Professional Master's Degree

Artificial Intelligence and Knowledge Engineering

tech technological university

tech technological university

Professional Master's Degree

Artificial Intelligence and Knowledge Engineering

- Modality: **online**
- Duration: **12 months**
- Certificate: **TECH Technological University**
- Dedication: **16h/week**
- Schedule: **at your own pace**
- Exams: **online**

Website: www.techtitute.com/pk/information-technology/professional-master-degree/master-artificial-intelligence-knowledge-engineering

Index

01

Introduction

This program is designed for professionals in the field of Engineering to immerse themselves in the exciting world of Artificial Intelligence and Knowledge Engineering. Through a highly competent program you will be able to take a solid and solvent step in this field, achieving the personal and professional skills necessary to practice as an expert in the field. A complete and effective program that will propel you to the highest level of competence.

Become one of the most in-demand professionals of today. Specialize with thi complete Professional Master's Degree in Artificial Intelligence and Knowledge Engineering"

Developments based on Artificial Intelligence have reached numerous applications within the field of engineering. From the automation of numerous procedures in Industry and companies, to process control itself. This means that engineering professionals need to know and master the operation of these complex techniques.

This essential knowledge also becomes the first step to gain access to the development capacity of this type of technology.

Throughout this program, a real working scenario is offered in order to be able to evaluate the convenience of its application in your own project, assessing its real indications, its way of development and the expectations you may have regarding the results.

Through experience you will learn how to develop the necessary knowledge to advance in this field of work. This program, which necessarily requires experience, is reconciled through distance learning and practical teaching, offering a unique option to give your CV the boost you are looking for...

This **Professional Master's Degree in Artificial Intelligence and Knowledge Engineering,** contains the most complete and up-to-date program on the market. Its most notable features are:

- The latest technology in online teaching software
- A highly visual teaching system, supported by graphic and schematic contents that are easy to assimilate and understand
- Practical cases presented by practising experts
- State-of-the-art interactive video systems
- Teaching supported by telepractice
- Continuous updating and recycling systems
- Autonomous learning: full compatibility with other occupations
- Practical exercises for self-evaluation and learning verification
- Support groups and educational synergies: questions to the expert, debate and knowledge forums
- Communication with the teacher and individual reflection work
- Availability of content from any fixed or portable device with internet connection
- Supplementary documentation databases are permanently available, even after the program

Join the elite with this highly effective instructional program and open new avenues for your professional advancement"

A Professional Master's Degree that will enable you to work in all areas of Artificial Intelligence and Knowledge Engineering with the solvency of a high-level professional"

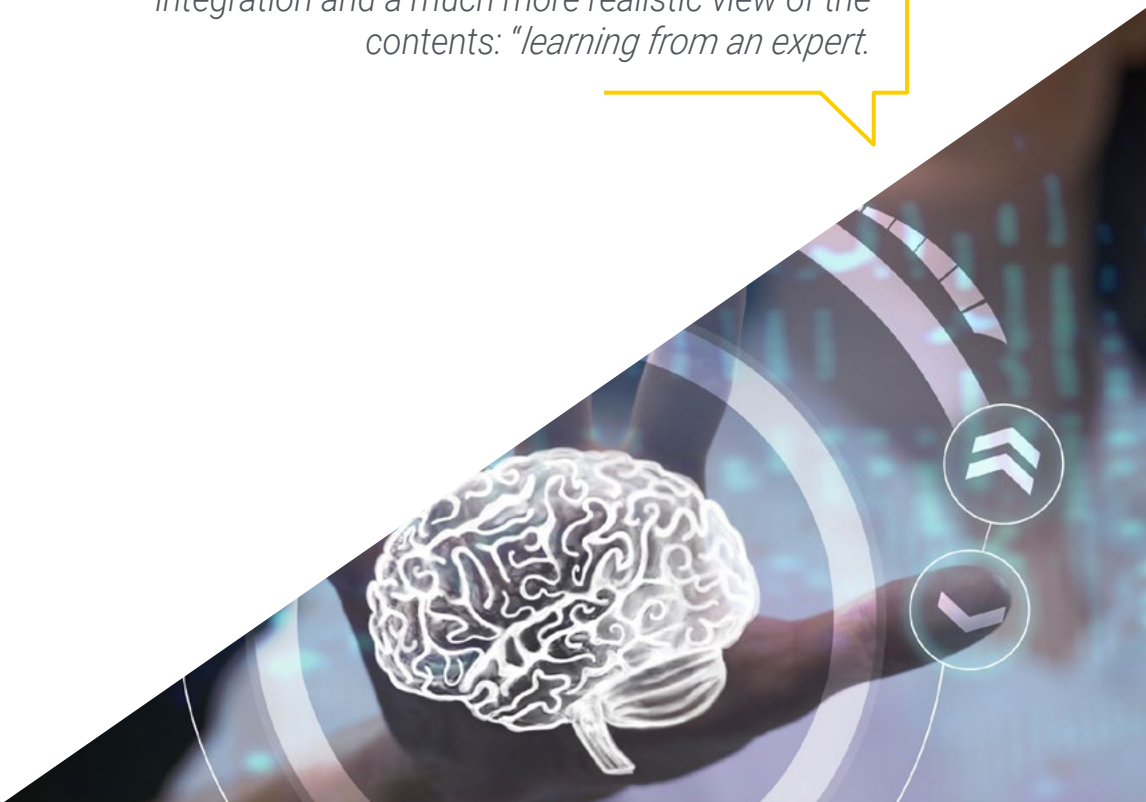
Our teaching staff is made up of professionals from different fields related to this specialty. In this way, the intended objective of instructional updating is achieved. A multidisciplinary faculty of trained and experienced professionals in different environments, who will develop the theoretical knowledge efficiently, but above all, will put at your service the practical knowledge derived from their own experience: one of the differential qualities of this program.

The efficiency of the methodological design of this Professional Master's Degree, enhances the student's understanding of the subject. Developed by a multidisciplinary team of e-learning experts, the Method integrates the latest advances in educational technology. In this way, you will be able to study with a range of comfortable and versatile multimedia tools that will give you the operability you need in your training.

The design of this program is based on Problem-Based Learning, an approach that views learning as a highly practical process. To achieve this remotely TECH will use telepractice. with the help of an innovative interactive video system and Learning from an Expert you will be able to acquire the knowledge as if you were facing the scenario about which you are currently learning. A concept that will make it possible to integrate and fix learning in a more realistic and permanent way.

With a methodological design based on teaching techniques proven for their effectiveness, this innovative Professional Master's Degree in Artificial Intelligence and Knowledge Engineering will take you through different teaching approaches to allow you to learn in a dynamic and effective way.

Our innovative telepractice concept will give you the opportunity to learn through an immersive experience, which will provide you with a faster integration and a much more realistic view of the contents: "learning from an expert.



02

Objectives

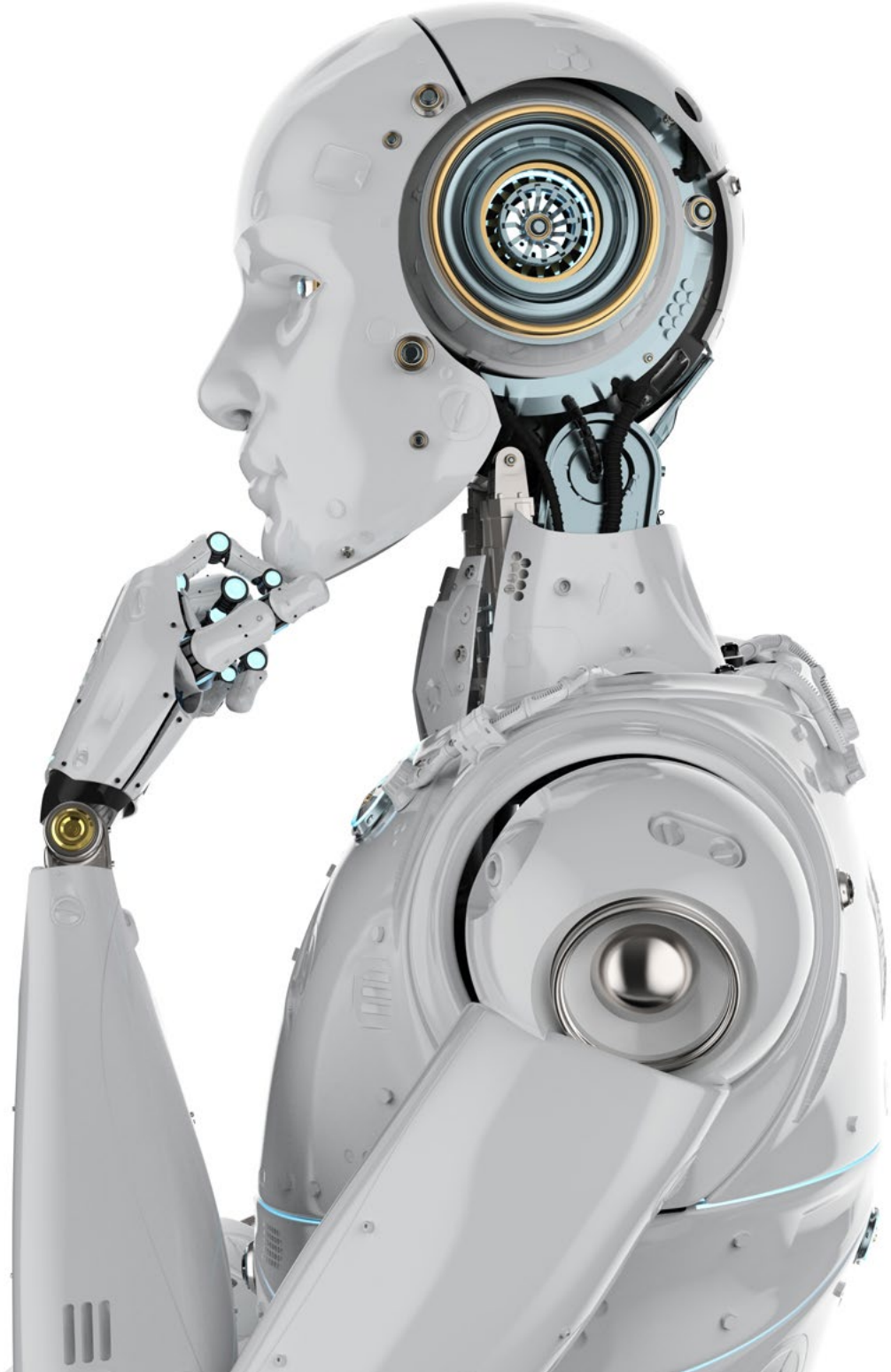
The objective is to specialize highly qualified professionals for work experience. An objective that is complemented in a global manner by promoting human development that lays the foundations for a better society. This objective is achieved by helping professionals to access a much higher level of competence and control. A goal that, in just a few months, can be achieved with high intensity and precision program.

