POZNAN UNIVERSITY OF TECHNOLOGY

EUROPEAN CREDIT TRANSFER AND ACCUMULATION SYSTEM (ECTS)

pl. M. Skłodowskiej-Curie 5, 60-965 Poznań

COURSE DESCRIPTION CARD - SYLLABUS

Course name

Scripting Languages in Bioinformatics

Course

Field of study	Year/Semester
Bioinformatics	2/3
Area of study (specialization)	Profile of study
	general academic
Level of study	Course offered in
First-cycle studies	Polish
Form of study	Requirements
full-time	compulsory

Number of hours

Lecture	Laboratory classes	Other (e.g. online)
15	30	
Tutorials	Projects/seminars	

Number of credit points

4

Lecturers

Responsible for the course/lecturer:	Responsible for the course/lecturer:
dr eng. Maciej Miłostan	
Wydział Informatyki i Telekomunikacji	

Prerequisites

The student starting this module should have basic knowledge of the paradigms of structured and object-oriented programming, know at least one object-oriented and structured programming language (preferably C ++ / C). He or she should be familiar with data structures, including complex ones, basic algorithms for solving combinatorial problems and their complexity. Should have the ability to solve basic problems. Should have the ability to search for information.

Course objective

1. Provide students with basic knowledge about operating systems to the extent that they can use Linux freely and automate tasks using the shell's mechanisms and scripting languages.

2. Provide students with knowledge of Python and development environments to the extent allowing implementation of classess and libraries by themselves.

3. To acquaint students with the practical use of scripting languages for biological analyzes on the example of Biopython.

4. Developing students' skills in solving problems that require processing biological data from various sources (including databases) - the ability to convert data. The skills include processing of character sequences and text files using regular expressions.

5. Teaching students the ability to create their own tools following the given specification.

Course-related learning outcomes

Knowledge

As a result of the conducted classes, the student:

1. Extends knowledge on structured and object-oriented programming in the context of Python interpreted languages

2. Gets basic knowledge on the practical use of operating systems and the mechanisms built into them. He is familiar with the basic data processing and automation mechanisms available in Linux (pipelining, basic shell scripts, I / O redirection).

3. Knows the methods of accessing databases and files using Python.

4. Has a well-established and theory-based knowledge of character sequence processing, in particular with the use of regular expressions and patterns. He is familiar with string search operations, string modification, text conversion and formatting, encoding (ASCI, UTF8, BASE64), aggregate / statistical data extraction. Can use various text data sources (files, data bases etc.)

5. Knows bioinformatics libraries and is able to use them for processing biological sequences and structures of proteins and nucleic acids.

6. Knows how to implement custom libraries and how to use them.

Skills

As a result of the conducted classes, the student:

1. Applies basic techniques and IT tools, available in Python libraries, for the analysis of biological data in the form of text files with amino acid sequences and structural data (PDB text files). He or she can retrieve data from biological databases using Python programs. He or she can perform basic transformations of biological sequences and basic operations on PDB and fasta files. These skills are essential for analyzing complex biological problems.

2. Can use functions, programming structures, libraries and classes available in Perl and Python to implement own tools that achieve goals in accordance with the given specification.

Social competences

1. Can properly define the priorities for the implementation of the task set by himself or others. The necessary element to pass is the timely implementation of a number of practical tasks, in particular programming.

2. Is aware of the responsibility for the decisions made - deficiencies in the implementation of tasks, untimely execution or attempts to plagiarize them affect the obtained grades.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

The knowledge acquired during the lecture is verified by means of a 60-minute test carried out during the 15th lecture. The test consists of 25-30 questions (test and open-ended), with different scores. Passing threshold: 50% of points. Final topics list, on the basis of which the questions are developed, will be sent to students by e-mail using the university's e-mail system.

