Stream Programming - 2019
Laboratory 1

To get a standard Scala installation, go to http://www.scalalang.org/downloads and follow the directions for your platform. Remember that the Java virtual machine must be installed before. You can also use a Scala plugin for Eclipse, IntelliJ or NetBeans.

The easiest way to get started with Scala is by using the Scala interpreter, an interactive śhell"for writing Scala expressions and programs. Simply type an expression into the interpreter and it will evaluate the expression and print the resulting value. The interactive shell for Scala is simply called scala. You use it by typing scala at a command prompt:

```
> scala
Welcome to Scala version 2.8.1.
Type in expressions to have them evaluated.
Type :help for more information.
scala>
```

To do the following tasks, the basic skills of java or C++ programming is needed.

Task 1 Create a text file of the name `Program.scala` with the following content:

```
object HelloScala {
    def main(args: Array[String]): Unit = {
        println("Hello World!")
    }
}
```

Then, compile it by `scalac Program.scala`. Check the file created as a result. Run the program using `scala Program`.

Task 2 The `main` method of a scala program has the parameter `args` which is an array of strings. It enables to invoke the program with various parameters. The number of elements in a parameter `args` can be determined by the method `length`. You can use `for` or `foreach` loop.

Write a program which prints in the subsequent lines the parameters. For example, the program `ScalaParameters` invoked with parameters: `-24 ggh "Hello world"56` should work as follows:

```
> scala ScalaParameters -24 ggh "Hello world" 56
-24
ggh
Hello world
56
```

Task 3 Write a program which prints the greatest divisor for all parameters of the program. In the program, you should create an object with the method `getDivisor`. It should have one parameter of the type `Int` where the number should be provided. It must return the greatest divisor of this number.

The program should work as follows:

```
> scala GreatestDivisor 24 13 50 46 9 abd
24 : 12
13 : 1
50 : 25
46 : 23
9 : 3
abd: the conversion is not possible
```

For conversion from string to Int, the instructions: `try, catch` can be used (see hints below). If the conversion of the argument is not possible, the `NumberFormatException` should be caught.

Hint 1 An example of the program which can throw and catch exceptions:

```
import java.io.FileReader
import java.io.FileNotFoundException
import java.io.IOException

object Demo {
   def main(args: Array[String]) {
      try {
         val f = new FileReader("input.txt")
      } catch {
         case ex: FileNotFoundException =>{
            println("Missing file exception")
         }
         case ex: IOException => {
            println("IO Exception")
         }
      }
   }
}
```

Hint 2 The string `foo` converted to `Int` throws `NumberFormatException` like this:

```
scala> "foo".toInt
java.lang.NumberFormatException: For input string: "foo"
    at java.lang.NumberFormatException.forInputString(NumberFormatException.java)
    at java.lang.Integer.parseInt(Integer.java:449)
    ... more output here ...
```