Academia de Studii Economice
Departamentul de Informatică şi Cibernetică Economică
Calea Dorobanţi, 15-17, Sector 1, Bucureşti, 010552 (camera 2314)
Tel.: +40 21 319 19 00, ext. 319, 336, Fax: +40 21 311 20 66
www.dice.ase.ro

Contest Topics for Professor
Position 27, year 2021-2022, semester 2

Disciplines: Software Development for Data Analysis; Multiparadigm Programming - Java

Software Development for Data Analysis:

1. Python basics. The structure of Python programs. Operators. Variables and objects. Control structures and functions in Python. Data types and structures in Python. Working with files in Python;
2. Matrix calculation using Numpy.Scientific calculus in Python. Scipy Library;
3. Using the Pandas library. Series, DataFrame and Categorical classes;
4. Using the Pandas library. Creation and operation of data tables - selection, summary, filtering, data aggregation;
5. View data using the matplotlib and seaborn libraries. Visualization of spatial data in Python using the Geopandas, Bokeh and Pysal libraries;
6. Preliminary data analysis. Fundamentals. Probability distributions. Concordance tests. Data types. Measures of similarity and dissimilarity;
7. Principal components analysis: concept definition, mathematical model, principal components computation;
8. Principal components analysis: evaluation and visualization of the results. Model application (additional data sets), model generalizations;
9. Factor analysis: concept definition, model hypotheses, factorability. Factor extraction methods, estimation of the number of factors - Bartlett test, rotation of factors;
10. Analysis of canonical correlations;
11. Discriminant analysis: variability indicators, model significance;
12. Linear discriminant analysis. Bayesian discrimination;
13. Cluster analysis. Hierarchical algorithms;
14. Factorial analysis of correspondences.

Bibliography:

1. Claudiu Vinţe, Titus Felix Furtună, Python pentru analiza datelor, ASE, București, 2020, România;
2. Wes McKinney, Python for Data Analysis, O'Reilly Media, 2018, Statele Unite ale Americii;
3. Gheorghe Ruxanda, Analiza datelor, ASE, București, 2001, România;
4. Ruxanda Gh., Analiza factorială – tehnica de investigare multidimensională, Studii şi Cercetări de Calcul Economic şi Cibernetică Economică, nr. 3/2000, Anul XXXIII, Academia de Studii Economice, Bucuresti, 2000, România;

5. Benzecri J. P., L'analyse des données, Dunod, Paris, 1979, Franţa;
6. Harman, H.H., Modern Factor Analysis, University of Chicago Press, Chicago, Ill., 1967, Statele Unite ale Americii;
7. Cherkassky V., Mulier F., Learning from Data: Concepts, Theory and Methods, John Wiley & Sons, Inc., New York, 1998, Statele Unite ale Americii.
8. Murtagh, F., Heck, A., Multivariate data analysis, Dordrecht, 1987, Olanda;

Multiparadigm Programming - Java

1. GUI Intro (FX) and event handling/event programming paradigm. Java FX 8 (MVP - Model View Presenter) design pattern architecture for RIA - Rich Interface Applications;
2. Java and JVM, JDK, JRE Overview, command line and IDE Eclipse/Netbeans/IntelliJ IDEA compiling and running - including build automation tools: ANT/Maven/Gradle. Java SE Synthax intro (if, switch, for, while), methods, Arrays, OOP intro and C++ analogy / Ubuntu Linux or MS Windows, the Java byte-code class running without any modification/re-compiling. Multi-paradigm programming intro;
3. Java Arrays and OOP Intro - class, object, abstract class, interface. Java SE Deployment alternatives from command line classes and libraries/JARs to ANT/Maven/Gradle build automation tools and Continous Integration Engineering in Jenkins;
4. Java OOP - Class, Objects + Immutable objects, Interface, Abstract Class, Inheritance, Polymorphism - late binding, "has a" versus "is a" relationship, Interface as Type (declarative versus real type) analogy with C++ "pure" polymorphism, ClassCastException + Robocode Assignments;
5. Java Generics Programming and Datastructures/JCF - Java Collection Framework in analogy with C++ STL (Standard Template Library) in terms of: containers, iterators, and algorithms. JCF - Arrays, Lists, Hashtable, etc.;
6. Java I/O - Input / Output at byte and char level in stream oriented approach, File and RandomAccessFile classes. Java Native Interface - JNI (for understanding native methods) and Annotation plus Reflection (for understanding FTP server/automation of tests and XML parsing) will be a plus for understanding future topics. Also minimal JUnit 4 Intro as examplification of the annotations and reflections topics;
7. Java I/O serialization, annotation + reflection, and JNI. Also as new topics the exceptions mechanism and two Source Code Design Patterns - Factory Methods and Singleton;
8. Java 8 Features and Functional Programming Paradigm: Nashorn JavaScript Engine, Call-back and Inner Classes, Method references, default method, lambda expressions + functional interfaces, processing streams, Optional, Date/Time, and new API (Base64);
9. Multi-threading vs. multi-process, Multi-threading models, features of the concurrent and parallel programming paradigms, atomic operations, JVM and OS threads; Java API for multi-threading, Java Multi-threading issues – Singleton vs. Immutable Objects - Software Design Patterns impacted by Multithreading; Parallelism/HPC - High Performance Computing - Q&A - sample for concurrent access and file parallel copying;

10. Advanced Java Multi-threading (java.util.concurrent - ExecutorService + Future-Callable + Lock/Semaphore + Producer/Consumer) and Java 8 Lamdba expressions for multi-threading mechanisms;
11. Java Networking Intro: TCP vs. UDP over IP with Java Socket programming. TCP case study with multi-threading for implementing FTP and HTTP protocol;
12. Java NIO (New Input/Output), RegEx (Regular Expressions) and JDK 9/11 Modules + Java 9/11 New Features (e.g. Reactive Streams, HTTP2 Client, modified try-catch statements, etc.);
13. Java XML (JAXB2) and JSON Parsing (JSON.org / Jackson);
14. JDBC and NoSQL Database Programming.

Bibliography:

1. Jonathan Knudsen, Patrick Niemeyer, Learning in Java, O'Reilly, 2005, Statele Unite ale Americii;
2. Bruce Eckel, Thinking in Java, Prentice Hall, 2003, Statele Unite ale Americii;
3. Joshua Bloch, Effective Java, 3rd Edition, Pearson Education / Addison-Wesley Professional, 2018, Statele Unite ale Americii;