If your objective is to broaden your skills set to include new paths of success and development: this is the program for you: a training that aspires to excellence"

General Objectives

- Train scientifically and technologically for the practice of computer engineering
- Obtain comprehensive knowledge in the field of computer science
- Obtain broad knowledge in the field of computer structure
- Acquire the necessary knowledge in software engineering
- Review the mathematical, statistical and physical bases essential for this subject

Don't miss the opportunity and get up to date on new developments in General Health Psychology to incorporate them into your daily medical practice"



Specific Objectives

Module 1. Programming Fundamentals

- Understand the basic structure of a computer, software and general purpose programming languages
- Learn how to design and interpret algorithms, which are the necessary basis for developing computer programs
- Understand the essential elements of a computer program, such as the different data types, operators, expressions, statements, I/O and control statements
- Understand the different data structures available in general purpose programming languages, both static and dynamic, and to acquire the essential knowledge for file handling
- Learn the different testing techniques in computer programs and the importance of generating good documentation together with good source code
- Learn the basic concepts of the C++ programming language, one of the most widely used languages in the world

Module 2. Data Structure

- Learn the fundamentals of programming in the C++ language, including classes, variables, conditional expressions and objects
- Understand abstract data types, linear data structure types, simple and complex hierarchical data structures and their implementation in C++
- Understand the operation of advanced data structures other than the usual ones
- Know the theory and practice related to the use of priority heaps and priority queues
- Learn the operation of hash tables, such as abstract data types and functions
- Understand graph theory, as well as advanced graph algorithms and concepts

Module 3. Algorithm and Complexity

- Learn the main algorithm design strategies, as well as the different methods and measures for algorithm computation
- Know the main sorting algorithms used in software development
- Understand the operation of the different algorithms with trees, heaps and graphs
- Understand how greedy algorithms work, their strategy and examples of their use in the main known problems We will also learn the use of greedy algorithms on graphs
- Learn the main strategies of minimum path search, with the approach of essential problems of the field and algorithms for their resolution
- Understand the backtracking technique and its main uses, as well as other alternative techniques

Module 4. Advanced Algorithms Design

- Learn more about the advanced design of algorithms, analyzing recursive and divide and conquer algorithms, as well as performing amortized analysis
- Understand the concepts of dynamic programming and algorithms for NP problems
- Understand the operation of combinatorial optimization, as well as the different randomization algorithms and parallel algorithms
- Know and understand the operation of different local and candidate search methods
- Learn the mechanisms of formal program verification and iterative program verification, including first-order logic and Hoare's formal system
- Learn the operation of some of the main numerical methods such as the bisection method, the Newton Raphson method and the secant method

Module 5. Computational Logic

- Learn the fundamentals of computational logic, what it is used for and its justification of use
- Know the different strategies of formalization and deduction in propositional logic, including natural reasoning, axiomatic and natural deduction, as well as the primitive rules of propositional calculus
- Acquire advanced knowledge in propositional logic, going into the semantics of this logic and the main applications of this logic such as logical circuits
- Understand predicate logic both for the calculation of natural deduction of predicates and for the formalization and deduction strategies for predicate logic
- Understand the basics of natural language and its deductive mechanism
- Introduce logic programming using the PROLOG language

Module 6. Artificial Intelligence and Knowledge Engineering

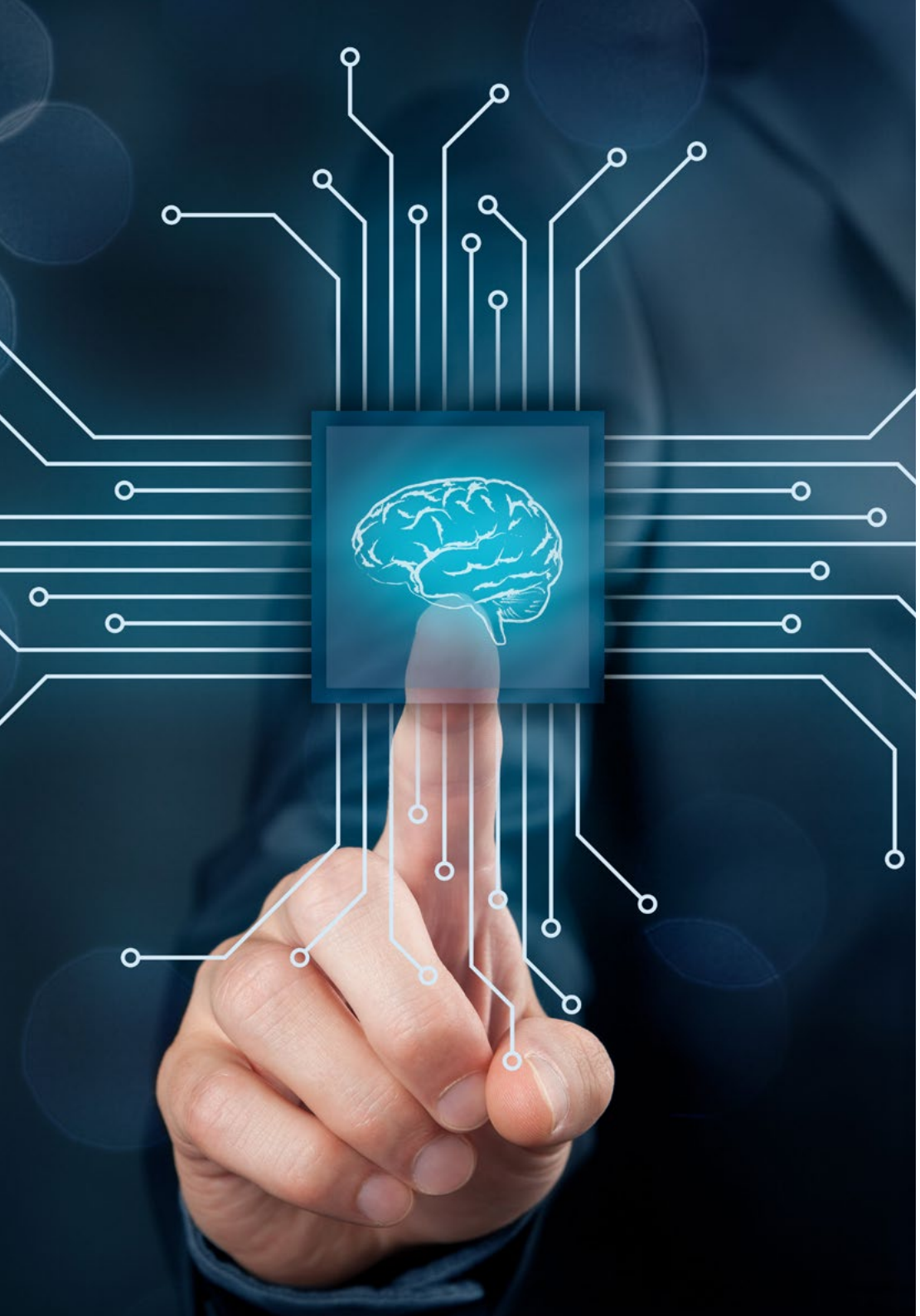
- Lay the foundations of Artificial Intelligence and Knowledge Engineering, making a brief tour through the history of Artificial Intelligence up to the present day
- Understand the essential concepts of search in Artificial Intelligence, both informed and uninformed search
- Understand how Artificial Intelligence works in games
- Learn the fundamental concepts of neural networks and the use of genetic algorithms
- Acquire the appropriate mechanisms to represent knowledge, especially taking into account the semantic web
- Understand the functioning of expert systems and decision support systems

Module 7. Intelligent Systems

- Learn all the concepts related to agent theory and agent architecture and its reasoning process
- Assimilate the theory and practice behind the concepts of information and knowledge, as well as the different ways of representing knowledge
- Understand the theory related to ontologies, as well as learn ontology languages and software for ontology creation
- Learn different models of knowledge representation, such as vocabularies, taxonomies, thesauri and mind maps, among others
- Understand the functioning of semantic reasoners, knowledge-based systems and expert systems
- Know how the semantic web works, its current and future state, as well as semantic web-based applications

Module 8. Machine Learning and Data Mining

- Introduce knowledge discovery processes and basic concepts of machine learning
- Learn data exploration and pre-processing methods, as well as different algorithms based on decision trees
- Understand the operation of Bayesian methods and regression and continuous response methods
- Understand the different classification rules and the evaluation of classifiers by learning how to use confusion matrices and numerical evaluation, the Kappa statistic and the ROC curve
- Acquire essential knowledge related to text mining and natural language processing (NLP) and clustering
- Expand your knowledge of neural networks, from simple neural networks to recursive neural networks



Module 9. Multiagent Systems and Computational Perception

- Understand the basic and advanced concepts related to agents and multi-agent systems
- Study the FIPA agent standard, considering agent communication, agent management and architecture, among other issues
- Deepen the learning of the JADE (Java Agent Development Framework) platform by learning to program in it both basic and advanced concepts, including topics of communication and agent discovery
- Lay the foundations of natural language processing, such as automatic speech recognition and computational linguistics
- Gain an in-depth understanding of computer vision, digital image analysis, transformation and segmentation of digital images

