wk4-Tools and Libraries for Everyone

Tuesday, December 15, 2020   11:23 AM

✓ **Congratulations! You passed!**
TO PASS 80% or higher

Keep Learning

GRADE
100%

Java: Tools and Libraries for Everyone

LATEST SUBMISSION GRADE

100%

1. What is the name of the method that starts a Java program?

   1 point

   ○ _start

   ○ firstMethod

   ○ begin

   ◉ main

2. What argument type should the method that starts a Java program take?

   1 point

   ○ an ArrayList<String>

   ○ no arguments

   ○ an int and a String

   ◉ an array of Strings

3. What effect comes from declaring a field "static"?

   1 point

   ○ The value in the field can only increase, never decrease.

   ◉ There is only one copy of that field for the entire class, not one per instance.

   ○ The value in the field cannot be changed once it is initialized.

   ○ The value in the field can only be changed by code inside the class.

4. In Java, how is the keyword "throw" used?

   1 point

   ○ To open a file for writing.

   ○ To transfer data across a network connection.

○ To handle an exception that has already occurred.

◉ To make an exception occur when the program detects a problematic circumstance that it cannot directly handle.

5. Creating a new socket, as with: 1 point

```
1   Socket s = new Socket(addr,port);
```

can throw an **IOException** according to the documentation of the **java.net.Socket** class.

Which of the following structures is the best way to create a socket while handling the exception?

◉
```
1 ▾ try {
2       Socket s = new Socket(addr, port);
3       //code that uses s
4   }
5 ▾ catch(IOException ioe) {
6       //code to handle the exception
7   }
```

○
```
1   Socket s = new Socket(addr, port);
2 ▾ try {
3       //code that uses s
4   }
5 ▾ catch(IOException ioe) {
6       //code to handle the exception
7   }
```

○
```
1 ▾ try {
2       Socket s = new Socket(addr, port);
3   }
4 ▾ catch(IOException ioe) {
5       //code that uses s
6   }
7   //code to handle the exception
```

○
```
1 ▾ try {
2       Socket s = new Socket(addr, port);
3   }
4 ▾ catch(IOException ioe) {
5       //code to handle the exception
6   }
7   //code that uses s
```

6. In Java, how is the keyword "finally" used?   1 point

○ To specify a piece of code to execute immediately after the end of the current loop.

○ To indicate that you are finally done debugging a particularly complicated piece of code.

◉ In exception handling to specify code that should be executed regardless of whether an exception happened or not.

○ To specify code to run when the program exits.

7. If you wanted to read the contents of a file without using the **edu.duke** package, you might call   1 point
Files.newBufferedReader.

What would you pass into **Files.newBufferedReader**?

○ a FileResource

○ a Comparator

⦿ a Path

○ an IOException

8. If you need to read data from a website without using the **edu.duke** package, you would probably want to use classes found in which package?   `1 point`

⦿ java.net

○ java.lolcats

○ java.internet

○ java.urls

9. You saw that a **BufferedReader** can be used to read data from a file on the local computer, as well as from a website. You could also use it with other sources of data, as long as you have an appropriate **Reader** class to access the data.   `1 point`

These capabilities are a great example of the benefits of which programming principles? Select the two best options.

☐ Write Robust Code: the BufferedReader class never throws an exception, instead it deals with every situation

☑ Abstraction: the BufferedReader class can work with any class that conforms to a specific interface, and does not need to know the details of how/where it reads data

☐ Astrachan's Law: you can only use the BufferedReader class to solve interesting problems, never for something boring or that you could do by hand

☑ Open/Closed Principle: the BufferedReader class is designed such that it can have its functionality expanded (to read from new data sources) without having to modify its code

☐ Everything Is A Number: the BufferedReader class can read data of any type, and return it as a Java object of an appropriate class