# Universidad Simón Bolívar Course Descriptions

These translations of the Course Descriptions belong to the Computer Science Pensum of Universidad Simón Bolívar. They are providad **AS IS** without warranty of completness or correctness. They distributed under the **LGPLv3** licence.

# Table of Contents

[Universidad Simón Bolívar Course Descriptions 1](#_Toc28266478)

[Table of Contents 1](#_Toc28266479)

[CI-2511 Symbolic Logic 2](#_Toc28266480)

[CI-2525 Discrete Structures I 4](#_Toc28266481)

[CI-2526 Discrete Structures II 6](#_Toc28266482)

[CI-2527 Discrete Structures III 7](#_Toc28266483)

[CI-2611 Algorithms and Structures I 9](#_Toc28266484)

[CI-2612 Algorithms and Structures II 11](#_Toc28266485)

[CI-2613 Algorithms and Structures III 13](#_Toc28266486)

[CI-2691 Algorithms and Structures Laboratory I 15](#_Toc28266487)

[CI-2692 Algorithms and Structures Laboratory II 17](#_Toc28266488)

[CI-2693 Algorithms and Structures Laboratory III 19](#_Toc28266489)

[CI-3311 Database Systems I 20](#_Toc28266490)

[CI-3391 Database Workshop I 22](#_Toc28266491)

[CI-3641 Programming Languages 23](#_Toc28266492)

[CI-3661 Languages Workshop I 25](#_Toc28266493)

[CI-3715 Software Engineering 27](#_Toc28266494)

[CI-3725 Translators and Interpreters 29](#_Toc28266495)

[CI-3815 Computer Organization 30](#_Toc28266496)

[CI-3825 Operating Systems I 32](#_Toc28266497)

[CI-4325 User Interfaces 34](#_Toc28266498)

[CI-4835 Computer Networks I 35](#_Toc28266499)

[CI-5437 Artificial Intelligence I 37](#_Toc28266500)

[CI-5438 Artificial Intelligence II 38](#_Toc28266501)

[CO-3121 Fundamentals of Probability for Engineers 40](#_Toc28266502)

[CO-3211 Numerical Analysis for Engineers 42](#_Toc28266503)

[CO-3221 Statistics for Engineers 43](#_Toc28266504)

[CO-6612 Introduction to Artificial Neural Networks and their applications 45](#_Toc28266505)

[EP-4793 Mini Software Development Project 47](#_Toc28266506)

[FS-1111 Physics I 48](#_Toc28266507)

[FS-1112 Physics II 49](#_Toc28266508)

[MA-1111 Mathematics I 50](#_Toc28266509)

[MA-1112 Mathematics II 52](#_Toc28266510)

[MA-1116 Mathematics III 53](#_Toc28266511)

[MA-2112 Mathematics V 55](#_Toc28266512)

[MA-2115 Mathematics IV 56](#_Toc28266513)

[PS-1111 Linear Models 57](#_Toc28266514)

[PS-1115 Information Systems I 59](#_Toc28266515)

# CI-2511 Symbolic Logic

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 4 | 8 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 2 | 2 | 2 | 72 |

## Objectives

### General

Introduce the student to formal reasoning modeling systems.

### Specifics

* Develop in the student the ability to abstract a problem through its formulation in First-Order Logic.
* Introduce the use of logic as a discipline of reasoning.
* Introduce the student to inductive tests.

## Topics

* Motivation, history, formal systems. (2 hours)
* Inductive definitions.
* Recursive definitions.
* Inductive proofs. (4 hours)
* First-order logic language: constants, relationships, functions and variables.
* Usual logical connectors: disjunction, conjunction, negation, conditional and biconditional. (2 hours)
* Modeling of propositional speeches: evaluations, truth table, feasibility, tautologies, logical consequence, and logical equivalences. (6 hours)
* The formal system of propositional logic: rules of inference, syntactic derivations, theorems. (2 hours)
* Consistency, solidity, and completeness of propositional logic. (2 hours)
* Variables and quantifiers: scope of quantifiers, variable ligatures, logical sentences. (2 hours)
* Discourse modeling with predicate logic (first-order). (4 hours)
* The semantics of predicate logic: interpretations and models, satisfiability, tautologies, logical consequence and logical equivalences. (6 hours)
* Consistency and completeness of predicate logic. (2 hours)
* The formal system of predicate logic: rules of inferences, syntactic derivations, theorems. (2 hours)
* Syntactic transformations of logical expressions: replacement by equivalence, normal forms (negative, disjunctive, conjunctive and prenex), completeness of a set of formal systems connectors. (4 hours)

## Literature

* The Language of First-Order Logic (Tarski World) from Barkwise.
* Guides “Introduction to Mathematical Logic” and “Introduction to First-Order Logic”.

## Required to Complete

Three exams.

## Validity

From September 1995.

# CI-2525 Discrete Structures I

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 4 | 8 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 2 | 0 | 72 |

## Objectives

* Introduction of useful mathematical tools for modeling and analyzing discrete phenomena or processes.
* Introduction to a discrete theory analogous to differential and integral calculus.

## Topics

* 1st Class: Objectives of the combinatorics. Fundamental principles of counting. Interpretation of functions as a sequence and as assignments to boxes. The cardinality of all the functions of a set A to a set B (finite). The number of injective functions (applications of counting principles).
* 2nd Class: Application of fundamental counting principles: place objects of a set X in m boxes, objects in boxes must be ordered (discuss various methods). Strictly increasing tuples in the alphabet 1, …, n, number of m-subsets of an n-set. The number of increasing m-words in an n-alphabet.
* 3rd Class: Number of ordered and generalized n-partitions of m (two methods). Consequences: Number of increasing and strictly increasing functions. Principles of inequality or Pigeon Hole.
* 4th Class: Binomial and multinomial coefficient. Properties Sums and products. Operations on sums.
* 5th Class: Binomial and multinomial theorem. Principle of inclusion-exclusion.
* 6th Class: Applications of inclusion-exclusion type: numbers of bijective functions and partitions of a set in m classes. Sets of objects that satisfy exactly certain properties. Derangements.
* 7th Class: Recurrence equations. Formulation Examples Direct Resolution and Combination methods.
* 8th Class: Linear recurrence equations with constant coefficients. Resolution Method.
* 9th Class: Resolution by generating functions. Equation and resolution to parenthesize an expression.
* 10th Class: Calculation of differences. Similarity with differential calculus. Differences of a polynomial.
* 11th Class: Properties of the operator difference. Factorial functions. Stirling numbers of the first kind.
* 12th Class: Second kind Stirling numbers. Combinatorial interpretation. The difference of special functions.
* 13th Class: Development in differences. Gregory-Newton’s formula. Sum calculations. Similarity with the integral calculation. Sum Operator. Properties Sums of special functions.
* 14th Class: Sums Calculation’s Fundamental Theorem. Abel Transforms. Other methods for calculating series.
* 15th Class: Asymptotic Behavior. Comparison relationships. Notation O, o, Teta, Omega, equivalent asymptotes.
* 16th Class: Calculation on comparison relations.
* 17th Class: Comparison scales.
* 18th Class: Estimates and asymptotic representations.
* 19th Class: Practical calculation of the complexity of algorithms.

## Literature

* D. Knuth. Fundamental Algorithms. Addison-Wesley. 1973.
* M. Spiegel. Theory and problems of calculus of finite differences and difference equations. Shaum’s Outline series. McGraw-Hill. 1971.
* C. Liu. Introduction to Combinatorial Mathematics. McGraw-Hill. 1968.
* C. Berge. Principles of Combinatorics. Academic Press. 1971.
* Jordan. Calculus of finite difference equations.
* Grahan, Knuth, Patasnik. Concrete Mathematics. Addison-Wesley. 1991.
* N. Xuong, Mathematiques Discretes. Dunod. 1992.

## Required to Complete

Two exams, two quizzes, and homework.

## Validity

From 2011.

# CI-2526 Discrete Structures II

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 4 | 8 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 2 | 0 | 72 |

## Objectives

### General

Definition of basic abstract concepts in computer science. Apply demonstrative arguments within a precisely defined context.

### Specifics

* Knowledge of structures defined in abstract form.
* Knowledge of the concept of relationship, relational structures, and its application
* Basic knowledge of cardinality

## Topics

* Definition of: Set, axioms of extension, empty set, separation, union. Paradoxes. Operations: intersection, union, difference, and complement. Properties
* Families: Definition and operations Axiom set of parts or power set. Properties
* Cartesian product. Definition and Properties. Relationships: Definition, operations, and properties of operations. Domain, Rank and graph of a relationship. Inverse relationship. Left and right restrictions of a relationship. Reflective, symmetric, antisymmetric and transitive properties among others. Sequencing and composition of relationships. Properties.
* Partial, total and topological order. Well-ordering principle.
* Induction. Axioms of Peano. Inductive definitions. Closure of Relationships definition, reflective, symmetric and transitive closures.
* Equivalence Relation. Definition and Properties. Definition of Partition and its Properties.
* Function: Definition, types, properties. Composition of functions and properties. Inverse function Reverse right, left.
* Definition of a finite set, properties. Infinite sets, countable and non-countable sets.

## Literature

* Elementos de Teoría Axiomática de Conjuntos. Vicente Yriarte. USB.
* Discrete Mathematics in Computer Science. Donald F. Stanat & David Mc Allister. Prentice-Hall. 1977.
* A Logical Approach to Discrete Math. David Gries & Fred B. Schneider. Springer Verlag. 1993.

## Required to Complete

Three exams.

## Validity

From January 2011.

# CI-2527 Discrete Structures III

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 4 | 8 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 2 | 0 | 72 |

## Objectives

### General

Definition of fundamental mathematical concepts of Computer Science with a unified and precise terminology. Exercise of formal reasoning with abstract entities.

### Specifics

* Study of abstract algebraic systems: Semigroups, Monoids, Groups, Boolean Algebras, and others.
* Study of basic concepts of Logic Design: Commutative Circuits, Karnaugh Maps.

## Topics

* Integers: Integrity Domains, Ordered and Well Ordered Domains, Prime Numbers, Congruences.
* Introduction to Algebras: Groupoids, Semigroups, Monoids, Subalgebras, Homomorphisms.
* Groups: Laws, Powers, Order. Subgroups Homomorphisms of Groups. Cosets. Cyclic Groups Permutations, Groups of Transformations. Quotient Group.
* Heterogeneous Algebras: Example - Linear Algebra.
* Boolean Algebras: Laws, Boolean Expressions, Partial Order (Reticulated), Atoms, Sum of Atoms, Finite Boolean Algebras.
* Logical Design: Commutative Circuits, Commutative Algebras, Min / Max terms, Shannon Theorem, Adjacencies, Karnaugh Maps.

