CS320 Programming Languages
README

We use the programming language Scala to implement interpreters of various programming languages in this course. All the programming homework run on sbt, which is the interactive build tool for Scala.

1 Installation

First, install Scala and sbt by following the instruction in the site:

https://www.scala-lang.org/download/

You can use IntelliJ or other IDEs but we recommend you to use terminal to build and run the homework project. How to use sbt in termainal is described in Section 3.

2 Structure

The homework project consists of the following directories and files:

```
cs320
├─ .idea
├─ build.sbt
├─ project
├─ README.pdf
└─ src/main/scala/cs320
   ├─ Homework.scala
   └─ hw01
      ├─ Homework01.scala
      ├─ hw01.pdf
      └─ package.scala
```

The .idea file contains the project information helpful when you use IntelliJ. The build.sbt and project files are related to the sbt configuration. The src/main/scala/cs320 directory contains all the source codes of homework. The Homework.scala file provides common utility functions that will be used in homework. The hw01 directory contains all the information of Homework #1.

From now, we will give you programming homework as the directory hwXX, such as, hw01 or hw05. Then, you just copy and paste it into the src/main/scala/cs320 directory. Each programming homework consists of three files: hwXX.pdf explains what you should do, HomeworkXX.scala describes the specification, package.scala is the main file that you should implement.

3 sbt Console

You can build and run your implementation using sbt console system. On the top of the project directory(i.e. cs320), just type the sbt console:

```
$ sbt console
[info] Loading global plugins from /Users/naldo/.sbt/0.13/plugins
[info] Loading project definition from /Users/naldo/project/cs320-homework/cs320/project
[info] Set current project to cs320 (in build file:.../cs320/)
[info] Starting scala interpreter...
[info]
Welcome to Scala 2.12.8 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_152).
Type in expressions for evaluation. Or try :help.

scala>
```

If you want to load Homework #1, then just type import cs320.hw01._. Then, you can use any functions or variables defined in hw01/package.scala. If you want to test some functionalities, we recommend you write some test cases in the tests function using test or testExc functions. The test function checks whether given two arguments are same and testExc function checks whether the first argument throws an error with a message containing the second argument string. If it fails, it prints out which tests are failed and their positions based on line numbers.

```
scala> import cs320.hw01._
import cs320.hw01._

scala> test(1,1)

scala> test(1,2)
FAIL [<console>:15]: 1 is not equal to 2

scala> testExc(error("abcd"), "a")

scala> testExc(error("abcd"), "e")
FAIL [<console>:15]: "[ERROR] abcd" does not contain "e"

scala> tests
FAIL [package.scala:21]: an implementation is missing
FAIL [package.scala:22]: an implementation is missing
FAIL [package.scala:23]: an implementation is missing
FAIL [package.scala:24]: an implementation is missing
FAIL [package.scala:25]: an implementation is missing
FAIL [package.scala:26]: an implementation is missing
FAIL [package.scala:27]: an implementation is missing
FAIL [package.scala:28]: an implementation is missing
FAIL [package.scala:29]: an implementation is missing
FAIL [package.scala:30]: an implementation is missing

scala>
```

If you want to see the cases that successfully pass tests, turn off the alertOnlyFail option:

```
scala> alertOnlyFail = false
alertOnlyFail: Boolean = false

scala> test(1,1)
PASS [<console>:15]
```

```
scala> test(1,2)
FAIL [<console>:15]: 1 is not equal to 2

scala>
```

We recommend you write enough test cases before implementation. It helps you consider the corner cases and organize your thoughts.

4 Submission

Please submit your solution into the homework server until the due date:

`https://kaist-cs320.appspot.com/`

You should upload only `package.scala` file into the server for programming homework.

5 CAUTION

There are several rules you should follow:

- DO NOT use mutable variables; `var x = ...`
- DO NOT use loop; `while(...){...}` or `do{...}while(...)`