Module 10. Bio-Inspired Computing

- Introduce the concept of bio-inspired computing, as well as to understand the functioning of the different types of social adaptation algorithms and genetic algorithms
- Deepen the study of the different models of evolutionary computation, knowing their strategies, programming, algorithms and models based on estimation of distributions
- Understand the main space exploration-exploitation strategies for genetic algorithms
- Understand the operation of evolutionary programming applied to learning problems and multi-objective problems
- Learn the essential concepts related to neural networks and understand the operation of real use cases applied to areas as diverse as medical research, economics and computer vision

03

Skills

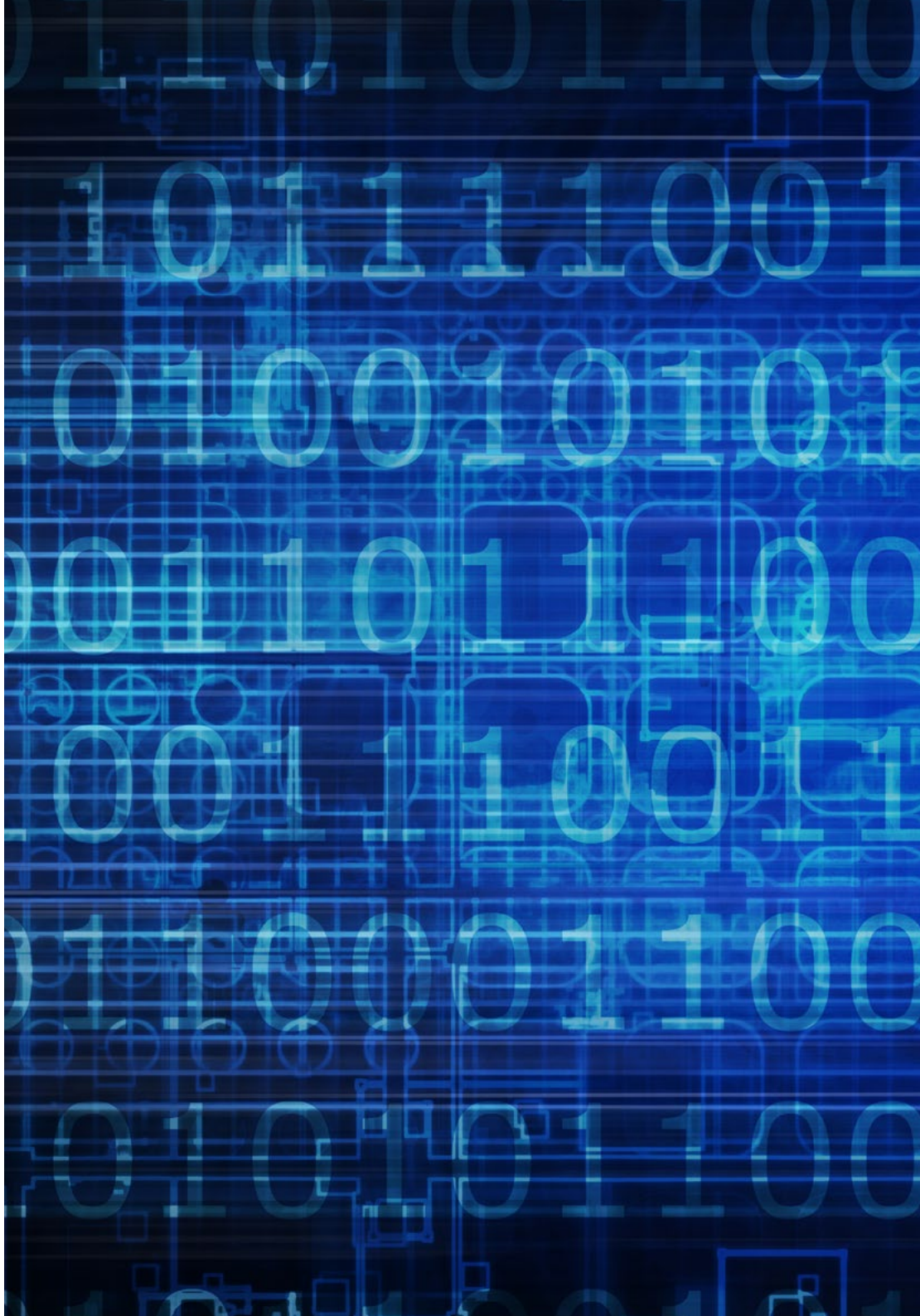
This Professional Master's Degree in Artificial Intelligence and Knowledge Engineering has been created as an educational tool for the professional. Its intensive specialization will prepare you to work in all areas related to Artificial Intelligence with the confidence of an expert in the field.

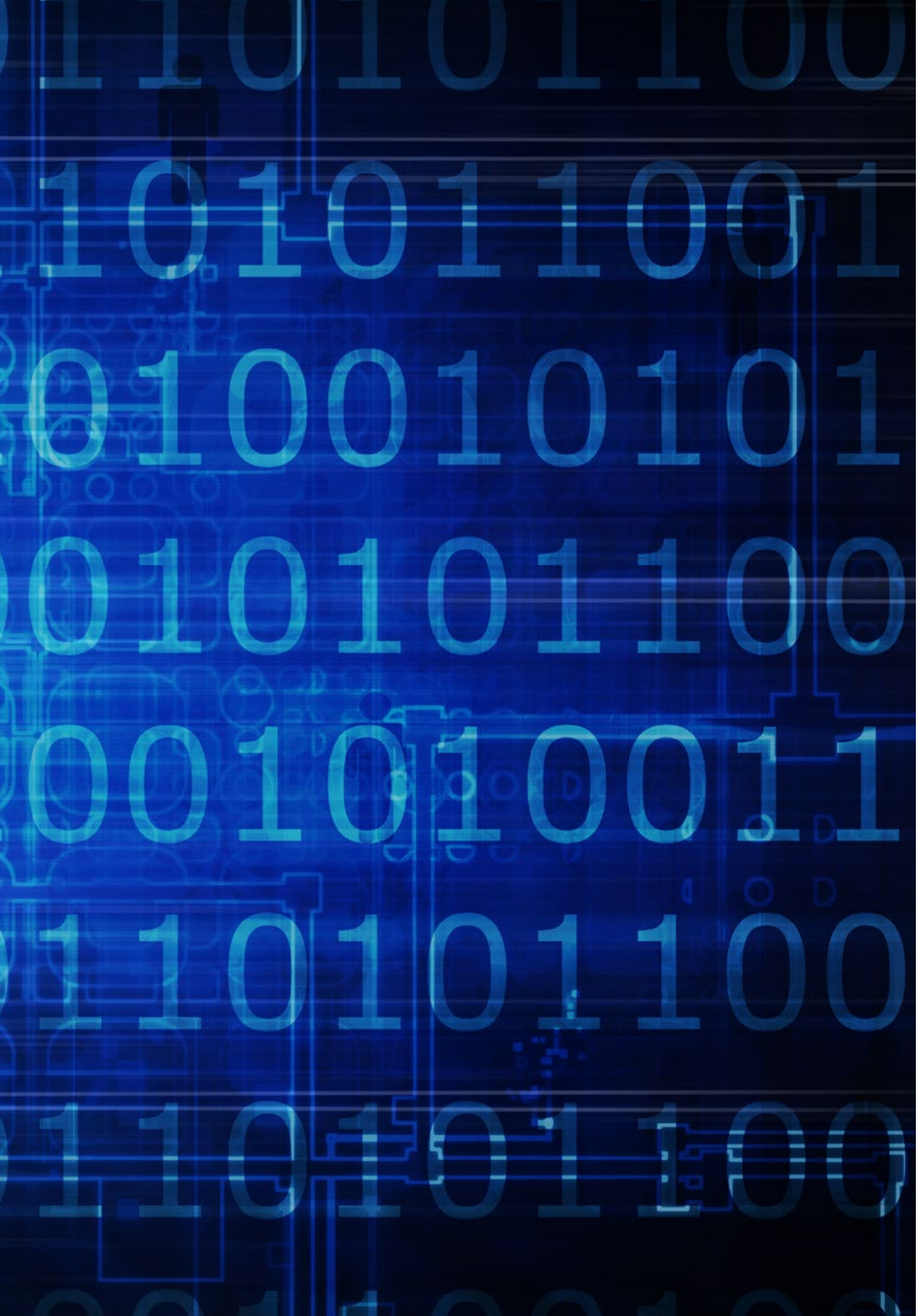
The Professional Master's Degree Artificial Intelligence and Knowledge Engineering will provide you with the essential personal and professional skills to play an adequate role in any professional situation in this field of intervention"

General Skill

- Acquire the necessary skills for the professional practice of computer engineering with the knowledge of all the necessary factors to perform it with quality and solvency

A unique, key, and decisive program to boost your professional development"





