

Laboratori de Programació

Codi: 102767
Crèdits: 6

Titulació	Tipus	Curs	Semestre
2502441 Enginyeria Informàtica	OB	2	1

La metodologia docent i l'avaluació proposades a la guia poden experimentar alguna modificació en funció de les restriccions a la presencialitat que imposin les autoritats sanitàries.

Professor/a de contacte

Nom: Gemma Sánchez Albaladejo
Correu electrònic: Gemma.Sanchez@uab.cat

Utilització d'idiomes a l'assignatura

Llengua vehicular majoritària: català (cat)
Grup íntegre en anglès: No
Grup íntegre en català: Sí
Grup íntegre en espanyol: Sí

Equip docent

Jorge Bernal del Nozal
Yael Tudela Barroso
Cristian Da Silva Campello
Estrella María Abril Gutiérrez

Prerequisits

L'assignatura no pot tenir cap prerequisit oficial per normativa. Però els estudiants que no han cursat i aprovat prèviament les assignatures de Fonaments d'informàtica i Metodologia de la Programació, suspenen en major proporció aquesta assignatura. Per això es recomana insistentment que l'estudiant hagi cursat i superat satisfactòriament les

assignatures prèvies de Fonaments d'Informàtica i Metodologia de la Programació així com Matemàtica Discreta. Per tant, està familiaritzat amb les estructures bàsiques i avançades de la programació, Orientació a objectes i el concepte de graf amb els diferents mètodes de recorregut sobre ells.

Objectius

Aquesta assignatura forma part de la matèria Algorísmica i informació i s'ha de veure com la continuació lògica de l'assignatura Metodologia de la Programació i la continuació pràctica de l'assignatura de Matemàtica Discreta. L'objectiu bàsic és aprofundir en les nocions de programació orientada a objectes introduïdes a Metodologia de la Programació i ampliar-les amb altres conceptes de programació i altres estructures de dades més complexes així com algorismes eficients per recorre-les. S'introduirà el concepte d'algorisme recursiu amb algorismes recursius simples i més complexos com els relacionats amb recorreguts d'arbres i grafs. A més s'introduiran algorismes de cerca i ordenació eficients i s'aprofundirà en el concepte de cost temporal i espacial d'un algorisme. Al finalitzar el curs l'estudiant ha de ser capaç de dissenyar i programar solucions a problemes complexos de forma òptima.

D'aquesta forma, els objectius formatius que es proposen per a l'assignatura són els següents:

- Ser capaç d'analitzar un problema complex, dissenyar una solució òptima, implementar-la, calcular el seu cost i provar-la.
- Entendre i saber utilitzar estructures de dades complexes com arbres, grafs etc. i utilitzar-les correctament i d'una manera eficient per resoldre problemes algorísmics complexes.
- Entendre i aplicar correctament els principis avançats de la programació orientada a objectes: templates, classes abstractes, funcions virtuals.
- Dotar l'alumne de la capacitat de disseny d'algorismes per a la resolució de problemes complexos, veient algorismes complexos de recorregut i cerca en estructures de dades complexes. A més d'analitzar la complexitat temporal i espacial d'ells per tal de triar la solució que més s'adapti a les necessitats de cada moment.
- Introduir el concepte de recursivitat i la seva aplicació al recorregut d'estructures complexes recursives, a més de ser capaç d'analitzar la complexitat d'algorismes recursius.
- Programar en un llenguatge de programació real i ser capaç de depurar els propis programes.
- Desenvolupar els programes seguint unes normes d'estil tendents a aconseguir programes de qualitat. Dins d'aquestes normes d'estil s'engloben aquelles que faciliten la comprensió del codi, com poden ser l'ús de comentaris, la indentació del codi, la utilització de noms adequats per variables i funcions, etc. I l'ús d'excepcions.

Competències

- Adquirir hàbits de pensament.
- Capacitat per concebre, desenvolupar i mantenir sistemes, serveis i aplicacions informàtiques emprant els mètodes de l'enginyeria del software com a instrument per a assegurar-ne la qualitat.
- Coneixement, disseny i utilització eficient dels tipus i estructures de dades més adequades a la resolució del problema.
- Conèixer i aplicar els procediments algorítmics bàsics de les tecnologies informàtiques per dissenyar solucions a problemes i per analitzar la idoneïtat i la complexitat dels algorismes proposats.

Resultats d'aprenentatge

1. Aplicar estratègies de depuració, prova i correcció de programes.
2. Conèixer i comprendre els paradigmes de programació.
3. Desenvolupar la capacitat d'anàlisi, síntesi i prospectiva.
4. Desenvolupar programes amb un bon estil de programació i documentant-los adequadament.
5. Identificar la complexitat computacional d'un algorisme en termes de recursos de memòria i temps d'execució.
6. Identificar possibles estratègies de solució a problemes mitjançant els conceptes propis del paradigma de programació orientada a objectes.
7. Seleccionar i aplicar la combinació d'estructures de dades i estratègies de resolució més apropiada per resoldre de manera eficient un problema informàtic.

