



Teaching Guide				
Identifying Data				2020/21
Subject (*)	Language Processing	Code	614G01067	
Study programme	Grao en Enxeñaría Informática			
Descriptors				
Cycle	Period	Year	Type	Credits
Graduate	1st four-month period	Fourth	Optional	6
Language	SpanishGalician			
Teaching method	Hybrid			
Prerequisites				
Department	Ciencias da Computación e Tecnoloxías da InformaciónComputación			
Coordinador	Dafonte Vazquez, Jose Carlos	E-mail	carlos.dafonte@udc.es	
Lecturers	Arcay Varela, Bernardino Dafonte Vazquez, Jose Carlos Gomez Garcia, Angel Martinez Perez, Maria	E-mail	bernardino.arcay@udc.es carlos.dafonte@udc.es angel.gomez@udc.es maria.martinez@udc.es	
Web	http://estudios.udc.es/es/study/detail/614G01V01#plan-structure			
General description	<p>Compiladores; tradutores e intérpretes; etapas dun compilador; optimización de código; macroprocesadores.</p> <p>O obxectivo é familiarizar ó alumno co funcionamento dos reconecedores da linguaxe e os compiladores como un caso particular, o entorno no que traballan así coma algunhas ferramentas software para a construción dos mesmos. É preciso asumir a característica interdisciplinar da asignatura. Adquirir os coñecementos necesarios para deseñar e implementar as diferentes etapas necesarias para o desenvolvemento dun reconecedor da linguaxe: análise (léxico, sintáctico e semántico) e síntese (xeración de código intermedio, optimización de código e xeración de código obxeto).</p>			
Contingency plan	<ol style="list-style-type: none"> Modifications to the contents Methodologies <ul style="list-style-type: none"> *Teaching methodologies that are maintained *Teaching methodologies that are modified Mechanisms for personalized attention to students Modifications in the evaluation <ul style="list-style-type: none"> *Evaluation observations: Modifications to the bibliography or webgraphy 			

Study programme competences / results	
Code	Study programme competences / results
A40	Capacidade para coñecer os fundamentos teóricos das linguaxes de programación e as técnicas de procesamento léxico, sintáctico e semántico asociadas, e saber aplicalas para a creación, o deseño e o procesamento de linguaxes.
B1	Capacidade de resolución de problemas
B3	Capacidade de análise e síntese
C2	Dominar a expresión e a comprensión de forma oral e escrita dun idioma estranxeiro.
C4	Desenvolverse para o exercicio dunha cidadanía aberta, culta, crítica, comprometida, democrática e solidaria, capaz de analizar a realidade, diagnosticar problemas, formular e implantar solucións baseadas no coñecemento e orientadas ao ben común.
C6	Valorar criticamente o coñecemento, a tecnoloxía e a información dispoñible para resolver os problemas cos que deben enfrontarse.



Learning outcomes			
Learning outcomes	Study programme competences / results		
Coñecer os conceptos teóricos básicos nos que se basean os compiladores. Diseñar e implementar cada unha das fases precisas para a implementación dun compilador. Coñecer as distintas ferramentas dispoñibles para a implementación de compiladores e manexar algunhas das máis habituais. O estudante tamén saberá verificar o funcionamento do procesador da linguaxe. En relación ás competencias tipo C, será preciso manexar material en inglés e castelán e orientarse o traballo de prácticas para obter un diagnóstico e unha solución ós problemas asociados coa temática da asignatura.	A40	B3	C2 C4
Partindo da base teórica da computación, plantéxanse diferentes técnicas e ferramentas (cunha orientación moi práctica) para o procesamento da linguaxe. En relación ás competencias tipo C, tamén será necesario manexar documentación en ambos idiomas e trátase de valorar as diferentes solucións e plantexamentos posibles coas súas vantaxes e inconvenientes.	A40		C2 C6
Entender as razóns e saber como optimizar o código obxecto en función da plataforma que o vai a executar. Saber elixir o tipo de optimización segundo a máquina obxecto, as súas capacidades e a súa arquitectura. Con respecto ás competencias tipo C, unha vez máis precisaremos manexar documentación en varios idiomas e as prácticas tratarán de proporcionar as ferramentas para que o estudantes poda descubrir por sí mesmo as diferentes posibilidades e vantaxes según o contorno.		B1	C2 C6
Saber analizar e diagnosticar o caso concreto, o obxectivo, para o que se precisa un recoñecedor da linguaxe. O estudante tamén será capaz de documentarse, sintentizar e adaptar a solución cun enfoque totalmente práctico pero baseado no coñecemento. Sobre as competencias tipo C, e dado que principalmente este resultado obterase a partir do proxecto proposto polo estudante, é obvio que precisamos utilizar técnicas avanzadas baseadas no coñecemento, diagnóstico de problemas así como una valoración crítica da proposta que él mesmo realiza.		B3	C4 C6