Specific Skills

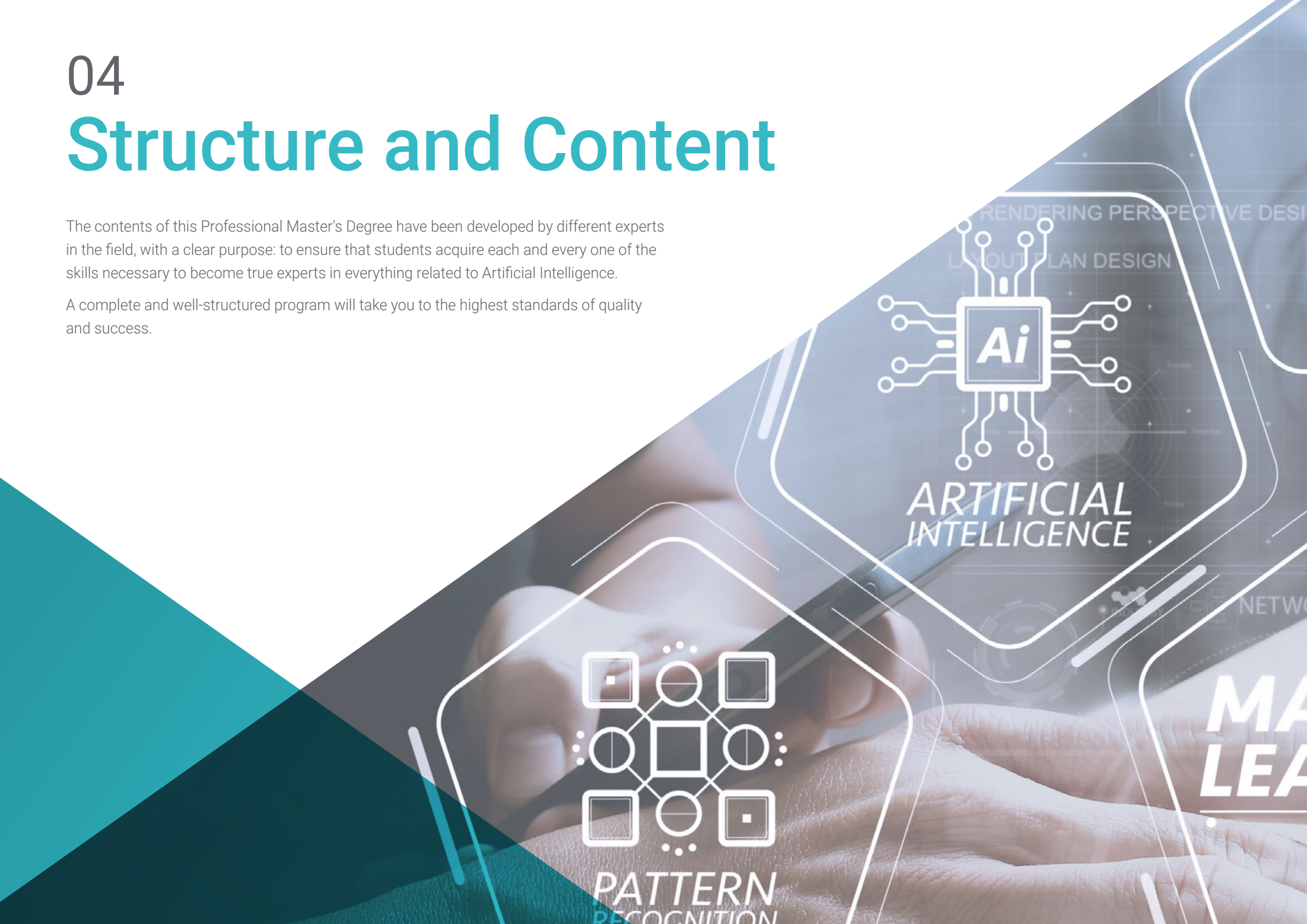
- Develop programming in the area of artificial intelligence taking into account all the factors of its development
- Know with solvency the data structure in C++ programming
- Design basic and advanced algorithms
- Understand computational logic and apply it in the design of projects
- Know about Artificial Intelligence, its uses and its developments and how to implement in your own projects
- Know what they are, how they work and how to work with intelligent systems
- Master the basic concepts of machine learning
- Knowledge of JADE, FIPA, computer vision and other multi-agent systems
- Knowledge of bio-inspired computing algorithms and utilization strategies

04

Structure and Content

The contents of this Professional Master's Degree have been developed by different experts in the field, with a clear purpose: to ensure that students acquire each and every one of the skills necessary to become true experts in everything related to Artificial Intelligence.

A complete and well-structured program will take you to the highest standards of quality and success.



A comprehensive teaching program, structured in well-developed teaching units, oriented towards learning that is compatible with your personal and professional life"

Module 1. Programming Fundamentals

1.1. Introduction to Programming
- 1.1.1 Basic Computer Structure
- 1.1.2 Software
- 1.1.3 Programming Languages
- 1.1.4 Computer Application Life Cycle

1.2. Algorithm Design
- 1.2.1 Problem Solving
- 1.2.2 Descriptive Techniques
- 1.2.3 Algorithm Elements and Structure

1.3. Program Elements
- 1.3.1 C++ Origin and Features
- 1.3.2 Development Environment
- 1.3.3 Concept of Program
- 1.3.4 Types of Fundamental Data
- 1.3.5 Operators
- 1.3.6 Expressions
- 1.3.7 Statements
- 1.3.8 Data Input and Output

1.4. Control Sentences
- 1.4.1 Statements
- 1.4.2 Branches
- 1.4.3 Loops

1.5. Abstraction and Modularity: Function
- 1.5.1 Modular Design
- 1.5.2 Concept of Function and Utility
- 1.5.3 Definition of Function
- 1.5.4 Execution Flow in a Function Call
- 1.5.5 Function Prototypes
- 1.5.6 Results Return
- 1.5.7 Calling Functions: Parameters
- 1.5.8 Parameter Passing According to Reference and Value
- 1.5.9 Scope Identifier

1.6. Statistical Data Structures
- 1.6.1 Arrays
- 1.6.2 Matrices Polyhedra
- 1.6.3 Searching and Sorting
- 1.6.4 Chaining: I/O Functions for Chains
- 1.6.5 Structures: Unions
- 1.6.6 New Types of Data

1.7. Dynamic Data Structures: Pointers
- 1.7.1 Concept. Definition of Pointer
- 1.7.2 Pointer Operators and Operations
- 1.7.3 Pointer Arrays
- 1.7.4 Pointers and Arrays
- 1.7.5 Chain Pointers
- 1.7.6 Pointers to Structures
- 1.7.7 Multiple Indirectness
- 1.7.8 Function Pointers
- 1.7.9 Passing of Functions, Structures, and Arrays as Function Parameters

1.8. Files
- 1.8.1 Basic Concepts
- 1.8.2 File Operations
- 1.8.3 Types of Files
- 1.8.4 File Organization
- 1.8.5 Introduction to C++ Files
- 1.8.6 Managing Files

1.9. Recursion
- 1.9.1 Definition of Recursion
- 1.9.2 Types of Recursion
- 1.9.3 Advantages and Disadvantages
- 1.9.4 Considerations
- 1.9.5 Recursive-Iterative Conversion
- 1.9.6 Recursion Stack

1.10. Testing and Documentation
- 1.10.1 Program Testing
- 1.10.2 White Box Testing
- 1.10.3 Black Box Testing
- 1.10.4 Testing Tools
- 1.10.5 Program Documentation

Module 2. Data Structure

2.1. Introduction to C++ Programming
- 2.1.1 Classes, Constructors, Methods and Attributes
- 2.1.2 Variables
- 2.1.3 Conditional Expressions and Loops
- 2.1.4 Objects

2.2. Abstract Data Types (ADT)
- 2.2.1 Types of Data
- 2.2.2 Basic Structures and ADT
- 2.2.3 Vectors and Arrays

2.3. Lineal Data Structures
- 2.3.1 ADT Ready Definition
- 2.3.2 Linked and Double-Linked Lists
- 2.3.3 Ordered Lists
- 2.3.4 C++ Lists
- 2.3.5 ADT Stack
- 2.3.6 ADT Queue
- 2.3.7 Stack and Queue in C++

2.4. Hierarchical Data Structures
- 2.4.1 ADT Tree
- 2.4.2 Tours
- 2.4.3 N-ary Trees
- 2.4.4 Binary Trees
- 2.4.5 Binary Search Trees

2.5. Hierarchical Data Structures: Complex Trees
- 2.5.1 Perfectly Balanced Trees or Trees of Minimum Height
- 2.5.2 Multi-Path Trees
- 2.5.3 Bibliographical References

2.6. Priority Mounds and Queue
- 2.6.1 ADT Mounds
- 2.6.2 ADT Priority Queue

2.7. Hash Tables
- 2.7.1 ADT Hash Table
- 2.7.2 Hash Functions
- 2.7.3 Hash Function in Hash Tables
- 2.7.4 Redispersion
- 2.7.5 Open Hash Tables

2.8. Graphs
- 2.8.1 ADT Graph
- 2.8.2 Types of Graph
- 2.8.3 Graphical Representation and Basic Operations
- 2.8.4 Graph Design

2.9. Algorithms and Advanced Graph Concepts
- 2.9.1 Problems about Graphs
- 2.9.2 Road Algorithms
- 2.9.3 Search Algorithms or Paths
- 2.9.4 Other Algorithms

2.10. Other Data Structure
- 2.10.1 Sets
- 2.10.2 Parallel Arrays
- 2.10.3 Symbol Tables
- 2.10.4 Tries

Module 3. Algorithm and Complexity

3.1. Introduction to Algorithm Design Strategies
- 3.1.1 Recursion
- 3.1.2 Divide and Conquer
- 3.1.3 Other Strategies

3.2. Algorithm Efficiency and Analysis
- 3.2.1 Efficiency Measures
- 3.2.2 Measuring Entry Size
- 3.2.3 Measuring Execution Time
- 3.2.4 Worst, Best and Average Case
- 3.2.5 Asymptotic Notation
- 3.2.6 Mathematical Analysis Criteria for Non-Recursive Algorithms
- 3.2.7 Mathematical Analysis for Recursive Algorithms
- 3.2.8 Empirical Analysis for Algorithms

3.3. Sorting Algorithms
- 3.3.1 Concept of Sorting
- 3.3.2 Bubble Sorting
- 3.3.3 Selection Sorting
- 3.3.4 Insertion Sorting
- 3.3.5 Merge Sort
- 3.3.6 (Quick Sort)

3.4 Tree-Based Algorithms
- 3.4.1 Concept of Tree
- 3.4.2 Binary Trees
- 3.4.3 Tree Range
- 3.4.4 Representing Expressions
- 3.4.5 Sorted Binary Trees
- 3.4.6 Balanced Binary Trees

3.5. Algorithms Using Heaps
- 3.5.1 Heaps
- 3.5.2 The Heapsort Algorithm
- 3.5.3 Priority Queues

3.6. Graph Algorithms
- 3.6.1 Representation
- 3.6.2 Width Range
- 3.6.3 In-Depth Range
- 3.6.4 Topological Sorting

3.7. Greedy Algorithms
- 3.7.1 Greedy Strategy
- 3.7.2 Greedy Strategy Elements
- 3.7.3 Currency Exchange
- 3.7.4 Traveler's Problem
- 3.7.5 The Backpack Problem