## Literature

* D. Gries & F.B. Schneider, A Logical Approach to Discrete Math., Springler-Verlag, 1993.
* J.D. Lipson, Elements of Algebra and Algebraic Computing, Addison-Wesley, 1981.
* A. Lloris & A. Prieto, Diseño Lógico, McGraw Hill, 1996.
* V. Yriarte, Estructuras Discretas II, Universidad Simón Bolívar, 2001 (available at <http://www.ldc.usb.ve/~yriarte> ).
* K.A. Ross & C. Wright, Discrete Mathematics, Prentice-Hall, 1993.
* J.B. Fraleigh, A First Course in Abstract Algebra, Addison-Wesley, 4th Edition, 1989.
* B. Kolman & R.C. Busby, Estructuras de Matemáticas Discretas para la Computación, Prentice Hall Hispanoamericana, 1986.
* J.R. Durbin, Modern Algebra, John Wiley & Sons, 1979.
* F. Preparata & R.T. Yeh, Introduction to Discrete Structures for Computer Science and Engineering, Addison-Wesley, 1973.
* M. Morris Mano & C.R. Kime, Fundamentos de Diseño Lógico y Computadoras, Prentice Hall Hispanoamericana, 1998.
* D.D. Gajski, Principles of Digital Design, Prentice-Hall, 1997.
* J.F. Wakerly, Diseño Digital: Principios y Prácticas, Prentice Hall Hispanoamericana, 1992.
* G.E. Hoernes & M.F. Heilweil, Introducción al Álgebra de Boole y los Dispositivos Lógicos, Ma. Graw Hill, 1972.
* G.E. Williams, Boolean Algebra with Computer Applications, Ma. Graw Hill, 1970.

## Required to Complete

Three exams and homework.

## Validity

From 2011.

# CI-2611 Algorithms and Structures I

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 3 | 6 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 3 | 1 | 0 | 48 |

## Objectives

### General

At the end of the course, the student is qualified to formally specify a simple computational problem and design a correct algorithmic solution under the approach of structured programming and descending analysis.

### Specifics

* Interpret an algorithm specified in the Guarded Command Language (GCL) formalism.
* Specify the inputs and outputs of a problem using the language of logic.
* Make the descending analysis of a computational problem.
* Conceive an algorithmic solution for a simple computational problem.
* Write algorithms using GCL formalism.
* Represent information with basic, enumerated, sequential and structured non-recursive data types.
* Develop algorithms that include data and functions from a specification.
* Formally verify that an algorithm is correct.

## Topics

* Definition of Algorithm. Specification, Programming, and Correction of programs. (2 hours)
* The notion of Sets, Sequence and Aggregation Functions (sum, production, maximum, minimum, count). The notion of constant and variable. Basic Types (integer, character, real, boolean). Input and output specifications, precondition and postcondition. (4 hours)
* The general structure of a program in GCL. Ordinal types. Arithmetic and logical expressions, precedence. Assignment, null instruction (skip) and sequencing. (2 hours)
* Control structures: selection and iteration. (2 Hours)
* Arrays. Using iteration with arrays. Multidimensional Arrays Nested Iterations. (2 hours)
* Program correction: Hoare triples, weakest precondition, general rules, assignment, skip, sequencing and selection (2 Hours).
* Correction of programs: invariants, bounding functions, invariance theorem, iteration rules (2 Hours).
* Program correction: intuitive search for invariants. (2 hours)
* Descending Analysis. Subprograms (functions and procedures). Parameters of passing. (2 hours)
* Recursion: Recursive definitions, calls to recursive subprograms. Tail Recursion. (4 hours)
* Correction of programs: Techniques for invariant derivation: elimination of a predicate from a conjunction, replacement of constants by variables, strengthening of invariants. (4 hours)
* Correction of programs: procedure call rule, function call rule, array modification rule, recursive call rule. (2 hours)

## Literature

* Oscar Meza. Introducción a la Programación. 2000. Available at: <http://ldc.usb.ve/~meza/ci-2615/>
* Kaldewaij Anne. “Programming: the derivation of algorithms”. Prentice-Hall. 1990. ISBN- 0-13-204108-1. Chapters 1, 2, 3 and 4.
* Gries David. “The Science of Programming”. Springer.Verlag.1981. ISBN 0-387-96480-0. Pages 1-85 and 310-319.
* Gries David, Schneider Fred. “A Logical Approach to Discrete Math”. Springer-Verlag. 1993.
* Jesús Ravelo. Ejemplos de Especificación de Problemas Algorítmicos. 2009. Available at: <http://ldc.usb.ve/~jravelo/docencia/algoritmos/material/especs.pdf>

## Required to Complete

Three exams and homework.

## Validity

From January 2015.

# CI-2612 Algorithms and Structures II

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 3 | 6 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 3 | 1 | 0 | 48 |

## Objectives

### General

At the end of the course, the student is expected to be able to design and implement an Abstract Data Type (ADT), apply classical algorithms and data structures in solving computational problems and analyze the complexity of simple algorithms.

### Specifics

* Interpret the behavior of an ADT in the model-based specification style.
* Formally specify an ADT using the style based models described by first-order logic and mathematical structures (sets, multi-sets, sequences, relationships, and functions).
* Use data refining techniques that ensure the consistency of the models in the implementation of an ADT.
* Select data structures to implement an ADT using a time and resource efficiency criteria.
* Solve problems by using some ADTs (ADT Stack, ADT Queue, ADT Dictionary, among others) and classic algorithms (Search and Sorting algorithms).
* Represent information with recursive structured data types.
* Design structured programs that include data and recursive functions.
* Apply the basic notions of algorithm analysis in simple cases.

## Topics

* Abstract Data Type (ADT): Concept. Formal specifications with models. For example, the Dictionary. Abstract models of representation: sets, multisets, sequences, relationships, and functions. Implementation: data refinement, concrete model, invariant coupling. The correctness of data refinement. Encapsulation and concealment. Modules and lessons. (6 hours)
* Concrete Data Types: arrays, registers, references or pointers. Implementation of recursive types with single and double linked structures. Introduction to Hash tables. (4 hours)
* Free Algebraic Types: examples (expressions), the definition of functions, demonstration of properties, programming primitives, programming of iterative and recursive examples. (4 hours)
* Concepts of Algorithm Complexity Analysis (asymptotic notation; recurrence management; Master Theorem). Application of these concepts in the analysis of classical algorithms (2 hours)
* Search and Sorting Algorithms: sequential search, binary search, insertion sorting, selection sorting, swap sorting (bubble sort), merge sorting (Mergesort), quick sorting (Quicksort), mound sorting (Heapsort). (4 hours)
* Trees. Binary Search Trees: definition, properties, search, insertion, and deletion. Implementation. (6 hours)
* Recursion vs. Iteration. Examples on trees. Use of auxiliary structures (stack and queue) to implement iterative algorithms. (4 hours)

## Literature

* T.H. Cormen, C.E. Leiserson, R.L. Rivest & C. Stein, Introduction to Algorithms, The MIT Press, 3rd edition, 2009. Chapters 1, 2, 3 and 4; 6 and 7; 10, 11, 12 and 13; Appendix A.
* J.N. Ravelo, Especificación e Implementación de Tipos Abstractos de Datos, Universidad Simón Bolívar, 2012, available at <http://www.ldc.usb.ve/˜jravelo/docencia/algoritmos/material.>.
* J.N. Ravelo & K. Fernández, Tipos Algebraico-Libres, Universidad Simón Bolívar, 2012, available at <http://www.ldc.usb.ve/˜jravelo/docencia/algoritmos/material.>.
* J.N. Ravelo & K. Fernández, Ordenamiento sobre Arreglos, Universidad Simón Bolívar, 2012, available at <http://www.ldc.usb.ve/˜jravelo/docencia/algoritmos/material.>.
* N. Wirth, Algorithms + Data Structures = Programs, Prentice-Hall, 1976. Chapters 1 and 4.
* B. Liskov con J. Guttag, Program Development in Java – Abstraction, Specification, and Object-Oriented Design, Addison-Wesley, 2001. Chapters 1 al 10.
* R. Mitchell, Abstract Data Types and Modula-2, Prentice-Hall, 1992. Chapters 1, 2 and 4.
* C. Morgan, Programming from Specifications, Prentice Hall, 2nd edition, 1994. Chapter 9.
* S. Thompson, Haskell – The Craft of Functional Programming, Addison-Wesley, 1996. Chapter 10.
* A.V. Aho, J.E. Hopcroft, J.D. Ullman, Data Structures and Algorithms, Addison-Wesley, 1983.

## Required to Complete

Two exams.

## Validity

From January 2015.

# CI-2613 Algorithms and Structures III

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 3 | 6 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 3 | 1 | 0 | 48 |

## Objectives

### General

At the end of the course, the student is expected to be able to concepts of Graph Theory in order to model and solve problems efficiently.

### Specifics

* Model problems using Graph Theory.
* Identify the algorithms that allow solving efficiently the problems modeled with Graph Theory.
* Evaluate the different algorithms available to solve the problems presented.
* Calculate various types of routes and coverages in graphs, according to specific conditions.
* Infer from the routes and coverages calculated other interesting results from Graph Theory.
* Establish the possible modifications to be made to the classical algorithms to solve the problems presented.

