

Introduction to programming for animal sciences

ECTS credits:

5

Course parameters:

Language: English

Level of course: PhD course

Semester/quarter: Two week course 7-21 April/Q4

Hours per week: 36

Capacity limits: 20

Objectives:

The main objective is to teach students the basic logics of algorithms and programming structure.

Learning outcomes:

After the course, the students should be able to:

- Identify the detailed logical steps involved in solving simple programming task
- Devise algorithms to solve simple programming tasks
- Implement simple algorithms in their language of choice
- Deconstruct a problem into small and manageable sub-problems
- Assess in an empirical way and compare the performance of different ways of solving a particular problem

Compulsory programme:

Course participation and completion of assignments.

Course contents:Module 1 – Introduction

1. Introduction to course: motivation;
2. Constructs vs. algorithms;
3. Variable/objects classification: understand the differences between integer, real, double, logical and character;
4. Comparing values according to type;
5. Conversion between types;
6. Introduce vectors (concept and usage) and factors (R type)
7. Boolean operators and statements;
8. Subset data given a Boolean statement.

Module 2 – Loops I

1. Simple loops ('for' loops): limited number iterations;
2. Iterative computation;
3. Nested loops.

Module 3 – Loops II/Conditions

1. Many operations inside a loop, logical order that operations should take place;
2. Loops determined by some criteria ('while' loops);
3. 'If/else' statements: write 'if/else' flowcharts, predict the outcome of 'if/else' statements;
4. Use of 'if/else' statements inside loops.

Module 4 – Reading and writing files

1. Files paths: working directory, absolute vs. relative paths;
2. Opening files: reading, writing, appending;
3. Regular and formatted output;
4. Reading formatted data;
5. Processing one line at a time;
6. Reading directories;
7. Closing errors;
8. Error checking.

Module 5 – Matrices and vectors

1. Basic operations with vectors and matrices: , adding, dot product, matrix product;
2. Handling data frames: read and write.

Module 6 – Mathematical issues

1. Missing values;
2. Mathematical operations that are of indeterminate form: problems and solutions;
3. Rounding;
4. Differences between large numbers;
5. Random numbers and seed.

Module 7 – Strings

1. What is a string;
2. Comparing strings;
3. Strings replacements;
4. Splitting and joining strings.

Module 8 – Functions

1. Code reuse;
2. Create functions;
3. Function elements: name, arguments, body, return value;
4. Within function variables and global variables;
5. Returning success/error;
6. Using functions inside a loop;
7. Recursion.

Module 9 – Optimization and debugging

1. Assessing the performance of a program;
2. Tools to reduce running time;
3. Memory cost of copying;
4. Improving loops;
5. Debugging.

Module 10 – Sorting algorithms and final considerations

1. What is sorting;
2. A naïve sorting algorithm;
3. Developing and implementing sorting algorithms;
4. Final considerations: the importance of comments, indentation and organization in a code.

Prerequisites:

Basic knowledge of matrix algebra.

Name of lecturers:

Beatriz Cuyabano and Bernt Guldbrandtsen

Type of course/teaching methods:

Two week intensive course including lectures, computer exercises and assignments.

Course format:

The course will be divided into 10 modules of 3 hours teaching each. These 3 hours will be divided in 2 hours lecture and 1 hour practical/computer exercise. At the end of every two modules, the students will be given an assignment summarizing the topics and skills taught in all completed modules. All assignments, except the final, are expected to require 7 hours of work each. The final assignment is expected to require 14 hours of work. The modules will be taught two by two, every other day, during two consecutive weeks as described below, summing up to a total workload of 72 hours:

Course plan:

April 7: Module 1 and 2
April 8: Assignment
April 9: Module 3 and 4

April 10: Assignment
April 13: Module 5 and 6
April 14: Assignment
April 15: Module 7 and 8
April 16: Assignment
April 17: Module 9 and 10
April 20: Assignment
April 21: Assignment

Course assessment:

Satisfactory completion of assignments

Provider:

Center for Quantitative Genetics & Genomics (QGG), Dept. of Molecular Biology and Genetics

Special comments on this course:

You need to bring your own laptop with R (<http://www.r-project.org/>), Rstudio (<http://www.rstudio.com/>) installed and Eduroam working.

Time:

7-21 April 2015

Place:

AU-Foulum, lecture room, building K21

Registration:

By e-mail to Beatriz.cuyabano@mbg.au.dk. Deadline 13 March 2015.

When registering, please indicate level of programming experience (none, minor, some experience, intermediate).

If you have any questions, please contact Beatriz Cuyabano (Beatriz.cuyabano@mbg.au.dk)