3.8. Search for Minimum Paths
- 3.8.1 Problem of the Minimum Path
- 3.8.2 Negative Arcs and Cycles
- 3.8.3 Dijkstra's Algorithm

3.9. Greedy Algorithms on Graphs
- 3.9.1 The Minimum Overlapping Tree
- 3.9.2 Prim's Algorithm
- 3.9.3 Kruskal's Algorithm
- 3.9.4 Complexity Analysis

3.10. Backtracking
- 3.10.1 Backtracking
- 3.10.2 Alternative Techniques

Module 4. Advanced Algorithms Design

4.1. Analysis of Recursive and Divide and Conquer Algorithms
- 4.1.1 Posing and Solving Homogeneous and Non-Homogeneous Recurrence Equations
- 4.1.2 Divide and Conquer Strategy Overview

4. 2. Amortized Analysis
- 4.2.1 Aggregate Analysis
- 4.2.2 The Accounting Method
- 4.2.3 The Potential Method

4.3. Dynamic Programming and Algorithms for NP Problems
- 4.3.1 Dynamic Programming Features
- 4.3.2 Backtracking
- 4.3.3 Branching and Pruning

4.4. Combinatorial Optimization
- 4.4.1 Representation of Problems
- 4.4.2 Optimization in 1D

4.5. Randomization Algorithms
- 4.5.1 Examples of Randomization Algorithms
- 4.5.2 The Buffon Theorem
- 4.5.3 Monte Carlo Algorithm
- 4.5.4 Las Vegas Algorithm

4.6. Local and Candidate Search
- 4.6.1 Garcient Ascent
- 4.6.2 Hill Climbing
- 4.6.3 Simulated Annealing
- 4.6.4 Tabu Search
- 4.6.5 Candidate Search

4.7. Formal Program Verification
- 4.7.1 Specification of Functional Abstractions
- 4.7.2 The Language of First-Order Logic
- 4.7.3 Hoare's Formal System

4.8. Iterative Program Verification
- 4.8.1 Rules of Hoare's Formal System
- 4.8.2 Concept of Invariant Iterations

4.9. Numerical Methods
- 4.9.1 The Bisection Method
- 4.9.2 The Newton Raphson Method
- 4.9.3 The Secant Method

4.10. Parallel Algorithms
- 4.10.1 Parallel Binary Operations
- 4.10.2 Parallel Operations with Graphs
- 4.10.3 Parallelism in Divide and Conquer
- 4.10.4 Parallelism in Dynamic Programming

Module 5. Computational Logic

5.1. Justification of the Logic
- 5.1.1 Object of Logic Study
- 5.1.2 What Is Logic for?
- 5.1.3 Components and Types of Reasoning
- 5.1.4 Components of a Logic Calculation
- 5.1.5 Semantics
- 5.1.6 Justification of the Existence of a Logic
- 5.1.7 How to Check that a Logic is Appropriate?

5.2. Calculation of Natural Deduction from Statements
- 5.2.1 Formal Language
- 5.2.2 Deductive Mechanism

5.3. Formalization and Deduction Strategies for Propositional Logic
- 5.3.1 Formalization Strategies
- 5.3.2 Natural Reasoning
- 5.3.3 Laws and Rules
- 5.3.4 Axiomatic Deduction and Natural Deduction
- 5.3.5 Calculation of the Natural Deduction
- 5.3.6 Primitive Rules of Propositional Calculus

5.4. Semantics of Propositional Logic
- 5.4.1 Truth Tables
- 5.4.2 Equivalence
- 5.4.3 Tautologies and Contradictions
- 5.4.4 Validation of Propositional Sentences
- 5.4.5 Validation by Means of Truth Tables
- 5.4.6 Validation Using Semantic Trees
- 5.4.7 Validation by Refutation

5.5. Applications of Propositional Logic: Logic Circuits
- 5.5.1 Basic Gates
- 5.5.2 Circuits
- 5.5.3 Mathematical Models of the Circuits
- 5.5.4 Minimization
- 5.5.5 The Second Canonical Form and the Minimum Form in Product of Additions
- 5.5.6 Other Gates

5.6. Calculation of Natural Deduction of Predicates
- 5.6.1 Formal Language
- 5.6.2 Deductive Mechanism

5.7. Formalization Strategies for Predicate Logic
- 5.7.1 Introduction to Formalization in Predicate Logic
- 5.7.2 Formalization Strategies with Quantifiers

5.8. Deduction Strategies for Predicate Logic
- 5.8.1 Reason for Omission
- 5.8.2 Presentation of the New Rules
- 5.8.3 Predicate Logic as a Natural Deduction Calculus

5.9. Applications of Predicate Logic: Introduction to Logic Programming
- 5.9.1 Informal Presentation
- 5.9.2 Prolog Elements
- 5.9.3 Re-Evaluation and Cut-Off

5.10. Set Theory, Predicate Logic and Its Semantics
- 5.10.1 Intuitive Set Theory
- 5.10.2 Introduction to Predicate Semantics

Module 6. Artificial Intelligence and Knowledge Engineering

6.1. Introduction to Artificial Intelligence and Knowledge Engineering
- 6.1.1 Brief History of Artificial Intelligence
- 6.1.2 Artificial Intelligence Today
- 6.1.3 Knowledge Engineering

6.2. Searching
- 6.2.1 Common Search Concepts
- 6.2.2 Uninformed Search
- 6.2.3 Informed Search

6.3. Boolean Satisfiability, Constraint Satisfiability and Automatic Planning
- 6.3.1 Boolean Satisfiability
- 6.3.2 Constraint Satisfaction Problems
- 6.3.3 Automatic Planning and PDDL
- 6.3.4 Planning as a Heuristic Search
- 6.3.5 Planning with SAT

6.4. Artificial Intelligence in Games
- 6.4.1 Game Theory
- 6.4.2 Minimax and Alpha-Beta Pruning
- 6.4.3 Simulation: Monte Carlo

6.5. Supervised and Unsupervised Learning
- 6.5.1 Introduction to Machine Learning
- 6.5.2 Classification
- 6.5.3 Regression
- 6.5.4 Validation of Results
- 6.5.5 Clustering

6.6. Neural Networks
- 6.6.1 Biological Fundamentals
- 6.6.2 Computational Model
- 6.6.3 Supervised and Unsupervised Neural Networks
- 6.6.4 Simple Perceptron
- 6.6.5 Multilayer Perceptron

6.7. Genetic Algorithms
- 6.7.1 History
- 6.7.2 Biological Base
- 6.7.3 Problem Coding
- 6.7.4 Generation of the Initial Population
- 6.7.5 Main Algorithm and Genetic Operators
- 6.7.6 Evaluation of Individuals: Fitness

6.8. Thesauri, Vocabularies, Taxonomies
- 6.8.1 Vocabulary
- 6.8.2 Taxonomy
- 6.8.3 Thesauri
- 6.8.4 Ontologies

6.9. Knowledge Representation: Semantic Web
- 6.9.1 Semantic Web
- 6.9.2 Specifications: RDF, RDFS and OWL
- 6.9.3 Inference/Reasoning
- 6.9.4 Linked Data

6.10. Expert Systems and DSS
- 6.10.1 Expert Systems
- 6.10.2 Decision Support Systems

Module 7. Intelligent Systems

7.1. Agent Theory
- 7.1.1 Concept History
- 7.1.2 Agent Definition
- 7.1.3 Agents in Artificial Intelligence
- 7.1.4 Agents in Software Engineering

7.2. Agent Architectures
- 7.2.1 The Reasoning Process of an Agent
- 7.2.2 Reactive Agents
- 7.2.3 Deductive Agents
- 7.2.4 Hybrid Agents
- 7.2.5 Comparison

7.3. Information and Knowledge
- 7.3.1 Difference between Data, Information and Knowledge
- 7.3.2 Data Quality Assessment
- 7.3.3 Data Collection Methods
- 7.3.4 Information Acquisition Methods
- 7.3.5 Knowledge Acquisition Methods

7.4. Knowledge Representation
- 7.4.1 The Importance of Knowledge Representation
- 7.4.2 Definition of Knowledge Representation According to Roles
- 7.4.3 Knowledge Representation Features

7.5. Ontologies
- 7.5.1 Introduction to Metadata
- 7.5.2 Philosophical Concept of Ontology
- 7.5.3 Computing Concept of Ontology
- 7.5.4 Domain Ontologies and Higher-Level Ontologies
- 7.5.5 How to Build an Ontology?

7.6. Ontology Languages and Ontology Creation Software
- 7.6.1 Triple RDF, Turtle and N3
- 7.6.2 RDF Schema
- 7.6.3 OWL
- 7.6.4 SPARQL
- 7.6.5 Introduction to Ontology Creation Tools
- 7.6.6 Installing and Using Protégé

7.7. Semantic Web
- 7.7.1 Current and Future Status of the Semantic Web
- 7.7.2 Semantic Web Applications

7.8. Other Knowledge Representation Models
- 7.8.1 Vocabulary
- 7.8.2 Global Vision
- 7.8.3 Taxonomy
- 7.8.4 Thesauri
- 7.8.5 Folksonomy
- 7.8.6 Comparison
- 7.8.7 Mind Maps

7.9. Knowledge Representation Assessment and Integration
- 7.9.1 Zero-Order Logic
- 7.9.2 First-Order Logic
- 7.9.3 Descriptive Logic
- 7.9.4 Relationship between Different Types of Logic
- 7.9.5 Prolog: Programming Based on First-Order Logic

7.10. Semantic Reasoners, Knowledge-Based Systems and Expert Systems
- 7.10.1 Concept of Reasoner
- 7.10.2 Reasoner Applications
- 7.10.3 Knowledge-Based Systems
- 7.10.4 MYCIN: History of Expert Systems
- 7.10.5 Expert Systems Elements and Architecture
- 7.10.6 Creating Expert Systems