## Topics

* Graphs: motivation, concept, characteristics, properties, types of graphs. (2 hours)
* Graph representation: lists of edges, adjacency matrices, incidence matrices, adjacency lists, incidence lists. Impact of the underlying structures in the implementation. (4 hours)
* Connectivity in graphs: paths, cycles, routes, reach, transitive closure, Roy Warshall algorithm, connected and strongly connected components, articulation points. (4 hours)
* Graph paths: general labeling model, general search model. General models instance: depth-first search algorithm (DFS) and breadth-first search algorithm (BFS). (4 hours)
* Route applications: calculation of connected components, determine if a graph is bipartite, calculation of strongly connected components, points of articulation and others. (4 hours)
* Minimum and maximum cost paths. Floyd-Warshall algorithm. Dijkstra’s algorithm. Bellman’s algorithm. (4 hours)
* Precedence graphs: leveling, topological ordering, minimum cost paths. (4 hours)
* Informed search. Heuristics and algorithm A\*. (2 hours)
* Trees and Arborescences. Properties. (2 hours)
* Minimum Spanning tree: Prim’s Algorithm. Kruskal’s Algorithm. Data structures for disjoint sets. (4 hours)

## Literature

* Ortega, Maruja, Meza, Oscar. “Grafos y Algoritmos”. Equinoccio Editorial, USB. Caracas, 1993.
* Sedgewick, Robert, and Wayne, Kevin. “Algorithms”, 4th edition. Addison-Wesley. 2011.
* Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C., Introduction to Algorithms, The MIT Press, 3rd edition, 2009. Chapters 1, 2, 3 and 4; 6 and 7; 10, 11, 12 and 13; Appendix A.
* Brassard, Gilles, and Bratley, Paul. “Fundamentals of Algorithmic”. Prentice-Hall. 1995.
* Aho, Alfred, Hopcroft, John, and Ullman, Jeffrey. “Estructuras de Datos y Algoritmos”. Addison-Wesley. 1983.
* Bondy, J.A. and Murty, U.S.R. “Graph Theory”, 3rd edition. Springer. 2008.
* Diestel, Reinhard. “Graph Theory”, 4th edition. Springer-Verlag. 2010.

## Required to Complete

Two exams.

## Validity

From January 2015.

# CI-2691 Algorithms and Structures Laboratory I

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 2 | 4 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 0 | 0 | 3 | 36 |

## Objectives

### General

At the end of the course, the student is expected to be able to implement a correct algorithmic solution to a simple computational problem, based on its formal specification, making use of an imperative language under the structured programming approach.

### Specifics

* Interpret an algorithm specified in the Guarded Command Language (GCL) formalism.
* Specify the inputs and outputs of a problem using a logical language.
* Make the descending analysis of a computational problem.
* Conceive an algorithmic solution for a simple computational problem.
* Write algorithms using GCL formalism.
* Represent information with basic, ordinal, sequential and structured non-recursive data types.
* Design structured programs that include data structures and subprograms.
* Process data contained in text files.
* Encode in some imperative language an algorithm specified in GCL, following the standards of good programming style.
* Verify the assertions of a program and the status of the variables through actions inserted in the code to support the development process.
* Write technical reports on program development.
* Apply pair programming practices, sustained rhythm, and development of test cases, from the agile development processes.

## Topics

* Familiarization with the work environment. The general structure of a program in an imperative language. Declaration of constants and variables. Basic Types (integer, character, real, boolean). Entry and exit instructions. Input and output specifications, precondition and postcondition. Comments. Code documentation. (3 hours)
* Ordinal types. Arithmetic and logical expressions, precedence. Assignment, null instruction, and sequencing. Indentation. Code documentation. Writing intermediate assertions as logical expressions. (3 hours)
* Control structures: selection and iteration. Equivalences between iterative instructions. Code Documentation (3 hours)
* Arrays. Use of loops with arrays. Multidimensional Arrays. Nested Iterations. Structured type constructor. Definition of new types. Arrays of Structures. (3 Hours)
* Verification of the assertions and the state of the variables corresponding to the preconditions, postconditions, instructions of assignment, instructions of sequencing, instructions of selection, invariants and bounding function of an iteration, by means of actions inserted in the code. (3 hours).
* Descending analysis. Non-recursive subprograms. Parameters passing by value and by reference, and their correspondence with the input, output and input-output parameters in GCL. Reach of identifiers. (3 hours)
* Principles of agile development processes: pair programming practices, sustained rhythm, and development of test cases. (3 Hours)
* Sequential files. (3 Hours)
* Recursion. (3 Hours)

## Literature

* Manual of the programming language used.
* Oscar Meza. Introducción a la Programación. 2000. Available at: <http://ldc.usb.ve/~meza/ci-2615/>
* Gries David. “The Science of Programming”. Springer.Verlag.1981. ISBN 0-387-96480-0. Pages 1-85 y 310-319
* Gries David, Schneider Fred. “A Logical Approach to Discrete Math”. Springer-Verlag. 1993.
* Jesús Ravelo. Ejemplos de Especificación de Problemas Algorítmicos. 2009. Available at: <http://ldc.usb.ve/~jravelo/docencia/algoritmos/material/especs.pdf>.

## Required to Complete

Laboratory Practices, Quizzes, Closed Workshops, and Projects.

## Validity

From January 2015.

# CI-2692 Algorithms and Structures Laboratory II

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 2 | 4 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 0 | 0 | 3 | 36 |

## Objectives

### General

At the end of the course, the student will be able to model and solve a problem that may arise, using Abstract Data Types (ADT) and algorithms known or designed by him. Additionally, the student must be able to make complexity comparisons between different solutions to the same problem.

### Specifics

* Represent as an ADT a concept of a given problem.
* Implement an ADT based on its formal specification.
* Empirically analyze the cost in time and space of the implementation of an ADT.
* Evaluate different data structures to represent ADTs conveniently.
* Implement search and sorting algorithms.
* Select the search or sorting algorithm that best suits a problem.
* Develop and maintain software libraries systematically.
* Code programs using good techniques and style.
* Design and apply unit testing strategies and functional tests.

## Topics

* Introduction to the programming language (imperative, which allows encapsulation and concealment of information) to use. Programming styles. (3 hours)
* Introduction to the implementation of ADTs. For this, the development of a library for the management of a particular ADT is assigned. Introduction to unit tests. (3 hours)
* Implementation of free algebraic types (3 hours).
* Dynamic implementation of simple types using the referencing operation (3 hours).
* Implementation of some of the most used ADTs: Set, Sequence, Stack, Queue, Dictionary, Binary Trees. Hash Tables (6 hours)
* Implementation of the solution of a problem using the ADTs of the software library that is built progressively. Introduction to functional tests (3 hours)
* Study and implementation of search and sorting algorithms. (6 hours)
* An empirical study of the cost of an algorithm. (3 hours)

## Literature

* T.H. Cormen, C.E. Leiserson, R.L. Rivest & C. Stein, Introduction to Algorithms, The MIT Press, 3rd edition, 2009. Chapters 1, 2, 3 and 4; 6 and 7; 10, 11, 12 and 13; Appendix A.
* J.N. Ravelo, Especificación e Implementación de Tipos Abstractos de Datos, Universidad Simón Bolívar, 2012, available at <http://www.ldc.usb.ve/˜jravelo/docencia/algoritmos/material>.
* J.N. Ravelo & K. Fernández, Tipos Algebraico-Libres, Universidad Simón Bolívar, 2012, available at <http://www.ldc.usb.ve/˜jravelo/docencia/algoritmos/material>.
* J.N. Ravelo & K. Fernández, Ordenamiento sobre Arreglos, Universidad Simón Bolívar, 2012, available at <http://www.ldc.usb.ve/˜jravelo/docencia/algoritmos/material>.
* N. Wirth, Algorithms + Data Structures = Programs, Prentice-Hall, 1976. Chapters 1 and 4.
* Manual of the programming language used.

## Required to Complete

Laboratory Practices, Quizzes, Closed Workshops, and Projects.

## Validity

From January 2015.

# CI-2693 Algorithms and Structures Laboratory III

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 2 | 4 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 0 | 0 | 3 | 36 |

## Objectives

### General

At the end of the course, the student must be able to propose and implement effective and efficient solutions to problems posed using graph theory.

### Specifics

* Design efficient applications using Graph Theory that model problems of interest.
* Evaluate the most appropriate structures to implement your solutions.
* Analyze and select the algorithms that best adapt to problem-solving.
* Develop adaptations to fundamental algorithms on graphs.

## Topics

* Introduction to the programming language (imperative and object-oriented, which allows encapsulation and concealment of information) to be used. (3 hours)
* Efficient representation of graphs, the study of the complexity of operations for various representations. (3 hours)
* Graph paths: general labeling and search model. Deep First Search Algorithms (DFS). Application of this algorithm. (3 hours)
* Graph routes: Breath First Search (BFS). Applications of this algorithm. (3 hours)
* Connectivity in graphs: scope, transitive closure, Roy Warshall’s algorithm, connected and strongly connected components, articulation points. (3 hours)
* Minimum and maximum cost paths. Dijkstra and Bellman’s algorithms. Implementation based on priority queues: representation model with heaps. (3 hours)
* Precedence graphs. Leveling, Topological ordering. (3 hours)
* Trees and Arborescences. Properties Minimum tree cover: Prim and Kruskal algorithm. Efficient implementations using Disjoint Sets with their optimizations: balancing and flattening trees. (3 hours)

## Literature

* Ortega, Maruja, Meza, Oscar. “Grafos y Algoritmos”. Equinoccio Editorial, USB. Caracas, 1993.
* Aho; Hopcroft; Ullman. “Estructuras de Datos y Algoritmos”. Addison-Wesley.
* Niklaus. Wirth . “Algoritmos y Estructuras de Datos”. Prentice-Hall Editorial.
* Manual of the programming language used.

## Required to Complete

Laboratory Practices, Quizzes, Closed Workshops, and Projects.

## Validity

From January 2015.

# CI-3311 Database Systems I

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 3 | 6 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 3 | 0 | 0 | 36 |

## Objectives

### General

* Study the concepts of Database (DB) Theory that allow data modeling.
* Study the relational model and its associated languages.
* Introduce what are the different tools and functions of a database manager.

### Specifics

* Study the concept of databases, database manager, database systems and the differences that exist between them and the file managers.
* Model a DB with the Entity-Relationship model and the Extended Entity-Relationship model.
* Master the basics of conceptual data modeling.
* Study the concepts of Functional Dependency and Normalization. Algorithms and heuristics to obtain normal forms.
* Study the relational model and its main data manipulation languages.
* Understand the basic notions of the catalog, concurrency management, and recovery, integrity and security.

