose.html https://bit.ly/javaOCP