Computer Science C24
University of Toronto Scarborough

Homework Exercise # 6

Your Task

Your task is to produce a lesson on a topic selected from the list of topics below. It is important to have an appropriate level of sophistication of your lesson. Remember that your target audience consists of people who earned an A+ in CSCC24! They have thorough knowledge of all the concepts and practical aspects of programming languages that we cover in the course.

Working in Teams

You can work on this exercise either individually or in a team of 2. Choose your own team.

Academic Integrity

The lesson you submit must be designed and implemented by you and you alone. Looking up a lesson online and "summarizing" it would constitute an Academic Offence. You must use at least three different sources to learn the material for your lesson, and you must properly cite your sources. All example code must be yours and yours alone.

For guidelines on how to make sure you do not commit an academic offence, take a look at How Not to Plagiarize.

Format and Handing In

You have the following options for presenting your lesson: HTML or markdown (equivalent of 3-4 typed up letter-sized pages) or a video (7-8 minutes).

Regardless of which format option you choose, your lesson should be well-organised, should have an introduction and a conclusion, and should have at least 2 examples for every concept defined / presented. You should also make a clear connection to our course material.

If your lesson is in HTML or markdown format, please upload it to MarkUs. If you record a video, please upload it to MyMedia (make sure it is publicly available) and also upload a text file with the link to MarkUs.

In addition, you need to submit all the code you use in your examples, together with a README file that clearly explains how to run the code.

Marking

This exercise counts for 4% of the course grade. Your work will be marked based on correctness and level of sophistication of the content, as well as organisation and presentation.

Make sure that everything you state is correct, and pay very careful attention to the terminology you use in your lesson.

List of Topics

- Continuations in Racket.
- Using (equivalents of) lambda, map, filter, and fold in Scala.
- Using (equivalents of) lambda, map, filter, and fold in C++.
- Polymorphic methods and generic classes in Scala.
- Generic functions and generic classes in C++.
- Variance, Covariance, and Contravariance with generic classes in Scala (if you like theory).
- Introduction to (untyped) Lambda Calculus (if you like theory).

If you have a topic you would very much like to explore for this project which is not on this list, please, contact me ASAP and we will see if we can accommodate your interest.