Practice Programming Assignment: Example Assignment

✔ Passed · 1/1 points

Deadline Pass this assignment by June 26, 11:59 PM PDT

Instructions

My submission

Discussions

Attention: You are allowed to submit **an unlimited number of times**. Once you have submitted your solution, you should see your grade and feedback about your code under the "My Submission" tab above, within 10 minutes. If you want to improve your grade, just submit an improved solution. The best of all of your submissions will count as the final grade. **(This assignment does not count towards your grade anyways, its just to teach you the workflow for submitting assignments, but this is the policy for all other assignments.)**

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 most important that you make sure that all tools are correctly installed.* Take a very careful look at the Tools Setup page and verify that all of the listed tools work on your machine.

The Assignment

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 until 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 in Eclipse

To work on the source code of the project, you have to import it into eclipse. Follow the description on 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*. Doing so will prevent our automated grading tools from working 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 run your code on a few examples to see if it works correctly. We present two possibilities to run the methods you implemented.

Using the Scala REPL

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

```
1  > console
2  [info] Starting scala interpreter...
3
4  scala>
```

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

```
1   scala> import example.Lists._
2   import example.Lists._
3
4   scala> max(List(1,3,2))
5   res1: Int = 3
```

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:

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

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 **sbt** 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.

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

```
1  > submit your.email@domain.com submissionPassword
2  [info] Connecting to coursera. Obtaining challenge...
3  [info] Computing challenge response...
4  [info] Submitting solution...
5  [success] Your code was successfully submitted: Your submission has been
      accepted and will be graded shortly.
6  [success] Total time: 2 s, completed Aug 30, 2012 4:30:10 PM
7  >
```

You are allowed to **resubmit an unlimited number of times**! 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.