## Topics

* Introduction. What is data, what is information and what is knowledge. Data processing, information, and knowledge system. File systems. (2 hours)
* Basic concepts of databases. Databases Database Managers. (2 hours)
* Differences between DB Managers and File Managers.
* ER modeling. (3 hours)
* EER modeling. (2 hours)
* Relational model. Relational Model Structures. Main features. (1 hour)
* ER and EER translation to the Relational Model. (2 hour)
* Normalization. What is normalization? Why is it necessary. Functional Dependency. (4 hours)
* Different normal forms: 1FN, 2FN, 3FN, BCNF. How to achieve different normal forms. Manipulation Languages of the Relational Model. Algebra. How to Calculate it. SQL. (9 hours)
* Network and Hierarchical Models. (2 hours)
* Relational Database Management System (RDBMS) Structure. What is a data dictionary? Elements of a data dictionary. Data catalog. (2 hours)
* Concurrence. Serialization interference problems. Deadlock locks. (3 hours)
* Recovery, integrity, and security. What is recovery? Commit, Rollback transactions. Recovery algorithms Integrity rules Security. (5 hours)
* Advanced DataBases (2 hours).

## Literature

* Elsmari, Navathe. Fundamentals of Database Systems. Benjamin/Cummings, 1989. Chapters 1, 2, 3, 6, 7, 8, 13, 14, 17, 19, 20, 21, 22.
* Jeffrey D. Ullman. Principles of Database Systems. Vol. 1, Computer Science Press, 1988.
* Batini, Ceri, Navathe. Conceptual Database Design. Chapter 5.

## Required to Complete

Three exams.

## Validity

From September 1995.

# CI-3391 Database Workshop I

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 2 | 4 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 0 | 0 | 3 | 36 |

## Objectives

### General

At the end of the course, the student must be able to apply the concepts of database design.

### Specifics

* Conceptual modeling of a database using the ER model and extensions proposed to it.
* Logical design of databases in the relational model.
* Database implementation design in an Relational Database Management System (RDBMS).
* Specification of requirements in SQL and / or procedural extensions of this language.
* Development of client/server applications using SQL and client communication libraries / Database Management System (DBMS).

## Topics

* Conceptual modeling using the ER model.
* Conceptual modeling using proposed extensions to the ER model.
* Translation of ER schemes extended to relational schemes.
* Normalization of relational schemes.
* SQL Data Definition / Manipulation Language.
* SQL procedural extensions (depends on the DBMS selected for the course).
* Use of client/server libraries (depends on the DBMS used).

## Literature

* Elsmari and Navathe S. “Fundamentals of Database Systems”. Addison-Wesley. Third edition. The Benjamin/Cumming Publishing Company. 1990
* Ullmann, J. “Principles of Knowledge and Database Systems”. Computer Science Press, 1998
* Manuals of the RDBMS used in the course.

## Required to Complete

Two exams, Projects and Homework.

## Validity

From January 1995.

# CI-3641 Programming Languages

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 3 | 6 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 3 | 0 | 0 | 36 |

## Objectives

### General

Study of the fundamental characteristics of high-level languages and the programming paradigms that they implement.

### Specifics

* Study of programming languages and the main computer science paradigms.
* Familiarization with the basic notions of the imperative programming model.
* Study of the programming paradigm with objects and the alternatives offered by high-level languages for the implementation of abstract types.
* Familiarization with the basic concepts and techniques of concurrent programming.
* Knowledge of the alternatives proposed by the functional and logical paradigms.
* Ability to compare between different languages and their adequacy to solve specific problems.
* Establishment of parameters for comparing programming languages.

## Topics

* Introduction. Historical introduction. Syntax and semantics. High, medium and low-level languages. General-purpose and specific purpose languages. Programming paradigms.
* Imperative Programming. Von Neumann architecture. Variables and memory states. Control flow.
* Variables and memory states. Type, value, scope and lifetime of a variable. Local and global variables. Scope and context: Static and dynamic model. Types of parameter passing. Types of abstract data and programming with objects. Evolution of the type concept. Encapsulation and privacy. Objects and classes. Generality, operator overload, and polymorphism. Reusability. Dynamic message-passing model.
* Control flow. General Transfers. Hierarchies of control structures. Transformations between models. Structured programming. Transfers between units: subroutines, coroutines, concurrent units. Concurrent Programming Processes and interaction between processes. Concurrence and non-determinism. Explicit and implicit parallelism. Mechanism of synchronization and mutual exclusion. “Deadlocks.” Synchronization between processes that share memory. Communicating processes.
* Logic Programming. Declarative or non-procedural languages. Logical languages and theorem proving. Knowledge bases. Inference Mechanism.
* Functional Programming. Essential features. Application languages: objects, functions and functional forms. Functional elements of traditional languages. Partially functional languages.
* Compared languages. Comparison between languages. Implicit and explicit parallelism in non-imperative models. Multiparadigm languages. Elements of analysis and design of programming languages.

## Literature

* Andrews, G.R. y Scheneider, F.B.: “Concepts and Notations for Concurrent programming”, Computing Surveys, Vol. 15, No. 1, March 1983, pp. 340.
* Ben-Ari, M.: Principles of Concurrent Programming, Second Edition. John & Sons. 1987.
* Ghezzi, C. y Jazayeri, M.: Programming Language Concepts, Second Edition, John Wiley & Sons. 1987.
* Ledgard, H.F. y Marcotty, M.: “A Genealogy of Control Structures”, CACM, Vol. 18, No. 11, November 1975, pp. 629-639.

## Required to Complete

Three exams.

## Validity

From September 1989.

# CI-3661 Languages Workshop I

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 2 | 4 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 0 | 0 | 3 | 36 |

## Objectives

### General

Study of the fundamental characteristics of high-level languages and the programming paradigms that they implement, complementing the course CI-3641 Programming Languages at the practical level.

### Specifics

* Study of programming languages and the main computer science paradigms.
* Familiarization with the basic notions of the imperative programming model.
* Study of the programming paradigm with objects and the alternatives offered by high-level languages for the implementation of abstract types.
* Familiarization with the basic concepts and techniques of concurrent programming.
* Knowledge of the alternatives proposed by the functional and logical paradigms.
* Ability to compare between different languages and their adequacy to solve specific problems.
* Establishment of parameters for comparing programming languages.

## Topics

* Introduction. Historical introduction. Syntax and semantics. High, medium and low-level languages. General-purpose and specific purpose languages. Programming paradigms.
* Imperative Programming. Von Neumann architecture. Variables and memory states. Control flow.
* Variables and memory states. Type, value, scope and lifetime of a variable. Local and global variables. Scope and context: Static and dynamic model. Types of parameter passing. Types of abstract data and programming with objects. Evolution of the type concept. Encapsulation and privacy. Objects and classes. Generality, operator overload, and polymorphism. Reusability. Dynamic message-passing model.
* Control flow. General Transfers. Hierarchies of control structures. Transformations between models. Structured programming. Transfers between units: subroutines, coroutines, concurrent units. Concurrent Programming Processes and interaction between processes. Concurrence and non-determinism. Explicit and implicit parallelism. Mechanism of synchronization and mutual exclusion. “Deadlocks.” Synchronization between processes that share memory. Communicating processes.
* Logic Programming. Declarative or non-procedural languages. Logical languages and theorem proving. Knowledge bases. Inference Mechanism.
* Functional Programming. Essential features. Application languages: objects, functions and functional forms. Functional elements of traditional languages. Partially functional languages.
* Compared languages. Comparison between languages. Implicit and explicit parallelism in non-imperative models. Multiparadigm languages. Elements of analysis and design of programming languages.

## Literature

* Andrews, G.R. y Scheneider, F.B.: “Concepts and Notations for Concurrent programming”, Computing Surveys, Vol. 15, No. 1, March 1983, pp. 340.
* Ben-Ari, M.: Principles of Concurrent Programming, Second Edition. John & Sons. 1987.
* Ghezzi, C. y Jazayeri, M.: Programming Language Concepts, Second Edition, John Wiley & Sons. 1987.
* Ledgard, H.F. y Marcotty, M.: “A Genealogy of Control Structures”, CACM, Vol. 18, No. 11, November 1975, pp. 629-639.

## Required to Complete

Three exams and three projects.

## Validity

From September 1989.

# CI-3715 Software Engineering

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 5 | 10 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 3 | 1 | 3 | 84 |

## Objectives

Introduce the student to the problematic and medium-sized software development techniques representative of the disciplined approach associated with Software Engineering. Design and systems validation techniques will be emphasized, especially the ones that:

* Increase the productivity of the software developer;
* Allow to control the complexity inherent in medium-sized systems;
* Introduce the student to team development.

## Topics

* Introduction to software engineering: The development of medium-sized software, the need for disciplined teamwork, stages in the software life cycle, critical factors, definition and importance of SI. Object-oriented software.
* Object-oriented modeling: Need for multiple models. Introduction to UML notation. UML history. UML diagrams.
* The problem statement, functional requirements and attributes of the software. The use case model: actors, use cases, scenarios, description of use cases, relationships between use cases, use case diagram.
* Class Diagrams: objects, classes, attributes, qualifiers, methods, aggregation, composition, inheritance and associations. Class Translation to Relational Scheme. Class Translation to Code.
* Interaction diagrams: Sequence diagrams of system events and Communication diagrams. Contracts. Translation of Communication Diagrams to Code.
* Layer architecture: presentation, application, storage. Design principles: expertise, coupling, and cohesion. The use of design patterns as an observer (model-view-controller, publish-subscribe), iterator, strategy, proxy, decorator, abstract and composite factory.
* Informal tests: Definition. Psychological barriers. Planning and monitoring of the testing process based on the registration of faults and defects. Unit tests. Integration testing. System tests

## Literature

* “Object-Oriented Software Engineering: Using UML, Patterns, and Java”. Bernd Bruegge, and Allen H. Dutoit. Prentice-Hall. 3rd Edition.
* “Applying UML and Patterns: An Introduction to Object-Oriented, Analysis and Design and Iterative Development”. Craig Larman. Prentice-Hall. 3rd Edition.

## Required to Complete

Exams, Quizzes, Homeworks, Presentations and Medium-Sized Development Project.

## Validity

From 2011.

# CI-3725 Translators and Interpreters

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 5 | 10 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 3 | 1 | 3 | 84 |

## Objectives

### Generals

* Introduction to the theoretical principles that underpin the construction of translators and interpreters.
* Study of regular languages and their properties.
* Study of context-free languages and their properties.
* Construction of simple Top-Down and Bottom-Up parsers.
* Introduction to translation schemes directed by syntax.

### Specifics

* Know the Chomsky hierarchy and its implications in language recognition.
* Have familiarity with grammars and automatons, and know their importance in the definition of languages and construction of parsers.
* Know how to transform a specification of a language into another equivalent language, within the framework of the equivalence seen in the course.
* Have familiarity with the concept of translation directed by syntax and its importance in the construction of translators and interpreters.

