Practice Programming Assignment: Example

You have not submitted. You must earn /10 points to pass.

ⓘ It looks like this is your first programming assignment. Learn more ✕

Deadline Pass this assignment by April 7, 11:59 PM PDT

Instructions

My submission

Discussions

Getting Started!

The goal of this assignment is to familiarize yourself with the infrastructure and the tools required during this class. Even though the grade in this assignment will be excluded from your final grade for the course, it is important that you work through this assignment carefully.

Installing Required Tools

Before anything else, it is crucial that you make sure that all tools are correctly installed. We will primarily support Java 8.

Part 1: Obtain the Project Files

Download the example.zip handout archive file and extract it somewhere on your machine.

Part 2: Using the Scala REPL

During this class we will always launch the Scala REPL (the interactive Scala console) through sbt. This way you don't need to install the Scala distribution on your machine, having sbt is enough. (In case you prefer to have the scala command available on your machine, you can download the Scala distribution from the scala-lang.org website.)

Open the Sbt Tutorial page and follow the first steps up to (and including) "Running the Scala Interpreter". Note that sbt can only be started inside a project directory, so first navigate to the project directory that you created in Part 1.

Enter a few Scala expressions to make sure everything is working as expected.

Part 3: Opening the Project

To work on the source code of the project, you can use an IDE. You may choose between Intellij and Eclipse. Follow the description on the Intellij Tutorial or the Eclipse Tutorial page and take a first look at the source code.

In the folder src/main/scala, open the package example and double-click the file Lists.scala. There are two methods in this file that need to be implemented (sum and max). When working on an assignment, it is important that you don't change any existing method, class or object names or types. When doing so, our automated grading tools from will not be able to recognize your code and you have a high risk of not obtaining any points for your solution.

Part 4: Running your Code

Once you start writing some code, you might want to experiment with Scala, execute small snippets of code or also run some methods that you already implemented. We present two possibilities to run Scala code.

Note that these tools are recommended for exploring Scala, but should not be used for testing your code. The next part of this assignment will explain how to write tests in Scala.

Using the Scala REPL

In the sbt console, start the Scala REPL by typing **console**:

```
> console
[info] Starting scala interpreter...
scala>
```

In the REPL you can try out arbitrary snippets of Scala code, for example:

```
scala> val l = List(3,7,2)
l: List[Int] = List(3, 7, 2)

scala> l.isEmpty
res0: Boolean = false

scala> l.tail.head
res1: Int = 7

scala> List().isEmpty
res2: Boolean = true
```

The classes of the assignment are available inside the REPL, so you can for instance import all the methods from object **Lists**:

```
scala> import example.Lists._
import example.Lists._

scala> max(List(1,3,2))
res1: Int = 3
```

In order to exit the Scala REPL and go back to sbt, just type **ctrl-d**.

Using a Main Object

Another way to run your code is to create a new **Main** object that can be executed by the Java Virtual Machine.

1. In eclipse, right-click on the package **example** in *src/main/scala* and select "New" - "Scala Object"
2. Use **Main** as the object name (any other name would also work)
3. Confirm by clicking "Finish"

In order to make the object executable it has to extend the type **App**. Change the object definition to the following:

```
object Main extends App {
  println(Lists.max(List(1,3,2)))
}
```

Now the **Main** object can be executed. In order to do so in eclipse:

1. Right-click on the file **Main.scala**
2. Select "Run As" - "Scala Application"

You can also run the **Main** object in the sbt console by simply using the command **run**.

Part 5: Writing Tests

Throughout the assignments of this course we will require you to write unit tests for the code that you write. Unit tests are the preferred way to test your code because unlike REPL commands, unit tests are saved and can be re-executed as often as required. This is a great way to make sure that nothing breaks when you have go back later to change some code that you wrote earlier on.

We will be using the ScalaTest testing framework to write our unit tests. In eclipse, navigate to the folder src/test/scala and open the file **ListsSuite.scala** in package **example**. This file contains a step-by-step tutorial to learn how to write and execute ScalaTest unit tests.

Part 6: Submitting your Solution

Once you implemented all the required methods and tested you code thoroughly, you can submit it to Coursera. The only way to submit your solution is through sbt, so you need to start the sbt console in your project directory.

In order to submit, you need to have your coursera username and your submission password. Note that the submission password is NOT your login password, instead it is a special password generated by Coursera. You can look it up on the right side of the assignments page. If you don't already have a token click on "Generate new token", then **"I don't want to verify" and "Continue without verification".**

Submitting in sbt is simply done by invoking the **submit** task:

```
> submit your.email@domain.com submissionPassword
[info] Connecting to coursera. Obtaining challenge...
[info] Computing challenge response...
[info] Submitting solution...
[success] Your code was successfully submitted: Your submission has been
    accepted and will be graded shortly.
[success] Total time: 2 s, completed Aug 30, 2012 4:30:10 PM>
```

Once you submit your solution, you should see your grade and a feedback about your code on the Coursera website within 10 minutes. If you want to improve your grade, just submit an improved solution. The best of all your submissions will count as the final grade.

How to submit

Copy the token below and run the submission script included in the assignment download. When prompted, use your email address **joel.vallone@gmail.com**.

BIANleJcdpjn4Wln

Generate new token

Your submission token is unique to you and should not be shared with anyone. You may submit as many times as you like.