# COURSE DESCRIPTION CARD - SYLLABUS

Course name
Information Theory

## Course

Field of study
Artificial Intelligence

Year/Semester
2/3

Area of study (specialization)

Profile of study
general academic

Level of study
First-cycle studies

Course offered in
English

Form of study
full-time

Requirements
elective

## Number of hours

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

## Number of credit points

3

## Lecturers

Responsible for the course/lecturer:
Robert Susmaga
e-mail: robert.susmaga@cs.put.poznan.pl
tel.: 616652934
Faculty of Computing and Telecommunications
Piotrowo 2, 60-965 Poznań

Responsible for the course/lecturer:

## Prerequisites

Basic knowledge regarding:
a) calculus (logarithmic function, exponential function, function derivative),
b) linear algebra (vectors, matrices, vector/matrix operations),
c) probability theory and statistics (probability, including conditional probability, random variables, probability distributions and probability distribution functions, mean values, expected values, variance),
d) data structures (one- and two-dimensional arrays, lists, trees).

Basic skills regarding designing, creating and testing computer programs (in a programming language of one's choice) that implement simple processing of static (vectors and matrices) and dynamic (lists, trees) data structures.

(recommended) A fair amount of cognitive curiosity and not less perseverance in pursuing the goals of personal development.

## Course objective

The objective of the course is to present a selection of aspects of the Information Theory, one of the most fundamental theories underlying theoretical Computer Science of modern-day. The Information Theory deals with representing, storing and communicating information expressed in the form of symbols. Owing to the fact that many important applications of this theory reach far beyond the core of Computing Science, the presented selection of aspects will be confined to the most fundamental ones, mainly those related to such domains of the Computer Science as Data Exploration and Data Compression, in particular: Lossless Data Compression. The fruits of rapid development of notions in the Information Theory, initially disputable and professedly unsolvable, have soon turned out to be incredibly useful and to have a great deal of practical value. In the modern-day these solutions show up in virtually all imaginable computer systems in existence, ones that could hardly survive nowadays without the ubiquitous multimedia content, the popularity and versatility of which has been consistently and unwaveringly influenced by the accessibility of data compression methods.
Detailed objectives of the course include sharing skills and knowledge sufficient for:
a) understanding the fundamental ideas underlying the Information Theory, especially those pertaining to Data Exploration and Data Compression,
b) identifying, formulating and solving basic problems in the Information Theory, Data Exploration and Data Compression,
c) designing and creating computer programs that successfully implement the presented methods and algorithms.

## Course-related learning outcomes

### Knowledge

The students:
1. have a basic, ordered and well-grounded knowledge essential for important areas of computer science such as algorithmics and programming languages -- [K1st_W2]
2. know and understand the basic techniques, methods, algorithms, and tools used for solving computer problems related to Information Theory and its applications to Data Exploration / Compression -- [K1st_W4]
3. know and understand the basic techniques, methods, algorithms, and tools used for solving computer problems as well as problems in artificial intelligence, including an automated recognition of patterns in data of different types and their processing -- [K1st_W5]
4. have a basic knowledge of key directions and the most important successes of artificial intelligence understood as an essential sub-domain of computer science, making use of the achievements of other scientific disciplines, like Information Theory to provide solutions with a high practical impact -- [K1st_W5]

### Skills

The students:
1. can collect information from the appropriate sources of different nature, perform its critical analysis, interpretation and synthesis as well as comprehensively draw and justify formulated conclusions regarding the information, especially in the context of Information Theory and its applications to Data Exploration / Compression -- [K1st_U1]

2. can efficiently plan and carry out experiments, including computer measurements and simulations related to various aspects of Information Theory, interpret the obtained results and draw conclusions based on the experimental outcomes -- [K1st_U4]
3. can retrieve, analyse and transform different types of data and carry out data synthesis to knowledge and conclusions useful for solving a variety of problems that arise in computer science, especially in AI -- [K1st_U10]
4. can -- following a pre-defined specification -- design and create an IT system by first selecting and then using available methods, techniques and computer tools (including programming languages) -- [K1st_U8]
5. can adapt existing algorithms as well as formulate and implement novel ones, including algorithms typical for different streams of AI, using at least one well-known tool -- [K1st_U9]

Social competences

The students:
1. understand that knowledge and skills quickly become outdated and perceives the need for constant additional training and raising one's qualifications -- [K1st_K1]
2 are aware of the importance of scientific knowledge and research related to computer science and AI in solving problems that are essential for the functioning of individuals, firms, organizations as well as the entire society -- [K1st_K2]

## Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Formative assessment (laboratory classes): evaluation of the solutions to the assigned programming problems (as they arise).
Final assessment:
-- (laboratory clasess): evaluation of the solutions to the assigned programming problems (final),
-- (lectures): evaluation of the results of a written test (45--60 min) with both multiple choice, short answer and (small) computational questions (mostly: micro-problems to be solved in writing).