Contents	
Topic	Sub-topic
Tema I. Introducción	1.1 Estructura dun compilador. 1.2 Exemplo das fases dun compilador.
Tema II. Linguaxes e Gramáticas	2.1 Notación e clasificación de Chomsky. 2.2 Gramáticas de contexto libre (GCL) e notación BNF. 2.3 Mellora e simplificación de gramáticas.
Tema III. Análise Léxica	3.1 Tipos de máquinas recoñecedoras ou autómatas. 3.2 Autómatas Finitos. 3.3 Conversión entre gramática regular, expresión regular e autómata finito. 3.4 Implementación de autómatas.
Tema IV. Análise Sintáctica (Parsing)	4.1 Máquinas teóricas, mecanismos con retroceso 4.2 Algoritmos sen retroceso. 4.2.1 Análise sintáctica ascendente por precedencia simple. 4.2.2 Analizadores descendentes LL(K). 4.2.3 Analizadores ascendentes LR(k). 4.2.4 Xenerador de analizadores sintácticos YACC.
Tema V. Análise Semántica	5.1 Definicións dirixidas pola sintáxis. 5.2 Esquema de tradución. 5.3 Comprobacións en tempo de compilación.
Tema VI. Xeneración de Código	6.1 Linguaxes intermedios. 6.2 Xeneración de código intermedio. 6.3 Xeneración de código dende linguaxe intermedio.



Tema VII. Optimización de Código	7.1 Algoritmo de Nakata. 7.2 Lazos nos grafos de fluxo. 7.3 Análise global do fluxo de datos. 7.4 Solución iterativa das ecuacións de fluxo de datos.
Tema VIII. Erros	8.1 Tipos de erros. 8.2 Recuperación de erros léxico-gráficos.
Tema IX. Intérpretes e Complementos	9.1 Estructura dun intérprete actual. 9.2 Arquitectura neutral de Java.

Planning				
Methodologies / tests	Competencies / Results	Teaching hours (in-person & virtual)	Student's personal work hours	Total hours
Guest lecture / keynote speech	A40 B3 C2 C4	21	31.5	52.5
Objective test	A40 B1 C2 C6	3	18	21
Supervised projects	B1 B3 C2 C4 C6	7	17.5	24.5
Laboratory practice	A40 B1 C2	14	14	28
Personalized attention		24	0	24

(*)The information in the planning table is for guidance only and does not take into account the heterogeneity of the students.

Methodologies	
Methodologies	Description
Guest lecture / keynote speech	Nestas sesións analizarase en profundidade os conceptos teóricos e técnicas principais para a implementación dun recoñedor da linguaxe
Objective test	Realizarase un exame ó final do cuatrimestre e a súa correspondente 2ª oportunidade. Este exame centrarase principalmente nos aspectos teóricos i exercicios asociados. Tamén poderase formular algunha cuestión relacionada coas prácticas de implementación realizadas.
Supervised projects	Levarase a cabo un proxecto a proposta de un grupo de 2 estudantes do que se fará un seguimento no horario de traballo en grupo.
Laboratory practice	Levaranse a cabo pequenas prácticas sobre os diferentes temas da asignatura que se irán explicando nas sesións maxistráis. Algunhas delas serán exercicios de desenvolvemento e outras de implementación.

