

TIPLPA Programming Language Paradigms

Multi-Paradigm Assignment

Group Registration: 12:00, 8 May 2014
CampusNet Submission: 23:59, 26 May 2014
Class Demonstration: 27 May 2014
Version 1

All tasks except Task 5 are group tasks.

Task 1 — Registration

Register your group by emailing `jwc@eng.au.dk` with the list of the group members and their emails. You must register by 14:00 on 6 May 2013. Joey will acknowledge this with a group number.

This assignment is to be done in groups of 3; if necessary, we may allow groups of 2, but only if the numbers are such that you cannot be evenly divided into groups of 3.

Task 2 — Base Application

Your task is to create a graphing application for mathematical functions. The general idea is that the user will type in an arbitrary, continuous function into an input box and then have its graph be displayed in a window. The function will be input as Scheme procedure, e.g. `(lambda (x) (* x x))`. You do not need to create a parser to translate “normal” mathematical notation to Scheme.

The function a user might enter must be side-effect free, but otherwise it may do anything as long as it conforms to the definition of a continuous function over numbers. It must be able to deal with real numbers, but it does not need to handle complex numbers.

The user must be able to specify certain parameters, such as the region that the graph will display (minimum and maximum coordinates for x and y), whether the scales are linear or logarithmic, and how many data points should be plotted.

All of the mathematical calculation must be done in Scheme, and all of the interface/GUI elements must be done in Java. So you should end up with an application which has a Scheme “engine” and a Java interface. (Note: we don’t expect a *pretty* interface, just a usable one.)

The function must be treated as an opaque string in the Java runtime, and all interpretation of the meaning of that function must be handled by the Scheme runtime.

Task 3 — Extensions

Multiple Functions The application must be extended to allow for multiple functions to be displayed on the same chart, preferably in different colours.

Derivatives The application must be extended to allow the user to select that both the function and its derivative be plotted on the same chart.

Integration The application must be extended to display for the user the area under the curve for a given range of a function; i.e. it must compute an approximation of the definite integral of the user’s function. Use the “rectangle method”, and be sure to allow the user to specify the number of rectangles used to approximate the area.

Task 4 — Testing

Your application must be tested, both the Scheme and Java portions. Explain your strategy and be prepared to show some of the tests during the demo.

Task 5 — Report (Individual Effort)

Submit a report that describes the application on focuses on three things about the application: its architecture; its functionality; and the testing strategy used. You must individually demonstrate that you understand the operation of the application, in particular the multi-paradigm aspects.

The report is to be a maximum of 6 pages long, everything included. (A formal cover page is not required, but ensure that your name and group number are at the top of the document.)

Task 6 — Demonstration

On 27 May you will be expected to demonstrate your application in class. Attendance is obligatory for the assignment to be accepted.