Module 8. Machine Learning and Data Mining

8.1. Introduction to Knowledge Discovery Processes and Basic Concepts of Machine Learning
- 8.1.1 Key Concepts of Knowledge Discovery Processes
- 8.1.2 Historical Perspective of Knowledge Discovery Processes
- 8.1.3 Stages of the Knowledge Discovery Processes
- 8.1.4 Techniques Used in Knowledge Discovery Processes
- 8.1.5 Characteristics of Good Machine Learning Models
- 8.1.6 Types of Machine Learning Information
- 8.1.7 Basic Learning Concepts
- 8.1.8 Basic Concepts of Unsupervised Learning

8.2. Data Exploration and Pre-processing
- 8.2.1 Data Processing
- 8.2.2 Data Processing in the Data Analysis Flow
- 8.2.3 Types of Data
- 8.2.4 Data Transformations
- 8.2.5 Visualization and Exploration of Continuous Variables
- 8.2.6 Visualization and Exploration of Categorical Variables
- 8.2.7 Correlation Measures
- 8.2.8 Most Common Graphic Representations
- 8.2.9 Introduction to Multivariate Analysis and Dimensionality Reduction

8.3. Decision Trees
- 8.3.1 ID3 Algorithm
- 8.3.2 C4.5 Algorithm
- 8.3.3 Overtraining and Pruning
- 8.3.4 Analysis of Results

8.4. Evaluation of Classifiers
- 8.4.1 Confusion Matrixes
- 8.4.2 Numerical Evaluation Matrixes
- 8.4.3 Kappa Statistic
- 8.4.5 The Roc Curve

8.5. Classification Rules
- 8.5.1 Rule Evaluation Measures
- 8.5.2 Introduction to Graphic Representation
- 8.5.3 Sequential Overlay Algorithm

8.6. Neural Networks
- 8.6.1 Basic Concepts
- 8.6.2 Simple Neural Networks
- 8.6.3 Backpropagation Algorithm
- 8.6.4 Introduction to Recurrent Neural Networks

8.7. Bayesian Methods
- 8.7.1 Basic Probability Concepts
- 8.7.2 Bayes' Theorem
- 8.7.3 Naive Bayes
- 8.7.4 Introduction to Bayesian Networks

8.8. Regression and Continuous Response Models
- 8.8.1 Simple Linear Regression
- 8.8.2 Multiple Linear Regression
- 8.8.3 Logistic Regression
- 8.8.4 Regression Trees
- 8.8.5 Introduction to Support Vector Machines (SVM)
- 8.8.6 Goodness-of-Fit Measures

8.9. Clustering
- 8.9.1 Basic Concepts
- 8.9.2 Hierarchical Clustering
- 8.9.3 Probabilistic Methods
- 8.9.4 EM Algorithm
- 8.9.5 B-Cubed Method
- 8.9.6 Implicit Methods

8.10. Text Mining and Natural Language Processing (NLP)
- 8.10.1 Basic Concepts
- 8.10.2 Corpus Creation
- 8.10.3 Descriptive Analysis
- 8.10.4 Introduction to Feelings Analysis

Module 9. Multiagent Systems and Computational Perception

9.1. Agents and Multiagent Systems
- 9.1.1 Concept of Agent
- 9.1.2 Architecture
- 9.1.3 Communication and Coordination
- 9.1.4 Programming Languages and Tools
- 9.1.5 Applications of the Agents
- 9.1.6 The FIPA

9.2. The Standard for Agents FIPA
- 9.2.1 Communication between Agents
- 9.2.2 Agent Management
- 9.2.3 Abstract Architecture
- 9.2.4 Other Specifications

9.3. The JADE Platform
- 9.3.1 Software Agents According to JADE
- 9.3.2 Architecture
- 9.3.3 Installation and Execution
- 9.3.4 JADE Packages

9.4. Basic Programming with JADE
- 9.4.1 The Management Console
- 9.4.2 Basic Creation of Agents

9.5. Advanced Programming with JADE
- 9.5.1 Advanced Creation of Agents
- 9.5.2 Communication between Agents
- 9.5.3 Discovering Agents

9.6. Artificial Vision.
- 9.6.1 Processing and Digital Analysis of Images
- 9.6.2 Image Analysis and Artificial Vision
- 9.6.3 Image Processing and Human Vision
- 9.6.4 Image Capturing System
- 9.6.5 Image Formation and Perception

9.7. Digital Image Analysis
- 9.7.1 Stages of the Image Analysis Process
- 9.7.2 Pre-Processing
- 9.7.3 Basic Operations
- 9.7.4 Spatial Filtering

9.8. Digital Image Transformation and Image Segmentation
- 9.8.1 Fourier Transform
- 9.8.2 Frequency Filtering
- 9.8.3 Basic Concepts
- 9.8.4 Thresholding
- 9.8.5 Contour Detection

9.9. Shape Recognition
- 9.9.1 Feature Extraction
- 9.9.2 Classification Algorithms

9.10. Natural Language Processing
- 9.10.1 Automatic Speech Recognition
- 9.10.2 Computational Linguistics

Module 10. Bio-Inspired Computing

10.1. Introduction to Bio-Inspired Computing
- 10.1.1 Introduction to Bio-Inspired Computing

10.2. Social Adaptation Algorithms
- 10.2.1 Bio-Inspired Computing Based on Ant Colonies
- 10.2.2 Variants of Ant Colony Algorithms
- 10.2.3 Particle Cloud Computing

10.3. Genetic Algorithms
- 10.3.1 General Structure
- 10.3.2 Implementations of the Main Operators

10.4. Space Exploration-Exploitation Strategies for Genetic Algorithms
- 10.4.1 CHC Algorithm
- 10.4.2 Multimodal Problems

10.5. Models of Evolutionary Computing (I)
- 10.5.1 Evolutionary Strategies
- 10.5.2 Evolutionary Programming
- 10.5.3 Algorithms Based on Differential Evolution

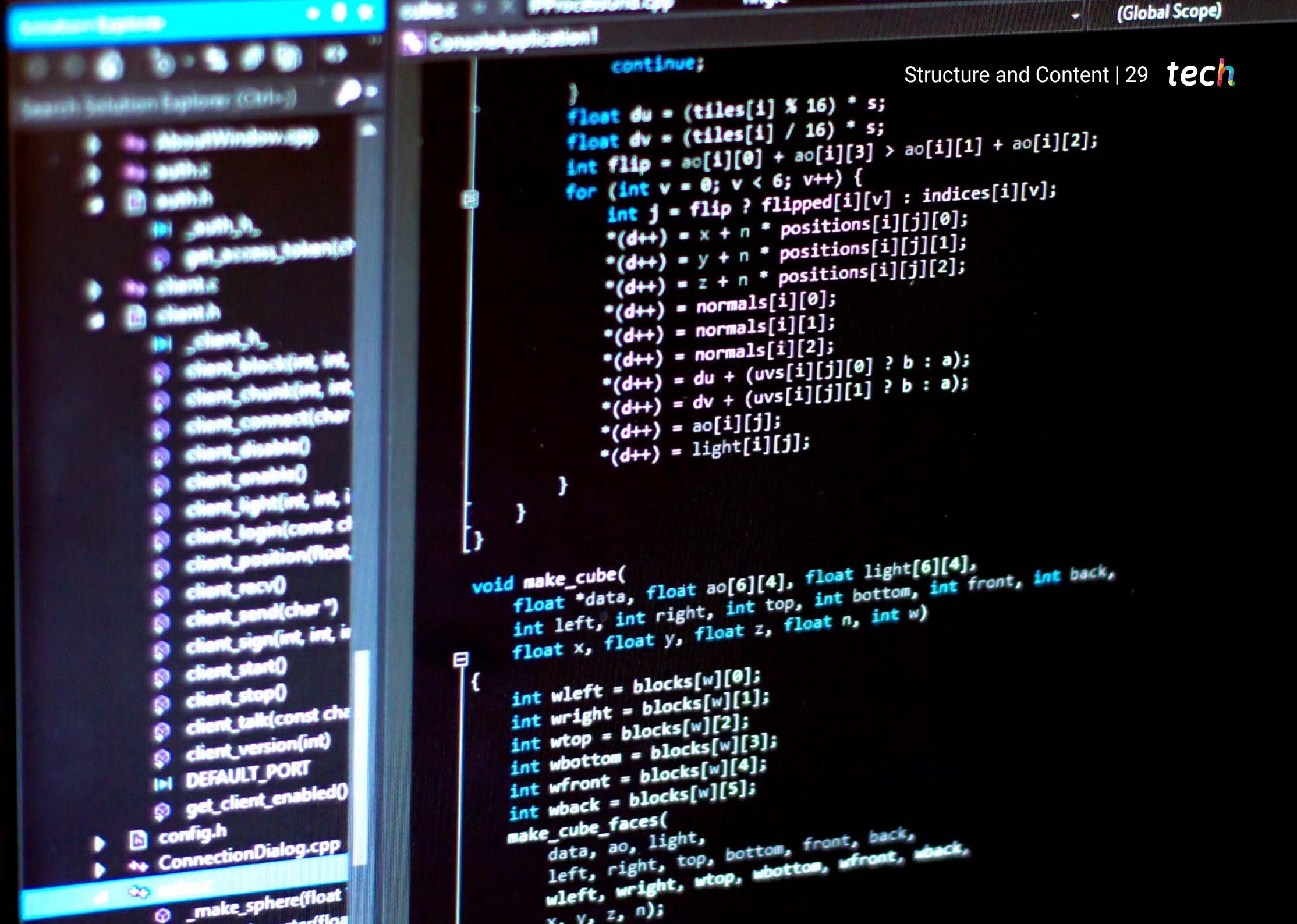
10.6. Models of Evolutionary Computing (II)
- 10.6.1 Evolution Models Based on Estimation of Distributions (EDA)
- 10.6.2 Genetic Programming

10.7. Developmental Programming Applied to Learning Disabilities
- 10.7.1 Rules-Based Learning
- 10.7.2 Evolutionary Methods in Instance Selection Problems

10.8. Multi-Objective Problems
- 10.8.1 Concept of Dominance
- 10.8.2 Application of Evolutionary Algorithms to Multi-objective Problems

