



## Course Syllabus

<b>Course Code</b>	<b>Course Title</b>	<b>ECTS Credits</b>
COMP-340	Big Data	6
<b>Prerequisites</b>	<b>Department</b>	<b>Semester</b>
COMP-302	Computer Science	Spring
<b>Type of Course</b>	<b>Field</b>	<b>Language of Instruction</b>
Required	Data Science	English
<b>Level of Course</b>	<b>Lecturer(s)</b>	<b>Year of Study</b>
1 <sup>st</sup> Cycle	Dr. D. Trihinas	3 <sup>rd</sup>
<b>Mode of Delivery</b>	<b>Work Placement</b>	<b>Corequisites</b>
Face-to-Face	N/A	None

### Course Objectives:

The main objectives of the course are to:

- Provide a comprehensive overview of the data evolution landscape and why traditional data solutions are inadequate for the specific requirements of modern scalable, reliable and fault-tolerant applications.
- Introduce the principles, concepts and modelling abstractions for Big Data and data-intensive computing at scale.
- Present the fundamental principles for popular Big Data programming models (e.g., MapReduce, Dataflow, etc.) and discuss how performance and robustness are significantly improved compared to traditional models.
- Introduce advanced programming design patterns used extensively for processing Big Data (e.g., counting, sorting, relational algebra, matrix multiplication, etc.).
- Describe alternative programming models and algorithms for data that must be processed online and also features low-latency requirements (e.g., streaming data).
- Demonstrate basic techniques towards architecting big-data solutions.
- Introduce various popular and open-source tools for big-data storage, processing and analytic insight extraction.

## Learning Outcomes:

After completion of the course students are expected to be able to:

- Describe the multiple dimensions and challenges involved in storing, processing and modelling Big Data.
- Comprehend the contexts in which Big Data principles and models are applied, while also recognizing potential implications and trade-offs depending on the context.
- Conceptually understand the capabilities and pitfalls of Big Data storage models towards relational storage models when applied on structured and unstructured data.
- Evaluate data analysis problems to determine whether and how Big Data algorithms, programming models and techniques can be applied.
- Understand the underlying principles and concepts of key Big Data programming models (e.g., MapReduce, Dataflow, etc.) and present the ability to design applications adopting these principles.
- Acknowledge how to model, adapt and extend data analysis techniques to process streaming data with low-latency requirements.
- Realize how different tools fit in the frame of Big Data analytics stacks.
- Demonstrate the ability to use open-source technologies to design basic components of Big Data solutions for data storage, processing and analytics extraction.

## Course Content:

1. Introduction to Big Data
  - a. The Data Evolution and Technology Landscape
  - b. The Big Data Dimensions
  - c. Future Predictions and Business Opportunities
2. Big Data Means and Scalability
  - a. Parallel and Distributed Computing
  - b. Data-Intensive Computing
3. Big Data Models
  - a. Data Replication
  - b. Data Partitioning/Schema Sharding
  - c. Rebalancing Data
  - d. Message Communication Models
4. Big Data Servicing
  - a. Service Discovery
  - b. Query Routing – Data Lookups
  - c. Distributed Indexing
  - d. Conflict Resolution and Consistency
5. Distributed Databases
  - a. Relational vs Non-Relational Data

- b. NoSQL Databases (e.g., column stores, document stores, graph databases)
  - c. NewSQL Models and Databases
- 6. The MapReduce Programming Model
  - a. Data Parallel Problems
  - b. MapReduce Design Patterns
  - c. MapReduce as an Execution Framework (e.g, Hadoop)
  - d. Local Aggregation and Latency Improvements
  - e. Abstractions for Relational “Big” Data
- 7. The Dataflow Programming Model
  - a. MapReduce Limitations
  - b. The Dataflow Programming Model
  - c. Dataflow Algorithmic Design Patterns
  - d. Resilient Distributed Datasets
- 8. Data Streams
  - a. The Data Stream Model
  - b. Stream vs Batch Data
  - c. Scalable Streaming Algorithms
  - d. Stream Processing Frameworks (e.g., Spark, Flink)

**Learning Activities and Teaching Methods:**

Lectures, In Class Exercises, Lab Sessions, Case-Study Presentations, Discussions.

**Assessment Methods:**

Final Exam, Midterm Exam, Semester Project, Weekly Homework..

**Required Textbooks / Readings:**

Title	Author(s)	Publisher	Year	ISBN
Big Data	Nathan Marz and James Warren	Manning	2015	978-1-617-29034-3
Data-Intensive Text Processing with MapReduce*	Jimmy Lin and Chris Dyer	Morgan and Claypool	2010	978-1-608-45342-9

Mining Massive Datasets (2 <sup>nd</sup> edition) **	Jure Leskovec, Anand Rajaraman and Jeff Ullman	Cambridge University Press	2014	978-1-107-07723-2
--	--	----------------------------	------	-------------------

\* Made freely available online by the authors: <https://lntool.github.io/MapReduceAlgorithms/MapReduce-book-final.pdf>

\*\* Made freely available online by the authors: <http://www.mmms.org/#book>

### Recommended Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Designing Data-Intensive Applications	Martin Kleppmann	O'Reilly	2017	978-1-449-37332-0
Hadoop: The Definitive Guide (4 <sup>th</sup> edition)	Tom White	O'Reilly	2015	978-1-491-90168-7
Big Data Fundamentals	Thomas Erl and Wajid Khattak and Paul Buhler	Prentice Hall	2016	978-0-134-29107-9
Spark: The Definitive Guide	Matei Zaharia and Bill Chambers	O'Reilly	2018	978-1-49191-221-8
Big Data: Principles and Paradigms	Rajkumar Buyya and Rodrigo N. Calheiros and Amir Vahid Dastjerdi	Morgan Kaufmann	2016	978-0-128-05394-2