## Programme content

The course includes, but is not limited to, the following.
Fundamentals of:
-- calculus (fundamental properties and graphs of exp(x), log(x) and x*log(x)),
-- linear algebra (linear / convex combinations of vectors, convex hulls, distance measures, metric spaces),
-- probability theory (discrete random variables): probability, conditional probability, probability distribution functions,
-- statistics (continuous random variables): variance, covariance and correlation.
The idea of Shannon information (shortly: information). The measure of information content: construction and properties. The Hartley measure of information content.
The idea of the Shannon entropy (shortly: entropy) and its basic mathematical properties, recognition of

dimensions in multi-dimensional entropy, graphs and extrema of multi-dimensional entropy, entropy as the measure of information content: construction and properties, in particular: the subdivision property.
Entropy variants and entropy-related measures in multi-dimensional Data Exploration problems: the idea, properties and interpretations of: joint entropy; conditional entropy, mutual information, cross-entropy, Kullback-Leibler divergence. Alternative definitions of entropy.
Entropy in Data Compression problems: compression limits and Shannon's source coding theorem. The universal compressing method.
Basic applications of entropy in data transfer.
Three types of data processing: encoding/decoding, encrypting/decrypting and hashing (with focus on the first).
Encoding/decoding (discrete data): the idea and objectives (including: data transfer reliability). Codes: definitions and examples, tree representations of codes, prefix codes. Kraft's inequality. Error-correcting codes and Hamming distance. Data encoding aimed at compressing.
Compression (discrete and continuous data, with focus on the former): idea and objectives (including: data transfer rate). Two basic types of data compression (lossless and lossy, with focus on the former).
Lossless compression (discrete data). Lossless compression measures. Examples of lossless compression: coding-based compression. Shannon-Fano encoding and Huffman encoding: the idea, code trees, optimal codes, algorithms, examples, properties, adaptive approaches, applications.
Lossless compression (discrete data): dictionary-based compression. Lempel-Ziv-Welch method: the idea, representations, dictionaries, algorithms, examples, properties, adaptive approaches.
Notes on lossy compression (continuous data). Lossy compression measures. Examples of methods and approaches: discretization and quantization, data transforms.

## Teaching methods

Lectures: slide show presentations (theoretical elements, explanations, examples, exercises).
Laboratory classes: designing and creating (in a programming language of one's choice) programs that solve the assigned problems (which illustrate the ideas and notions presented during the lectures).

## Bibliography

Basic

1. D.J.C. MacKay: Information Theory, Inference, and Learning Algorithms, Cambridge University Press, Cambridge, UK, 2003.
2. T.M. Cover, J.A. Thomas, Elements of Information Theory, 2nd Edition, Wiley and Sons, Hoboken, New Jersey, 1991.
3. K. Sayood (red.): Lossless Compression Handbook, Academic Press, Elsevier Science, San Diego, California, 2003.

Additional

1. Lecture notes (slide show presentations)
2. K. Sayood: Introduction to Data Compression, 3rd Ed., Morgan Kaufmann Publishers, San Francisco, California, 2006.

3. A. Drozdek: Wprowadzenie do kompresji danych WNT, Warszawa, 1999 (in Polish).
4. A. Przelaskowski: Kompresja danych. Podstawy, metody bezstratne, kodery obrazów, BTC, Legionowo, 2005 (in Polish)

## Breakdown of average student's workload

| | Hours | ECTS |
|---|---|---|
| Total workload | 70 | 3,0 |
| Classes requiring direct contact with the teacher | 30 | 1,3 |
| Student's own work (revising the lectures, solving exercises, programming for laboratory classes, preparing for the final test) [1] | 40 | 1,7 |

[1] delete or add other activities as appropriate