## Topics

* Introduction.
  + Languages and virtual machines.
* Regular languages.
  + Regular expressions.
  + Right linear grammar.
  + Finite automata.
  + Equivalence results.
  + Properties of regular languages.
  + Applications: Lexicographic analysis, pattern matching.
* Context-free languages.
  + Derivation tree (concrete syntax tree).
  + Elimination of left recursion.
  + LL Grammar (definition).
  + LL Grammar (1).
  + Construction of Top-Down parsers based on LL grammars (1).
  + LR Grammar (definition).
  + LR Grammar (0).
  + Construction of Bottom-up parsers based on LR grammars (0).
* Translation directed by syntax.
  + Introduction to translation directed by syntax.
  + Construction of the abstract tree.

## Literature

* Backhouse. Syntax of Programming Languages, Theory, and Practice.
* Aho Ullman. The Theory of Parsing Translation and Compiling. Vol. I. 1972.

## Required to Complete

Three Exams and a Simple Parser for an example language created by the professor, in one of the allowed languages.

## Validity

From September 1995 (Revised on February 2003).

# CI-3815 Computer Organization

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 5 | 10 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 2 | 0 | 72 |

## Objectives

### Generals

The main objective of the course is for the student to understand the concepts of real machines and virtual machines. In addition, to obtain basic knowledge about the key components of a computer, that is, the central processing unit, memory hierarchy, I/O devices and interconnection mechanisms.

### Specifics

* Know the basic components of a computer: CPU, memory and peripheral devices.
* Introduce concepts related to the basic components of modern architectures, such as pipelines and multiprocessors.
* Understand the relationship between the concept of a machine and the language that it interprets.
* Know the type of instructions offered at the machine level.
* Study different design criteria of conventional machine level instruction format.
* Study format of representation in memory of integers, characters, and reals.
* Understand the difference between the level of an assembly language and the level of a conventional machine language, and also know how to translate written programs into assembly languages and into machine languages.
* Understand how the control operation of peripheral devices is carried out.

## Topics

* The basic structure of a computer.
* Hierarchy of memories and I/O devices.
* Set and format of instructions.
* Numerical and character systems. The binary representation of integers in two’s complement, octal and hexadecimal. Representation of characters and numbers in floating-point.
* Management of data structures in an assembly language.
* Subroutines Conventions and parameter passing.
* Mechanisms of interruption handling.

## Literature

* William Stallings. Computer Organization and Architecture: Designing for performance. Prentice-Hall.
* J. Hennessy and D. Patterson. Computer Organization and Desing. The Hardware and Software Interface. Morgan Kaufmann.
* Tanenbaum S. Andrew. Structured Computer Organization. Tercera Edición. Prentice-Hall.

## Required to Complete

Two Exams and Two projects in an assembly language.

## Validity

Nonspecified.

# CI-3825 Operating Systems I

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 5 | 10 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 3 | 1 | 3 | 84 |

## Objectives

### Generals

Introduce the basic concepts of operating systems, with emphasis on multi-user systems.

### Specifics

* The behavior of a CPU through multiprogramming and concurrences with processes and threads. Policies and mechanisms for dispatching programs.
* Main memory behavior. Use of base and limit records. Management of virtual memory through paging and segmentation policies and mechanisms. Use of the Working Set.
* Secondary memory compartment. Use of logical and physical file systems. Name systems (directory). The basic scheme of File Protection.
* Managing Interlocking and Starvation.

## Topics

* Topic 1. Introduction: Structures of a computer system. Definition of operating systems. Historical perspective of operating systems. Components of an operating system.
* Topic 2. Structure of the Operating Systems: Basic structure. I/O structure. Structure of Direct Memory Access (DMA). Protection needs. System calls.
* Topic 3. Processes: Process definition. Process models. Context changes. Operations on processes. Cooperating processes. Threads.
* Topic 4. Process planning (scheduling): Scheduling queues. Scheduling levels. Scheduling structures. Scheduling algorithms.
* Topic 5. Process coordination: Cooperating processes. Relationship between processes. Concurrent processes. Threads. Race conditions, mutual exclusion, and critical sections. Mechanisms of mutual exclusion and synchronization.
* Topic 6. Interlock (deadlock): System model. Deadlocks characterization. Deadlock prevention methods. Methods to avoid deadlock. Deadlock detection method.
* Topic 7. Main memory: Virtual address space and physical address space. Multiple fixed partitions. Multiple partitions of variable size. Pagination. Segmentation.
* Topic 8. Virtual Memory: Pagination. Segmentation. Page replacement. Thrashing. Working set.
* Topic 9. File systems: Files. Directories. Access protection. File implementation. Directory implementation. Efficiency and performance in file systems. Reliability in file systems.
* Topic 10. Secondary memory management: Disk structure, organization, and addresses. Free space administration. Disk planning.
* Topic 11. Design of operating systems: monolithic, microkernel, layered, virtual machines, object-oriented, client-server. Definition and use of upcalls.

## Literature

* Silberschatz A. Galvin P:B: Operating System Concepts. 4th Edition. Addison-Wesley Publishing Company. 1994.
* Tanenbaum A. & “Modern Operating Systems”. Prentice-Hall. 1992.
* Tanenbaum A. & Woodhull. “Operating Systems: Desing and Implementation”. 2nd Edition. Prentice-Hall. 1998.
* Stalling W. “Operating Systems Internals and Desing Principles”. 3rd Edition. Prentice-Hall. 1998.

## Required to Complete

Two Exams, two projects in C and the third project in either C or Bash.

## Validity

Nonspecified (Revised on January 2000).

# CI-4325 User Interfaces

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 5 | 10 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 3 | 1 | 3 | 84 |

## Objectives

### General

Transmit to the student the most appropriate approach under which to perceive, design, create and evaluate the interfaces.

### Specifics

Allow the student to understand the area of human-machine interaction and see the discipline from the point of view of the technology of the I/O devices, the ergonomics of these devices, the design methods and development of interaction software, the psychology of the programming, the users of the application systems and the modeling and measurements of the human-machine interaction.

## Topics

* Introduction. The history of computer interfaces, texts, and fundamental authors: Vannevar Busch, J.C.R. Licklider, Doug Engerbart, Ted Nelson, Ivan Stherland, Alan Kay, Nicholas Negroponte, etc. Importance of Interfaces.
* Importance of interface design in the software life cycle.
* Presentation of the I/O devices, its ergonomics functionality.
* Knowledge of the three main channels of communication: the visual channel, the auditory channel and the gestural channel.
* Methods of design and creation of Interfaces.
* Methods of evaluation of Interfaces.
* Presentation of some commercial interface systems: Windows, Macintosh toolbox, Mac App, etc.
* Design and creation of an interface. For example a system of interaction with icons, a system of design and management of menus. Formalization and realization of an interaction metaphor, etc.
* Multimedia systems: the concepts of hypertext and electronic book. The video on the screen, videos DVI, CDI and CDXA.D.

## Literature

* J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, T. Carey. “Human-Computer Interaction”, Addison-Wesley, 1994.
* B. Laurel, “The Art of Human-Computer Interface Desing”. Addison-Wesley, 1990.
* R. Baecker, W. Buxton (eds) “Readings in Human-Computer Interaction. A Multidisciplinary Approach”, Morgan-Kauffmann. 1987.
* R. Beacker, J. Grudin, W. Buxton & s. Greemberg (eds) “Readings in Human-Computer Interaction: toward the year 2000”. Second Edition. Kauffmann. 1997.

## Required to Complete

Exams, Quizzes, Presentations, Homeworks and to develop a User Interface of a system.

## Validity

From 1995.

# CI-4835 Computer Networks I

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 5 | 10 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 3 | 1 | 3 | 84 |

## Objective

At the end of the course, students will have a basic knowledge about the notion of communication protocols, the software structure/hardware of local networks, the main functions of the software layers in the TCP / IP model and some network programming models, especially the client-server model.

## Topics

* Topic 1: Basic concepts. Internet. Local networks, wide area, etc. Protocols of use and implementation of services. Network software structuring by layers. TCP / IP and OSI models. Question-answer based protocols. Network services.
* Topic 2: Introduction to the Internet (or IP networks). Machine identifiers: “physical” address and IP address. IP address space: domains and machines. Nemonic names: DNS service. Association of the IP number with the DNS name.
* Topic 3: Introduction to the transport layer. Networking programming interface. Sockets: pair of pairs (IP address, port).
* Topic 4: Standard protocol case studies on transport layers, such as RFC822, POP, HTTP.
* Topic 5: Other network programming models. RPC.
* Topic 6: Components of an IP network. Local networks. Broadcast networks, point-to-point networks. Elements to build a local network. Transmission media and properties. Devices: modems, switches, hubs.
* Topic 7: Control and management of data exchange over a link. Frames and frame synchronization. Flow control at the link level. Reliable transmission. Error control. Error detection.
* Topic 8: IP in detail. Package format. Switching, fragmentation, and reassembly. Introduction to routing. Algorithms of distance vector and link state.
* Topic 9: UDP and TCP. Segment format. Protocols Induction to flow control and congestion control.
* Topic 10: Induction of safety aspects. Encryption. Authentication.

## Literature

* Petterson, L. Y Davie, B. “Computer Networks: A Systems Approach”, Morgan Kaufmann Publishers, Inc, 1996.
* Tanembaum, A. “Computer Networks”, Third Edition, Prentice-Hall, 1996.

## Required to Complete

Exams and two projects.

## Validity

Nonspecified.

# CI-5437 Artificial Intelligence I

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 4 | 8 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 2 | 0 | 72 |

## Objective

A consistent, advanced and updated introduction to Artificial Intelligence with emphasis on the fundamentals and use of AI techniques for modeling and problem-solving.