10.9. Neural Networks (I)
- 10.9.1 Introduction to Neural Networks
- 10.9.2 Practical Example with Neural Networks

10.10. Neural Networks (II)
- 10.10.1 Use Cases of Neural Networks in Medical Research
- 10.10.2 Use Cases of Neural Networks in Economy
- 10.10.3 Use Cases of Neural Networks in Artificial Vision


(Global Scope)
ConsoleApplication1
continue;
}
float du = (tiles[i] % 16) * s;
float dv = (tiles[i] / 16) * s;
int flip = ao[i][0] + ao[i][3] > ao[i][1] + ao[i][2];
for (int v = 0; v < 6; v++) {
int j = flip ? flipped[i][v] : indices[i][v];
*(d++) = x + n * positions[i][j][0];
*(d++) = y + n * positions[i][j][1];
*(d++) = z + n * positions[i][j][2];
*(d++) = normals[i][0];
*(d++) = normals[i][1];
*(d++) = normals[i][2];
*(d++) = du + (uvs[i][j][0] ? b : a);
*(d++) = dv + (uvs[i][j][1] ? b : a);
*(d++) = ao[i][j];
*(d++) = light[i][j];
}
}
}
void make_cube(
float *data, float ao[6][4], float light[6][4],
int left, int right, int top, int bottom, int front, int back,
float x, float y, float z, float n, int w)
{
int wleft = blocks[w][0];
int wright = blocks[w][1];
int wtop = blocks[w][2];
int wbottom = blocks[w][3];
int wfront = blocks[w][4];
int wback = blocks[w][5];
make_cube_faces(
data, ao, light,
left, right, top, bottom, front, back,
wleft, wright, wtop, wbottom, wfront, wback,
x, y, z, n);
AboutWindow.cpp
auth.c
auth.h
_auth_h_
client.c
client.h
_client_h_
client_chunk(int, int,
client_connect(char
client_disable()
client_enable()
client_login(const ch
client_position(float
client_recv()
client_send(char *)
client_start()
client_stop()
client_version(int)
DEFAULT_PORT
get_client_enabled()
config.h
ConnectionDialog.cpp
_make_sphere(float

05

Methodology

This academic program offers students a different way of learning. Our methodology uses a cyclical learning approach: **Relearning.**

This teaching system is used, for example, in the most prestigious medical schools in the world, and major publications such as the **New England Journal of Medicine** have considered it to be one of the most effective.

Discover Relearning, a system that abandons conventional linear learning, to take you through cyclical teaching systems: a way of learning that has proven to be extremely effective, especially in subjects that require memorization"

Case Study to contextualize all content

Our program offers a revolutionary approach to developing skills and knowledge. Our goal is to strengthen skills in a changing, competitive, and highly demanding environment.

At TECH, you will experience a learning methodology that is shaking the foundations of traditional universities around the world"

You will have access to a learning system based on repetition, with natural and progressive teaching throughout the entire syllabus.

The student will learn to solve complex situations in real business environments through collaborative activities and real cases.

A learning method that is different and innovative

This TECH program is an intensive educational program, created from scratch, which presents the most demanding challenges and decisions in this field, both nationally and internationally. This methodology promotes personal and professional growth, representing a significant step towards success. The case method, a technique that lays the foundation for this content, ensures that the most current economic, social and professional reality is taken into account.

> *Our program prepares you to face new challenges in uncertain environments and achieve success in your career"*

The case method has been the most widely used learning system among the world's leading Information Technology schools for as long as they have existed. The case method was developed in 1912 so that law students would not only learn the law based on theoretical content. It consisted of presenting students with real-life, complex situations for them to make informed decisions and value judgments on how to resolve them. In 1924, Harvard adopted it as a standard teaching method.

What should a professional do in a given situation? This is the question that you are presented with in the case method, an action-oriented learning method. Throughout the course, students will be presented with multiple real cases. They will have to combine all their knowledge and research, and argue and defend their ideas and decisions.

Relearning Methodology

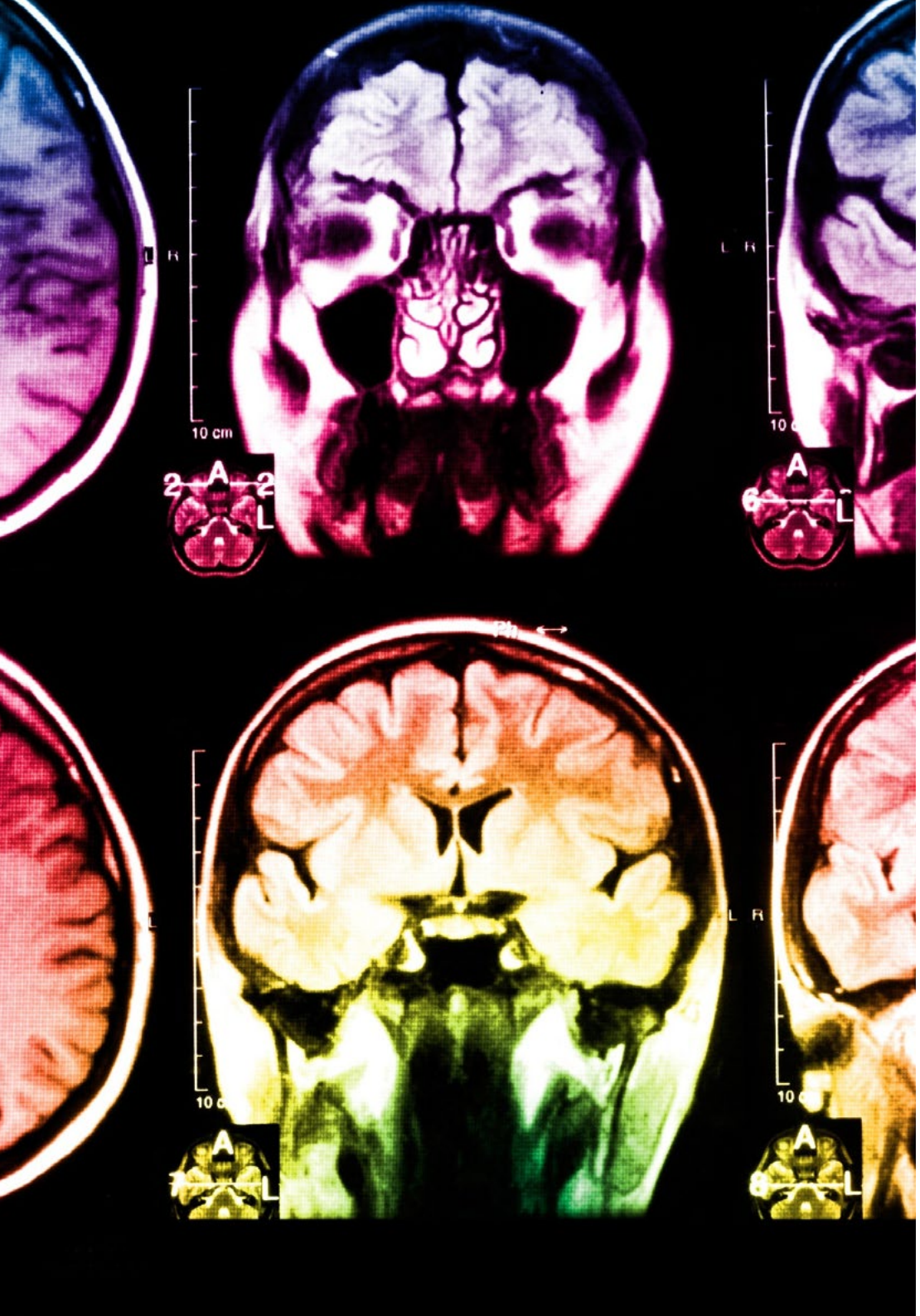
TECH effectively combines the Case Study methodology with a 100% online learning system based on repetition, which combines different teaching elements in each lesson.

We enhance the Case Study with the best 100% online teaching method: Relearning.

In 2019, we obtained the best learning results of all online universities in the world.

At TECH you will learn using a cutting-edge methodology designed to train the executives of the future. This method, at the forefront of international teaching, is called Relearning.

Our university is the only one in the world authorized to employ this successful method. In 2019, we managed to improve our students' overall satisfaction levels (teaching quality, quality of materials, course structure, objectives.) based on the best online university indicators.



In our program, learning is not a linear process, but rather a spiral (learn, unlearn, forget, and re-learn). Therefore, we combine each of these elements concentrically. This methodology has trained more than 650,000 university graduates with unprecedented success in fields as diverse as biochemistry, genetics, surgery, international law, management skills, sports science, philosophy, law, engineering, journalism, history, and financial markets and instruments. All this in a highly demanding environment, where the students have a strong socio-economic profile and an average age of 43.5 years.

Relearning will allow you to learn with less effort and better performance, involving you more in your training, developing a critical mindset, defending arguments, and contrasting opinions: a direct equation for success.

From the latest scientific evidence in the field of neuroscience, not only do we know how to organize information, ideas, images and memories, but we know that the place and context where we have learned something is fundamental for us to be able to remember it and store it in the hippocampus, to retain it in our long-term memory.

In this way, and in what is called neurocognitive context-dependent e-learning, the different elements in our program are connected to the context where the individual carries out their professional activity.

This program offers the best educational material, prepared with professionals in mind:

Study Material

All teaching material is produced by the specialists who teach the course, specifically for the course, so that the teaching content is highly specific and precise.

These contents are then applied to the audiovisual format, to create the TECH online working method. All this, with the latest techniques that offer high quality pieces in each and every one of the materials that are made available to the student.

Classes

There is scientific evidence suggesting that observing third-party experts can be useful.

