CAPCO

Scala Developer Test

Questions

Version 1.0

July 2020

Question 1

SCALA · EXCEPTIONS

Description:

Consider the following code snippet:

```
def readFirstLine(path: String): Try[String] = {
    Try {
            var file = Source.fromFile(path)
            var buffer = file.getLines().next
            file.close
            buffer
    }
}
```

Question: Which statements are correct if *Source.fromFile()* throws a *FileNotFoundException* exception?

Please select the correct statement(s) below

#	Statement
A	The variable *buffer* will be assigned to the result of *file.getLines().next*
B	The *file.close* function will be called
C	The function will return *Failure(FileNotFoundException)*
D	The function will throw *FileNotFoundException exception*
E	println(readFirstLine("") getOrElse "Error!")

Question 2

SCALA · STRINGS · PREMIUM

Description:

Convert a string of numbers to a sentence. Each number represents a letter. Numbers in the string are separated by a space and words in the sentence are separated by a plus character.

Conversion table:

1 = A
2 = B
...
26 = Z

Example: *numbersToLetters("20 5 19 20+4 15 13 5")* should return "TEST DOME".

Question 3

SCALA · LAZY EVALUATION · LAZYLIST

Description:

Implement the *allPrefixes* method so that, in the given set of words, it finds prefixes whose length is equal to *prefixLength* and returns all distinct prefixes. Words contain only lowercase letters. Words shorter than the required length should be ignored. The method must maintain its deferred execution, so do not evaluate elements before they are needed (e.g. toArray).

For example, the following code snippet:

```
val words = "flow" #:: "flowers" #:: "flew" #:: "flag" #:: "fm" #:: LazyList.empty
Prefix.allPrefixes(3, words).foreach(println)
```

should print "flo", "fle", and "fla" (in any order).

Question 4

SCALA · INHERITANCE · OOP

Description:

Create a class *LanguageStudent* with:

- public method *getLanguages(): Iterable[String]* - returns all languages that the student knows
- public method *addLanguage(language: String)* - adds a new language to the list of languages

Create a class *LanguageTeacher* that inherits *LanguageStudent* and has one additional public method:

- *teach(student: LanguageStudent, language: String): Boolean* - if *LanguageTeacher* knows the required language, it teaches *LanguageStudent* and returns true; otherwise, it returns false

For example, the following code shows how *LanguageTeacher* teaches *LanguageStudent* the new language ("English"):

```
val teacher = new LanguageTeaching.LanguageTeacher
teacher.addLanguage("English")

val student = new LanguageTeaching.LanguageStudent
teacher.teach(student, "English")

for(language <- student.getLanguages())
System.out.println(language);
```

Question 5

SCALA · ARITHMETIC · BUG FIXING

Description:

The code works correctly for the given example, but users have complained that it doesn't work for some of their cases. Users also expressed a preference for returning exceptions rather than throwing exceptions. Fix the bugs in the code and implement the users' preference.

The function *mean* should return ***Try[Double]***. The function should return the weighted mean as *Double* for valid inputs or an *Exception* for any illegal argument value or combination. The expected arguments are two arrays of *Int* with the same length.

For example, *WeightedAverage.mean* called with numbers *{3, 6}* and weights *{4, 2}* should return the weighted mean (3 * 4 + 6 * 2) / (4 + 2) = 4.0