## Topics

* Search Problems; State Models:
  + Blind Search (Search in Depth, Search in Amplitude, Iterative Deepening).
  + Heuristic Search: Hill-Climbing, A \*, IDA \*, Branch & Bound.
  + Properties: Completeness, Optimality, Complexity in Time and Space.
  + Suboptimal algorithms: WA \*, WIDA \*, LDS,…
  + Extensions: Transposition tables, Pattern Databases.
  + Origin and Design of Heuristic Functions; Heuristics as Functions.
  + Optimal Cost of Simplified Problems.
  + Applications: Games (Sokoban, 15-puzzle, Rubik); TSP,…
* Planning:
  + Planning as general problem-solving.
  + Representation languages to model actions and objectives: Strips, ADL, Functional Strips.
  + Planning Algorithms: Linear and non-linear planning, planning as a heuristic search, Graphplan, planning as ‘model checking’.
* Representation of Knowledge and Logic:
  + Propositional Logic: Language, Semantics and Test Theory. Normal Forms and Normal Conjunctive Form: Clauses.
  + SAT: The Problem of Satisfactability of a set of Clauses; Algorithms: Davis & Putnam, GSAT.
  + Logic for problem-solving: Planning as SAT, Scheduling as SAT, Diagnosis as SAT.
  + First Order Logic: Language and Semantics.
  + Horn clauses; Resolution.
  + Logic Programming.
  + Logic Control Representation: Golog, Tlplan.

## Literature

* Teacher notes.
* Papers delivered in class.
* Reference (but not text): Russell and Norvig; Artificial Intelligence: A Modern Approach, Prentice-Hall, 1995.
* Links on the course page.

## Required to Complete

One final exam, three projects about Search, one Game, SAT.

## Validity

From January 1998.

# CI-5438 Artificial Intelligence II

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 4 | 8 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 2 | 0 | 72 |

## Objective

Extension of the fundamental ideas in Artificial Intelligence 1 (CI-5437) to deal with problems involving uncertainty and partial observability.

## Topics

* Simple Decision Problems:
  + Probability Theory; Semantics and Axioms.
  + Bayes Rule: Intuition and Use.
  + Bayesian Networks: Representation and Inference Algorithms.
  + Theory of Utilities; Axioms.
  + Decision Trees.
  + Influence Diagrams: Representation and Algorithms.
* Sequential Decision Problems with Full Observability:
  + Markov chains.
  + Markovian Decision Processes (MDPs).
  + Algorithms: Dynamic programming (value iteration and policy iteration).
  + The Dimensionality Problem.
  + Dynamic Real-Time Programming (RTDP).
  + Resolution of MDPs by trial and error methods: reinforcement learning.
* Sequential Decision Problems with Observability with Partial Observability:
  + Partially Observable Markovian Decision Processes (POMDPs).
  + Dynamics of belief states, sensor modeling.
  + Problem Modeling in the framework of POMDPs.
  + POMDPs Resolution: RTDP.
* Approximate Dynamic Programming:
  + Linear approximation of the value function.
  + Non-linear approximations of the value function: Neural Networks.
* High-Level Languages for Modeling MDPs and POMDPs.
* Applications:
  + Backgammon, Manufacturing Processes, Elevator Control,…

## Literature

* Teacher notes.
* Papers delivered in class.
* Pearl; Probabilistic Reasoning in Intelligent Systems, Morgan Kaufmann, 1989.
* Bertsekas and Tsitsiklis, Neuro-Dynamic Programming, Athena, 1996.
* Barto and Sutton: Introduction to Reinforcement Learning, MIT Press, 1997.
* Links on the course page.

## Required to Complete

Three Projects.

## Validity

From January 1998.

# CO-3121 Fundamentals of Probability for Engineers

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 3 | 6 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 2 | 0 | 72 |

## Objectives

* The purpose of this course is to provide the student with the basic knowledge of probabilities, discrete and continuous random variables, as well as the applications of the Large Number Theorems.
* Prepare students for further specialized studies, both in a mathematical discipline and in any of the sciences that require good probabilistic foundations.

## Topics

* Motivation. Probability models. Operations with Sets. Finite sample spaces. Properties of the probabilities and principle of Inclusion-Exclusion.
* Counting. Conditional Probability. Independent Events. Total Probability Formula. Bayes theorem. Continuity of the Probabilities.
* Random variables. Distribution of a random variable. Random discrete variables. Main examples.
* Poisson distribution. Continuous Distributions. Main examples. Cumulative distribution function and properties. Mixed distributions.
* One-dimensional variable changes. Joint Distributions. Joint probability function. Joint density.
* Marginal and conditional distributions. Independence of random variables.
* Change of multidimensional variables. Convolution of distributions.
* Expected value and variance. Properties, interpretation, and applications.
* Covariance and correlation. Properties and interpretation. Probability generating function. Applications.
* Convergence in probability. Weak law of large numbers. Central Limit Theorem. Applications.

## Literature

The sections refer to the text of Evans and Rosenthal, “Probability and Statistics, The Science of Uncertainty” (Reverté, 2005). Support texts: First we recommend the Walpole, Myers, Ye, “Probability and Statistics for Engineering and Science”, 8th edition (Pearson). Also, the following two are very close to our program: Wackerly, Mendenhall and Scheaffer, “Probability and Mathematical Statistics”, 6th edition. (Thomson) and Meyer, P L. “Probability and Statistical Applications”. Inter-American Educational Fund, 1973.

## Required to Complete

Three exams, Weekly Laboratories and Projects.

## Validity

From October 2007.

# CO-3211 Numerical Analysis for Engineers

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 4 | 8 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 3 | 0 | 84 |

## Topics

* The problem of finite precision. Absolute and relative error. Significant digits. Cancellation. Propagation of errors.
* Matrix Algebra Review. Positive orthogonal and Positive-definite matrices; vector and matrix norms; Cauchy-Schwarz inequality; eigenvalues and eigenvectors.
* Matrix factorizations. LU factorization with and without pivoting. Cholesky factorization in the positive symmetric and definite case. QR factorization using Householder transforms. Operations count. Special cases for matrices with special structures.
* Linear systems solution. Direct methods based on the previous factorizations. Conditioning. Special case: linear minimum squares. Iterative stationary methods (briefly), examples: Jacobi, Gauss-Seidel, SOR. Iterative non-stationary methods in the symmetric and definite positive case: The Cauchy method, and the conjugate gradient method. Convergence properties.
* Interpolation techniques. Interpolation polynomials: Lagrange polynomial, Newton’s formula (divided differences), Hermite interpolation, error analysis, Chebyshev nodes. Interpolation Spline. Linear and natural cubic spline.

## Literature

* K. Atkinson. An Introduction to Numerical Analysis, 2nd Ed. John Wiley, N.Y., 1989. [QA297 A84 1989]
* D. Kincaid y W. Cheney. Numerical Analysis: Mathematics of Scientific Computing, 3rd Ed. Brooks/Cole, 2002. [QA297 K5635]
* M.T. Heath. Scientific Computing, 2nd Ed. McGraw-Hill, 2002. [Q183.9 H4 2002]
* N. J. Higham. Accuracy and Stability of Numerical Algorithms. SIAM, 1996. [QA297 H53]

## Required to Complete

Three exams and Laboratory Projects.

## Validity

From July 2012.

# CO-3221 Statistics for Engineers

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 4 | 8 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 1 | 1 | 72 |

## Objectives

### Generals

* This subject is intended to provide the student with the knowledge about basic tools developed in the field of Statistics and its application in practical problems.
* Simple statistical models will be used to motivate the development of some fundamental aspects of inference and hypothesis testing. Theoretical developments will be motivated by applications and models.

### Specifics

* Learn to build graphs and statistical measurements to treat data sets and obtain from them visual and numerical information about the phenomenon or problem of interest.
* Know and apply the basic statistical tools to measure, compare and contrast definite assumptions about the phenomenon or problem of interest.
* Interpret the information provided by the results of the measurements, comparisons or contrasts obtained through basic statistical tools, to make decisions that involve the phenomenon or problem of interest.
* Write reports that present in a clear and structured way the information, results, and interpretation obtained through the use of basic statistical tools.

## Topics

* Objectives of Statistics. Descriptive Statistics: Mean and sample variance, median and percentiles (quantiles), histograms, box diagrams. Estimation: Population parameters and estimators. Examples Goodness of an estimator: Bias, variance, and EMC. Common point estimators and their properties. Confidence (probabilistic) of an estimator. Duration 1 week.
* Confidence intervals. Pivot Method. Confidence intervals for the mean, difference of means and variance in the normal case. Normal approximation for large samples. Relationship with sample size. Duration 1 week.
* Relative efficiency of estimators. Consistency. Method of the moments. An Estimate by the Maximum Likelihood Method. Examples and asymptotic properties. Duration 1 week.
* Hypothesis tests. Test statistic, rejection region, Type I and Type II errors. Examples. Normal approximation for large samples. Probability of type II errors and sample size. Relationship between hypothesis tests and confidence intervals. p-values. Hypothesis tests for the mean and difference of means for normal data. Hypothesis tests for the variance of normal data. Neyman-Pearson lemma and likelihood ratio tests. Resampling and a non-parametric test for the difference of means. Duration 2½ weeks.
* Linear models. Adjustment for least-squares in simple linear regression. Inference regarding ith-Beta’s. Inference regarding linear combinations of the ith-Betas. Prediction of Y(x). Correlation. Multiple linear regression. Adjustment and inferences about the parameters and their linear combinations. Prediction of Y(x) in multiple linear regression. Hypothesis tests on the parameters. Resampling and non-parametric inference about ith-Beta’s in simple linear regression. Duration 2½ weeks.
* Analysis of Variance: Motivation and procedure in the case of a factor (comparison of two means or more than two means). ANOVA table. Statistical model, additivity of the sums of squares and estimation in the design of a factor. Duration 1 week.
* Non-parametric methods for paired samples: Sign test and sign ranges test. Wilcoxon and Mann-Whitney statistics for the problem of two unpaired samples. Duration 2 weeks.

## Literature

* Pérez, María Eglée: “Notes for the CO-3311 course”.
* Wackerly, Mendenhall and Scheaffer, Mathematical Statistics with applications.
* Peña, Daniel: “Statistics, Models and Methods, Vol. I and II”, Editorial Alliance, Madrid.
* Montgomery and Runger: “Probability and Statistics applied to Engineering”, McGraw-Hill.
* Walpole, Myers, Myers: “Probability and Statistics for Engineers”. Prentice-Hall.

## Required to Complete

Two exams and two Projects.

## Validity

From January 2007.

# CO-6612 Introduction to Artificial Neural Networks and their applications

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 4 | 8 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 1 | 0 | 60 |

## Objectives

### Generals

It is desired that the student knows alternative techniques to conventional statistics such as neural networks and is able to apply them in real situations where statistical techniques could have important limitations.