The skills acquired during the laboratory classes are verified on the basis of: reports from labs; one session in front of the computer, consisting of 5 tasks with different scores depending on the degree of their difficulty; and the implementation of the Python program supporting selected bioinformatics data analyzes. Passing threshold: 50% of points.

Programme content

The lecture program covers the following topics.

Lecture 1-2: Biological data, sequences and texts - outlining the context and indicating the areas in which scripting languages are used with great success. Highlighting the importance of sequence data in biology and identifying elementary tools that can be used for their analysis. Brief introduction to linux instructions and an introduction to scripting languages and interpreters, showing the elements of shell and AWK programming.

Lecture 3-5: Programming paradigms and good programming practices in the context of Python. Setup of the development environment. Discussion of language syntax, notation and its philosophy. Discussion of the basic data types and the idea of objectivity in Python analogous to lectures 3-5. Indicating the methods of code documentation and methods of using it. The evolution of the language and the fundamental differences in functionality between the two major versions of Python (Python 2 and Python 3). Defining your own classes and the mechanism of inheritance.

Lecture 6-8: Analysis of data using bioinformatics libraries including Biopython Biological Library. Implementation of basic GUI using builtin Tkinter library. Final test.

Laboratory exercises are conducted in the form of fifteen two-hour classes taking place in the computer laboratory. The first classes are intended to familiarize students with the rules of using the laboratory and completing the exercises. Classes are carried out by students individually. The laboratory program covers the following topics. Introduction to working in Linux - basic instructions, pipeline processing, redirection of input and output streams. Shell programming components - bash scripts, word processing components using AWK programs. Python interpreter, and its help and documentation system that is a part of the development environment. Implementation of programs aimed to increase students' familiarity with data types and language control structures. Acquainting with mechanisms of regular expressions and their application to data analysis in text files, including files with sequences and biological structures (PDB files). Conducting tests of the implemented programs on the test data sets.

Implementation of programs acquiring data from publicly available biological databases and transforming them in accordance with the given specification using the Biopython libraries. Design and implementation of your own class that performs a set of operations on biological data in particular PDB and fasta files

Teaching methods

1. Lecture: multimedia presentation as needed, illustrated with additional examples given on the blackboard

2. Laboratory exercises: practical exercises at the computer carried out according to the given scenario, the configuration of programs and scripts solving shared problems, discussion of applied solutions and programming structures

Bibliography

Basic

1. Bioinformatics Programming using Python / Mitchell L Model; O'Reilly 2010

2.Learning Python, 5th Edition Python. (in polish: Wprowadzenie.Wydanie V) / Mark Lutz; O'Reilly, 2020 [Translation: Grzegorz Kowalczyk, Andrzej Watrak, Anna Trojan, Marek Pętlicki, Helion 2020].

3. Python 3 : kompletne wprowadzenie do programowania / Mark Summerfield ; [tł. Robert Górczyński]., Gliwice, Helion 2010.

4. Linux Pocket Guide, 2nd Edition, Daniel J. Barrett, O’Reilly Media 2012 (Dostępne tłumaczenie na język polski: Linux. Leksykon Kieszonkowy, Tł. Adam Bąk, Helion 2013)

5. The Linux Command Line: A Complete Introduction, William E. Shotts, Jr., No Starch Press, 2012 (dostępna również jako darmowy ebook: http://linuxcommand.org/tlcl.php)

6. System operacyjny LINUX- przewodnik użytkownika, Cezary Sobaniec, Nakom 2002

7. Linux in a Nutshell, 6th Edition, Ellen Siever, Stephen Figgins, Robert Love, Arnold Robbins, O’Reilly Media 2009

Additional

1. Mastering Regular Expressions, Jeffrey E.F. Friedl, O'Reilly Media, 2006

2. Sed & Awk, Dougherty and Arnold Robbins, O'Reilly and Associates, 19977

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,0
Classes requiring direct contact with the teacher	45	2,0
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) [1]	55	2,0

[1] delete or add other activities as appropriate