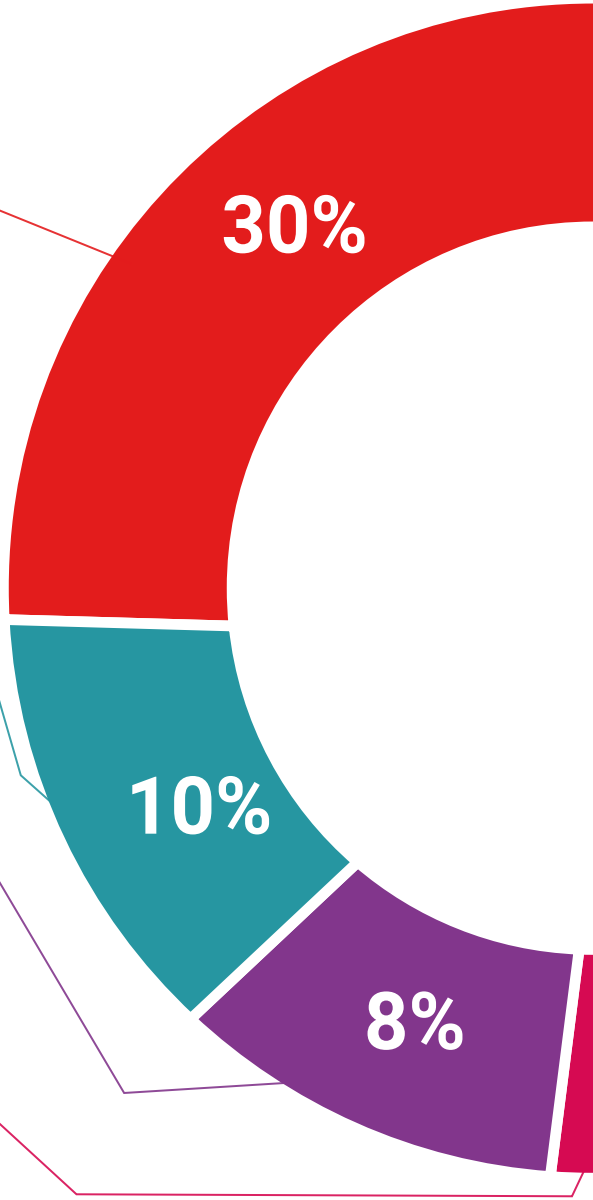
Learning from an Expert strengthens knowledge and memory, and generates confidence in future difficult decisions.

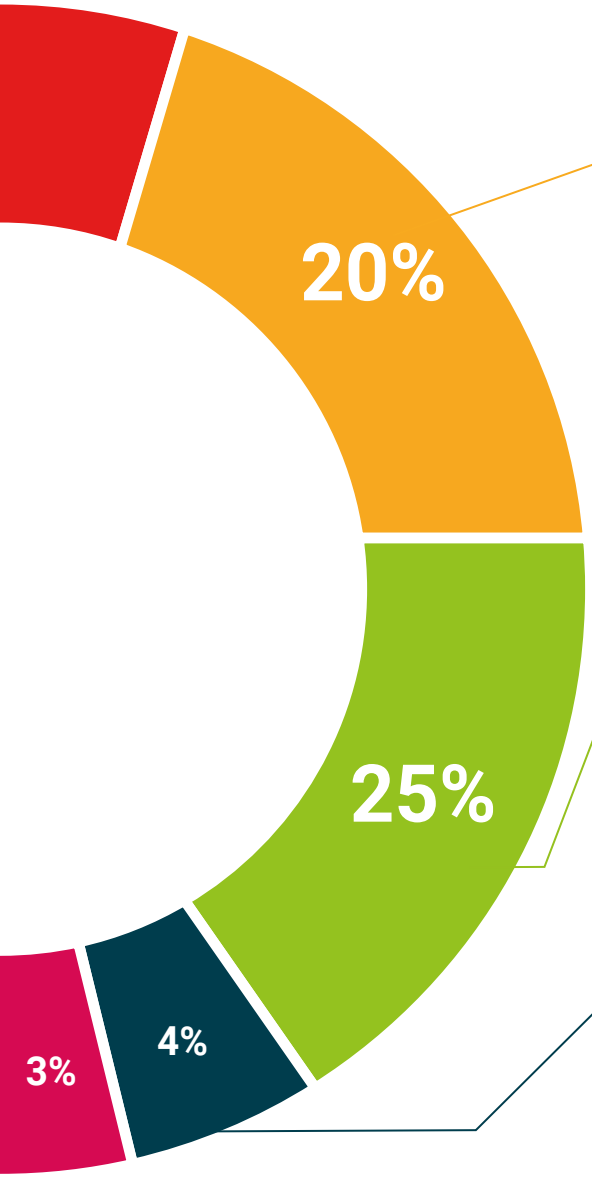
Practising Skills and Abilities

They will carry out activities to develop specific skills and abilities in each subject area. Exercises and activities to acquire and develop the skills and abilities that a specialist needs to develop in the context of the globalization that we are experiencing.

Additional Reading

Recent articles, consensus documents and international guidelines, among others. In TECH's virtual library, students will have access to everything they need to complete their course.





Case Studies

Students will complete a selection of the best case studies chosen specifically for this program. Cases that are presented, analyzed, and supervised by the best specialists in the world.

Interactive Summaries

The TECH team presents the contents attractively and dynamically in multimedia lessons that include audio, videos, images, diagrams, and concept maps in order to reinforce knowledge.

This exclusive educational system for presenting multimedia content was awarded by Microsoft as a "European Success Story".

Testing & Retesting

We periodically evaluate and re-evaluate students' knowledge throughout the program, through assessment and self-assessment activities and exercises, so that they can see how they are achieving their goals.

06

Certificate

The Professional Master's Degree in Artificial Intelligence and Knowledge Engineering guarantees students, in addition to the most rigorous and up-to-date education, access to a Professional Master's Degree issued by TECH Technological University.

Successfully complete this program and receive your university qualification without having to travel or fill out laborious paperwork"

This **Professional Master's Degree in Artificial Intelligence and Knowledge Engineering**, contains the most complete and up-to-date program on the market.

After the student has passed the assessments, they will receive their corresponding **Professional Master's Degree certificate** issued by **TECH Technological University** via tracked delivery*.

The certificate issued by **TECH University** will reflect the qualification obtained in the Professional Master's Degree, and meets the requirements commonly demanded by labor exchanges, competitive examinations, and professional career evaluation committees.

Title: **Professional Master's Degree in Artificial Intelligence and Knowledge Engineering**

Official Nº of hours: **1,500 h**.


tech technological university

Awards the following

Certificate

to

Mr./Ms. _____, with identification number _____.

For having successfully passed and accredited the following program

PROFESSIONAL MASTER'S DEGREE

in

Inteligencia Artificial e Ingeniería del Conocimiento

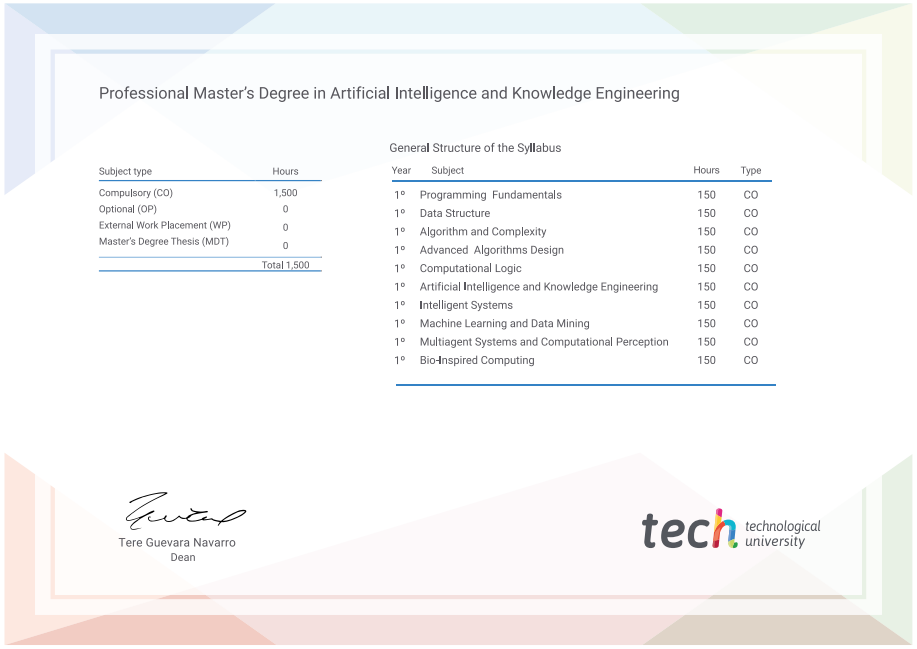
This is a qualification awarded by this University, equivalent to 1,500 hours, with a start date of dd/mm/yyyy and an end date of dd/mm/yyyy.

TECH is a Private Institution of Higher Education recognized by the Ministry of Public Education as of June 28, 2018.

June 17, 2020

Tere Guevara Navarro
Dean

This qualification must always be accompanied by the university degree issued by the competent authority to practice professionally in each country. Unique TECH Code: AFWORD23S techtitute.com/certificates


Professional Master's Degree in Artificial Intelligence and Knowledge Engineering

Subject type	Hours
Compulsory (CO)	1,500
Optional (OP)	0
External Work Placement (WP)	0
Master's Degree Thesis (MDT)	0
	Total 1,500

General Structure of the Syllabus

Year	Subject	Hours	Type
1º	Programming Fundamentals	150	CO
1º	Data Structure	150	CO
1º	Algorithm and Complexity	150	CO
1º	Advanced Algorithms Design	150	CO
1º	Computational Logic	150	CO
1º	Artificial Intelligence and Knowledge Engineering	150	CO
1º	Intelligent Systems	150	CO
1º	Machine Learning and Data Mining	150	CO
1º	Multiagent Systems and Computational Perception	150	CO
1º	Bio-Inspired Computing	150	CO

Tere Guevara Navarro
Dean

tech technological university

*Apostille Convention. In the event that the student wishes to have their paper certificate issued with an apostille, TECH EDUCATION will make the necessary arrangements to obtain it, at an additional cost.

tech technological university

Professional Master's Degree

Artificial Intelligence and Knowledge Engineering

- Modality: **online**
- Duration: **12 months**
- Certificate: **TECH Technological University**
- Dedication: **16h/week**
- Schedule: **at your own pace**
- Exams: **online**

Professional Master's Degree

Artificial Intelligence and Knowledge Engineering

tech technological university