The course is designed for the student to master the concepts and techniques associated with the various neuronal paradigms that are studied, as well as to be able to use software for their application or implement their own routines and use them. In this way, the theoretical knowledge obtained is strengthened by the immediate application thereof to real problems of interest.

The course emphasizes the various configurations of neural networks that can be obtained and the various training algorithms that can be used, according to the problem being attempted.

### Specifics

* Understand and apply concepts and techniques of neural networks, appropriate selection of training and validation sets for them, detect patterns and make approaches using neural paradigms.
* Write software for the application of the techniques studied to data sets in different areas, and interpret the results obtained within the framework of the problem being analyzed.

## Topics

* Introduction to MATLAB. Construction and manipulation of vectors, basic arithmetic, logical variables., Data reading, data structures: matrices, arrays, datasheets, lists. Graphics. (1 week)
* Introduction. Motivation. The analogy between artificial and biological neural networks. (1 week)
* Single-layer networks. Optimization techniques as learning algorithms. Linear filters of least squares. Algorithms Training Curves. Perceptron convergence theorem. Limitations. (3 weeks)
* Multilayer networks. Backpropagation algorithm. Exit representation and decision rules. Backpropagation and differentiation. Cross-validation. Network simplification techniques. The convergence of the backpropagation algorithm. Improvements and limitations to the backpropagation algorithm. Examples: feature detection, approximation, XOR, numerical experiments. (3 weeks)
* Functional networks of radial bases. Cover theorem. Theory of regularization, regularized networks. Estimation of regularization parameters. Comparison with multilayer networks. Learning strategies. Examples: interpolation problems, XOR, numerical experiments. (1 week)
* Unsupervised networks. Motivation and definition. Kohonen unsupervised networks. Training Algorithm. Map of self-organized features. Examples (two weeks)

## Literature

* Simon Haykin, “Neural Networks, a comprehensive foundation”, second edition. Prentice-Hall. 1999.
* MATLAB Neural Networks toolbox user manual, (available online), <http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/>.
* Sergios Theodoridis and Konstantinos Koutroumbas, “Pattern Recognition”, second edition. Elsevier Academic Press. 2003.
* Christopher Bishop, “Neural Networks for pattern recognition”, Oxford Press, 1995.

## Required to Complete

Two exams and Projects for all the paradigms seen in class.

## Validity

From January 2005.

# EP-4793 Mini Software Development Project

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 8 | 16 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 3 | 0 | 12 | 180 |

## Objectives

### General

This course aims to develop competencies in students to develop a software project, with a part-time dedication, during a trimester. In the development of the mini software development project, the student will have the opportunity to carry out activities of the various phases of the software development cycle, in accordance with a work plan proposed by a USB teacher. The student will cover some of the phases of development, not necessarily all.

### Specifics

* Establish strategic planning of the processes related to the development of a software project.
* Demonstrate in the professional practice of the mini software development project, the fundamental aspects around the conceptual, procedural and attitudinal competencies that a student must possess at the level of training that the software development process contemplates.
* Systematically systematize the professional experience that constitutes the software development process with the fundamental guidelines established by the Teaching Coordination of the career.

## Topics

The Director of the Mini Project of Development of Software will be a professor of the USB who must write a plan of the work to be done, as complete as a long internship plan, but whose development occurs throughout the 12 weeks of a period of the USB.

The work plan must include a general objective (what is the product to be built), specific objectives (how the general objective is subdivided into smaller objectives in order to achieve the general one) and the plan of activities to be carried out to achieve the objectives. It must also be specified for how many students the Mini Software Development Project is defined and what are the responsibilities of each student if it is offered for more than one student. Likewise, it must be specified if the student must meet additional requirements to the requirements to pursue the Mini Software Development Project, for example, having passed a specialization chain or a particular subject may be additional requirements.

The Teaching Coordination of the career must approve the work plan of the Mini Software Development project before the student can take it. And also verify that the student meets the requirements to take it. The follow-up of the work will be in charge of the Director of the Mini Project of Development of Software and will be subject to the evaluation standards of the Mini Software Development projects determined by the Coordination, who is the unit in charge of planning, supervising, advising and monitor compliance with the Mini Software Development Project.

## Literature

According to the work plan and the area of application.

## Required to Complete

The Project must be evaluated by two Jurors selected by the Coordination.

## Validity

From April 2017.

# FS-1111 Physics I

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 3 | 6 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 3 | 2 | 1 | 72 |

## Topics

* Introduction to Physics: Physical quantities, measurements, units. (2 hours)
* Vectors: Graphical and analytical representation. Components of a vector. Sum of vectors. Product of a scalar by a vector. Scalar and vector product. (8 hours)
* Particle Kinematics: Vectors position, speed, and acceleration. 1-D movement; freefall. 2-D movement. Projectile movement. Circular motion. 3-D movement. Relative movement. Galileo transformations. (14 hours)
* Particle Dynamics: Inertial reference systems. Newton’s laws. Amount of movement. Friction force. Dynamics of circular motion. Hooke’s law. (16 hours)
* Work and Energy: Definition of work. Work and kinetic energy. Conservative Forces. Potential energy. Energy conservation. Power. Impulses and collisions. (14 hours)
* Oscillatory Movement: Stable balance and oscillatory movement. Simple harmonic movement. Simple pendulum. Body at the end of a spring. Energy considerations of simple harmonic motion. Cushioned movement. (8 hours)

## Literature

* Sears, Zemansky, Young and Freedman. University Physics Vol. I. Pearson, Addison, Wesley.
* Resnick, Halliday, and Krane. Physical. Vol. I. Continental Publishing Company.
* Bauer and Westfall. Physics for Engineering and Sciences. Vol. I. Mc Graw Hill
* Tipler / Fly. Physics for Science and Technology. Vol I. Editorial Reverté.
* Serway and Jewett. Physics for Sciences and Engineering. Vol. I. Thomson.

## Required to Complete

Three Exams.

## Validity

From January 1991.

# FS-1112 Physics II

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 3 | 6 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 3 | 2 | 1 | 72 |

## Topics

* Rotation Motion: Kinematics of rotation; rigid body definition. Vectors. Angular velocity and angular acceleration. (8 hours)
* Rotation Dynamics: Definition of torque and amount of angular motion of a particle. Particle systems, calculation of its center of mass and its amount of angular movement. The energy in the rotation motion. Moment of inertia. The combined movement of rotation and translation of a rigid body. (12 hours)
* Static Equilibrium of Rigid Bodies: Center of mass and center of gravity. Stability of balance. (5 hours)
* Movement Under Central Forces: Amount of angular movement and angular velocity. Central forces; gravitation as an example of a central force, universal gravitation law, Kepler laws, energy considerations of satellites' movement. (9 hours)
* Hydrostatic: Definition fluid pressure and density. Pascal principle and Archimedes principle. (6 hours)
* Heat Phenomenology: Temperature, thermal equilibrium. Specific heat and heat. Heat and work, the mechanical equivalent of heat. First Law of Thermodynamics. (8 hours)
* Kinetic Theory: Ideal gas. Kinetic theory of the ideal gas. Specific heat. Energy sharing. (6 hours)
* Second Law of Thermodynamics: Reversible and irreversible processes. Carnot cycle. The second law of thermodynamics. Entropy (8 hours)

## Literature

* Sears, Zemansky, Young, Freedman, University Physics, Vol. I. Pearson Addison Wesley.
* Resnick, Halliday, and Krane. Physical. Vol. I. Continental Publishing Company.
* Serway and Jewett. Physics for Sciences and Engineering. Vol. I. Thomson.
* Bauer and Westfall. Physics for Engineering and Sciences. Vol. I. Mc Graw Hill.
* Tipler / Mosca. Physics for Science and Technology. Vol. I. Editorial Reverté.

## Required to Complete

Three Exams.

## Validity

From January 1991.

# MA-1111 Mathematics I

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 4 | 8 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 2 | 0 | 72 |

## Topics

* Real number properties. Logic. Inequalities.
* Absolute Value. Inequalities with Absolute Value.
* Coordinate system. Equation of the circumference and the line. Parallel and perpendicular lines.
* Functions. Domain and Rank. Absolute Value Function, Integer Part Function. Operations with functions: sum, differences, product, quotient, and composition. Translations.
* The natural and general exponential function. Injective function. Inverse function.
* Definition of the natural and general logarithm as the inverse of the corresponding exponentials.
* Trigonometric functions and their inverse. Hyperbolic functions and their inverse. Hyperbolic Identities.
* Limit theorems. Limits involving trigonometric, exponential, logarithmic and hyperbolic functions where the rule of L’Hôpital is not necessary
* Limits to infinity. Vertical asymptote, horizontal and oblique.
* Continuity of functions. Types of discontinuities. Continuity in an interval. The theorem of the intermediate value.
* Straight tangent to the graph of a function. Instant speed. Derivatives. Derivability implies continuity.
* Referral rules. Derivatives of rational polynomial functions, trigonometric, exponential, etc.
* Chain rule. Derivatives of a higher order. Implicit derivation.
* The derivative of inverse functions. The derivative of logarithmic and inverse trigonometric functions. Logarithmic derivation.
* Derivatives of hyperbolic functions and their inverse.
* Rolle’s Theorem. The theorem of the mean value for derivatives. Their geometric interpretations and their applications. Bisection method.
* Undetermined forms of type 0/0. L’Hôpital rule for such indeterminate form.
* L’Hôpital for Infinite / Infinite. Other indeterminate forms.

## Literature

* Purcel, Varbeg and Rigdon. Calculus. Editorial Prentice-Hall. Ninth edition (2007).
* T. Apostol Calculus Vol. 1 Editorial Reverté. Second Edition (2006)
* J. Stewart Calculus of a single variable.
* Spivack. Calculus. Editorial Reverté.

## Required to Complete

Three Exams.

## Validity

From September 2008.