Personalized attention	
Methodologies	Description
Guest lecture / keynote speech Laboratory practice Supervised projects	Especialmente no caso do proxecto a desenvolver, organizado por grupos de alumnos, realizarase un seguimento. A iso se adicarán principalmente as horas de traballo en grupo reducido.

Assessment			
Methodologies	Competencies / Results	Description	Qualification
Laboratory practice	A40 B1 C2	Levaranse a cabo pequenas prácticas sobre os diferentes temas da asignatura que se irán explicando nas sesións maxistráis. Algunhas delas serán exercicios de desenvolvemento e outras de implementación. A porcentaxe por asistencia i entrega destas pequenas prácticas é testimonial (10%). O 20% restante corresponde ó desenvolvemento do proxecto.	30
Objective test	A40 B1 C2 C6	Realizarase un exame ó final do cuatrimestre e a súa correspondente 2ª oportunidade. Este exame centrarase principalmente nos aspectos teóricos i exercicios asociados. Tamén poderase formular algunha cuestión relacionada coas prácticas de implementación realizadas.	50



Supervised projects	B1 B3 C2 C4 C6	Levarase a cabo un proxecto a proposta de un grupo de 2 estudantes do que se fará un seguimento no horario de traballo en grupo.	20
Others			

Assessment comments

En calquera caso, é preciso aprobar as dúas partes (proba obxectiva e prácticas-traballo). En caso contrario, a nota máxima que se poderá acadar é un 4.5.

IMPORTANTE: O 40% da nota apórtao o proxecto a propoñer polos grupos, o 20% corresponde ó seguimento e o 20% á entrega final.

ESTUDANTES CON MATRÍCULA A TEMPO PARCIAL OU CON DISPENSA ACADÉMICA DE EXENCIÓN DE DOCENCIA: Deberán poñerse en contacto cos profesores da asignatura para posibilitar a realización das tarefas fóra da organización habitual de materia.

Sources of information

Basic	
Complementary	"Compiladores: Principios, técnicas y herramientas", Aho, A.V.; Lam M.; Sethi, R. ; Ullman, J.D., Addison-Wesley, Reading, Massachussetts 2008."Construcción de compiladores. Principios y Práctica", Louden D. K., Paraninfo Thomson Learning, 2004.Garrido, A. ; Iñesta J.M. ; Moreno F. ; Pérez J.A. [2004] Diseño de compiladores, Publicaciones Universidad de Alicante."Compiladores, teoría y construcción", Sanchis, F.J.; Galán, J.A., Ed. Paraninfo, 1987."The theory of parsing, translation and compiling" (I y II), Aho, A.V.; Ullman, J.D., Prentice-Hall, 1972."Principles of compiler design", Aho, A.V.; Ullman J.D., Addison-Wesley, 1977."Introducción a la teoría de autómatas, lenguajes y computación", Hopcroft, J.E. ; Motwani R. ; Ullman, J. D. [2002] I, Addison-Wesley, 2002"Compiler design in C", Allen I.; Holub, Prentice-Hall, 1991."Compiladores e Intérpretes", Sánchez, G.; Valverde J.A., Ed. Díaz de Santos, 1984."Languages and machines?", Sudkamp T.A., Addison-Wesley, 1994

Recommendations

Subjects that it is recommended to have taken before

Theoretical Computer Science/614G01039

Subjects that are recommended to be taken simultaneously

Subjects that continue the syllabus

Programming Language Design/614G01065

Other comments

Aínda que ten relación con outras materias, está fortemente relacionada con "Deseño das Linguaxes de Programación", que a complementa, e "Teoría da Computación", que é a base inicial para poder afrontar os contidos.

(*The teaching guide is the document in which the URV publishes the information about all its courses. It is a public document and cannot be modified. Only in exceptional cases can it be revised by the competent agent or duly revised so that it is in line with current legislation.