# MA-1112 Mathematics II

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 4 | 8 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 2 | 0 | 72 |

## Topics

* Antiderivatives. Indefinite Integrals (including trigonometric functions and their inverse).
* Sum and sigma notation. Introduction to areas. Definite integral.
* Properties of the definite integral (including the additive property of intervals). The first fundamental theorem of calculus.
* The second fundamental theorem of calculus. The theorem of the mean value for integrals.
* Evaluation of definite integrals. Rule of substitution (theorem A and B). Areas.
* Integration by substitution. Trigonometric integrals.
* Natural logarithmic function. Properties. Logarithmic derivative.
* Natural exponential function. General logarithmic and exponential functions.
* Rationalizing substitutions. Integration by parts.
* Integration by parts.
* Hyperbolic functions and their inverse.
* Integration of rational functions.
* Review of the L’Hopital rule. Other indeterminate forms for limits. Improper integrals (infinite limits integration).
* Improper Integrals. (Infinite integrands).
* Volumes of revolution. (Discs and Washers).
* Volumes of revolution. (Shells).

## Literature

* Purcell, Varberg, and Rigdon. Calculus. Editorial Prentice Hall, 8th Edition.
* Louis Leithold. Calculus. Publisher: Oxford University Press, 7th Edition.
* James Stewart. Calculus of a Single Variable. 4th Edition.

## Required to Complete

Three Exams.

## Validity

From January 2005.

# MA-1116 Mathematics III

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 4 | 8 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 2 | 0 | 72 |

## Topics

* Matrices. Operations with matrices. Examples.
* Systems of m equations with n variables. Elementary row operations. Stepped matrix, stepped reduced. Gauss and Gauss-Jordan methods.
* Systems with one solution, with infinite solutions and inconsistent; Homogeneous and non-homogeneous.
* Matrices and linear equations systems. Identity matrix. Invertible Matrix. Calculation of the inverse of a matrix. Equivalent matrices by rows.
* Transposed matrix, symmetric matrix. Determinants.
* Determinants properties.
* The determinant of the Inverse. Adjoint matrix. Calculating the inverse using the adjoint.
* Vectors in the plane and in space. Scalar product and projections.
* Vector product. Lines and planes in space.
* Real vector spaces. Subspaces.
* Linear combination and generated space. Linear independence.
* Base and dimension. Range, nullity. Row space and column space.
* Inner product spaces on real vector space.
* Orthogonal projection. Orthogonal sets. Gram-Schmidt process.
* Linear transformations. Properties of Linear Transformations.
* Image and core. Matrix associated with a linear transformation. Give examples using the canonical bases of the Real Vector Space, the Polynomial Space and the Matrix Space.
* Eigenvalues and eigenvectors. Real and Complex.
* Similar matrices. Diagonalization. Symmetric matrices. Orthogonal diagonalization.

## Literature

* Grossman, Stanley. Linear algebra. 5th Edition (*). Ed. Mc Graw Hill 1999. (* the fourth edition can also be used)
* Fraleigh. Linear algebra. Ed. Addison-Wesley Iberoamericana, 1989.

## Required to Complete

Three Exams.

## Validity

From April 2006.

# MA-2112 Mathematics V

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 4 | 8 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 2 | 0 | 72 |

## Topics

* Functions of several variables. Graphics. Level sets. Vector fields.
* Open sets, closed sets, border of a set. Limits and continuity.
* Partial derivatives. Differentiability.
* Properties of the derivative. Chain rule.
* Gradient. Directional derivative. Tangent plane.
* Iterated partial derivatives. Implicit derivation.
* Taylor’s theorem of second order. Critical points.
* Classification of critical points.
* Conditioned extremes. Lagrange multipliers.
* Trajectories. Arc length. Path integral.
* Line Integral.
* Double integrals. Fubini’s theorem.
* Integration of elemental regions. Changing the order of integration.
* Triple integral.
* The geometry of the functions of R2 in R2. Change of variables theorem.
* Polar, cylindrical and spherical coordinates. Applications.
* Green’s theorem.

## Literature

* J. Marsden and A. Tromba: Vector Calculus. 4th edition. Addison-Wesley.
* T. Apostol: Calculus. Volume II, 2nd edition. Editorial Reverté.
* R. Morales Bueno. Differential Calculus and Integral Calculus Exercises in several Real Variables. 3rd. Edition. USB.

## Required to Complete

Two Exams.

## Validity

From April 2007.

# MA-2115 Mathematics IV

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 4 | 8 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 2 | 0 | 72 |

## Topics

* Infinite successions.
* Infinite series.
* Positive series. Integral test and other tests.
* Alternating series. Absolute convergence.
* Power series. Operations with Power series.
* Taylor and Mac Laurin Series. Error estimation.
* Ordinary differential equations. Introduction and Examples.
* Directional Fields. Integral Curves. Existence and uniqueness of the solution.
* First Order Linear Equations. Bernouilli equation.
* Equations in separable variables and homogeneous equations.
* Some cases of reduction of order.
* Systems of First Order Linear Equations. Existence and uniqueness of the solution.
* Resolution of homogeneous linear systems with constant coefficients.
* Reduction of non-homogeneous linear systems.
* General Theory of n-th Order Linear Differential Equations.
* Resolution of differential equations.
* Linear differential equations with variable coefficients.

## Literature

* Purcell, D. Varberg & SE Rigdon Calculation 8th. Edition. Prentice Hall.
* A. & J. Viola-Prioli. Ordinary differential equations. 3rd. Edition. Department of Mathematics, USB.

## Required to Complete

Two Exams.

## Validity

From September 2007.

# PS-1111 Linear Models

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 4 | 8 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 3 | 1 | 96 |

## Objectives

### General

Introduce the student to the art of modeling, the use of models and their resolution for management and that serve as support in decision making.

### Specific

Provide the student with optimization techniques and their fundamentals, in order to acquire the necessary knowledge to optimize some management processes such as linear programming problems, physical distribution and project control (PERT-CPM).

## Topics

* Introduction:
  + Introduction to linear programming.
  + Models of linear programming.
  + Mathematical formulation.
* The Problem of Linear Programming:
  + Definition and properties of the linear programming problem.
  + Formulation of linear programming problems.
  + Applications of linear programming problems.
  + Graphic resolution of linear programming problems in two dimensions.
* Duality and Sensitivity Analysis:
  + Definition of the dual problem.
  + Formulation of the dual problem.
  + Interpretation of the dual variables.
  + Objectives of sensitivity analysis.
  + Graphical sensitivity analysis of linear programming problems.
  + Study of the different cases of sensitivity analysis.
  + Sensitivity analysis through computational packages.
* The Transportation Problem:
  + Definition of the transport problem.
  + Unbalanced transport models.
  + Transportation through computer packages.
  + Transshipment models.
  + Transshipment through computer packages.
  + Special transport cases.
* PERT-CPM:
  + Introduction to the PERT-CPM technique.
  + Diagramming for PERT-CPM.
  + Determination of the critical path.
  + Construction of the timing diagram.
  + Probabilistic considerations.
  + Consideration of costs.
  + PERT-CPM through computer packages.

## Literature

* Anderson, D. R .; Sweeney, D. J .; Williams, T. A. (2005). Quantitative methods for business. 9th Edition, Thomson.
* Eppen, G. D .; Gould, F. J .; Schmidt, C. P .; Moore, J. H .; Weatherford, L. R. (2000). Operations Research in Administrative Science. Pearson (Prentice-Hall).
* Hillier, F. S; Liberman, G. J. (2002). Operations research. 7th Edition, McGraw-Hill.
* Render, Quantitative methods for business. 9th Edition, Thomson.
* Taha, H. A. (2004). Operations research. 7th Edition, Pearson (Prentice-Hall).
* Winston, W. L. (2005). Operations research. Applications and Algorithms. 4th Edition, Thomson.

## Required to Complete

Two Exams and a Project through the course.

## Validity

From July 2006.

# PS-1115 Information Systems I

## Credits

|  |  |
| --- | --- |
| USB Credits | ECTS |
| 4 | 8 |

## Hours

|  |  |  |  |
| --- | --- | --- | --- |
| Theory Hours Per Week | Practice Hours Per Week | Laboratory Hours Per Week | Total Hours (12 week Trimester) |
| 4 | 3 | 1 | 96 |

## Objectives

### Generals

* Establish the theoretical basis that allows the deduction of a dynamic methodology for the development of information systems based on conceptual principles of the general theory of development systems and methodologies that are currently used.
* Have practical experience in the use of the knowledge acquired in theory by applying a work methodology in solving a real case.

### Specifics

* Identify the basic concepts of general systems theory.
* Understand information systems. Its categorization and its role within organizations.
* Establish differences between the different methodologies of information system development, (oriented to function, data, objects, etc.).
* Learn in detail the methodologies for developing systems most frequently used.
* Apply them conveniently in solving a problem or real-life case.

## Topics

* Chapter I: Introduction General Systems Theory (TGS). Approaches: Synergy.
* Chapter II: Recursivity. Systemic and systematic.
* Chapter III: Systems and concepts.
* Chapter IV: Organizations as systems.
* Chapter V: Systems management. Role of the Information Systems Analyst.
* Chapter VI: Classic life cycle.
* Chapter VII: Analyst Skills.
* Chapter VIII: Methodologies. Concepts.
* Chapter IX: Function-oriented methodology, instruments, and techniques.
* Chapter X: Methodology oriented to data, instruments, and techniques.
* Chapter XI: Object-oriented methodology, instruments, and techniques.
* Chapter XII: Types of Information Systems systems.
  + Data processing systems.
  + Management information systems.
  + Decision support systems.
  + Executive support systems.
  + Expert systems.
  + Office automation systems. Groupware.

## Literature

* Alter, S. Information systems. A management perspective. Second edition. The Benjamin/Cumming Publishing Company, 1996.
* Conger, S.; Valacich J. The new software engineering. Wadsworth Publishing Company, 1996.
* Donaldson D., Sandra. Systems Analysis and Design and the Transition to Objects. McGraw Hill Inc., 1996.
* Hoffer J., George J. Modern systems analysis and design. The Benjamin/Cumming Publishing Company, 1996.
* Kendall, E. y Kendall, J. Systems analysis and design. Third edition. Prentice-Hall, 1995.
* Mintzberg, H. y Brian, J. The strategy process concepts, contexts, cases. Third edition. Prentice-Hall, 1996.
* Norma R. Object-oriented systems analysis and design. Prentice-Hall, 1996.
* O'Brien, J. Introduction to information systems, Eighth Edition, Irwin/McGraw-Hill, 1997.
* Whitten, Jeffrey L. & Whitten Bentley, Lonnie D. Systems Analysis and Desing Methods. 4th Edition. Erwin/Mc Graw Hill, 1998.

## Required to Complete

Two Exams, Quizzes, Homework and a Project through the course.

## Validity

From May 2013.