Continguts

0. Introducció

Objectius i presentació de l'assignatura. Repàs de Programació Orientada a Objectes i Estructures de Dades Dinàmiques.

1. Orientació a objectes

Paradigma d'objectes avançat. Templates. Herència, classes abstractes, funcions virtuals i polimorfisme.

2. Estructures de dades no lineals. Hash

Tècniques de "Hashing". Matrius hash i llistes hash. Funcions hash.

3. Recursivitat i Algorismes d'ordenació

Introducció als algorismes recursius. Mètode de la bombolla, QuickSort, MergeSort. Recursivitat. Càlcul complexitat.

4. Estructures de dades no lineals. Grafs
Representacions i recorreguts. BFS, DFS, Resolució de problemes amb grafs.

5. Estructures de dades no lineals. Arbres.

Definició i representació d'un arbre. Recorreguts d'arbres. Binary Heaps. Red-Black Tree.

6. Conceptes bàsics de Python.

Conceptes bàsics de programació en Python.

Metodologia

NOTA INFORMATIVA DEL CURS DEGUT AL COVID: Aquesta assignatura es farà en mode on-line degut a la pandèmia, per tant totes les activitats que figuren com a presencials es faran virtuals mitjançant caronte i teams a no ser que les indicacions del govern i/o la UAB canviïn al llarg del curs.

La metodologia docent de l'assignatura parteix del principi que diu que "*programar és l'única forma d'aprendre a programar*" i, per tant, estarà centrada principalment en el treball pràctic de l'estudiant. També es basa en aprofitar al màxim el temps presencial que l'estudiant està amb el professor. D'aquesta manera conceptes teòrics descriptius, fàcilment assolits per l'estudiant mitjançant la visualització de vídeos o lectura d'articles, es faran de manera autònoma (i guiada) per part de l'estudiant. Mentre que la posada en pràctica d'aquests conceptes o l'ampliació dels mateixos es faran a classe amb ajuda del professor. L'objectiu principal de l'assignatura és que l'estudiant sàpiga resoldre un problema donat, de manera eficient, utilitzant estructures de dades complexes, si cal. Per aquesta raó l'aprenentatge es centrarà en acompanyar a l'estudiant en la seva tasca de resolució de problemes a partir d'uns conceptes teòrics estudiats prèviament de manera autònoma. Es farà servir principalment el llenguatge de programació C++, i es donaran alguns conceptes de programació en python.

La metodologia general de l'assignatura es pot dividir en tres fases:

Preparació de la classe: l'objectiu d'aquesta fase és que l'alumnat pugui aprendre els conceptes que es treballaran a la sessió següent mitjançant diverses activitats ofertes pel professorat com pot ser el visionat de vídeos, la lectura de textos, etc.

Classe presencial: l'objectiu d'aquesta fase és la de consolidar els conceptes vistos i posar-los en valor dins del context de l'assignatura. El professorat vetllarà perquè l'alumnat aprofundeixi en aquests conceptes mitjançant exercicis (més o menys) guiats durant la sessió, i afegint nous matisos als conceptes apresos de manera autònoma quan sigui necessari. D'aquesta manera la classe presencial s'aprofita molt més donat que els conceptes descriptius ja són coneguts per l'estudiant i es pot entrar de ple en la seva utilització i ampliació que són els punts en els quals l'estudiant pot necessitar més l'ajuda del professor.

Treball autònom: per tal que l'alumnat agafi desimboltura en la utilització d'estructures complexes i els algorismes associats a les mateixes aquest haurà de fer una part del treball pel seu compte, ja siguin exercicis solts o dins d'un projecte.

El treball del projecte s'haurà de fer en grups de 2 persones.

La gestió de la docència de l'assignatura es farà a través de Caronte (<http://caronte.uab.cat/>), que servirà per poder veure els materials, gestionar els grups de pràctiques, fer els lliuraments corresponents, veure les notes, comunicar-se amb els professors, etc.

La gestió de les classes presencials que degut a la pandèmia es facin on-line es farà a través de teams <https://teams.microsoft.com/>, qualsevol canvi al llarg del curs es notificarà a Caronte.

Competències Transversals

- T01.02 Desenvolupar la capacitat d'anàlisi, síntesi i prospectiva. Aquesta competència es desenvoluparà durant les sessions presencials, on l'estudiant, després d'haver consolidat uns conceptes teòrics, haurà d'analitzar un problema pràctic i proposar una solució òptima. Es tindrà en compte a l'avaluació de les diferents parts de l'assignatura.

Activitats formatives

Títol	Hores	ECTS	Resultats d'aprenentatge
Tipus: Dirigides			
Classes presencials	50	2	1, 2, 3, 4, 5, 6, 7
Tipus: Supervisades			
Tutories	1	0,04	1, 2, 3, 4, 5, 6, 7
Tipus: Autònomes			
Estudi Individual	13	0,52	2, 3, 5, 6
Preparació prèvia de les classes.	34	1,36	2, 3, 5, 6
Treball autònom	46	1,84	1, 3, 4, 5, 6, 7

Avaluació

L'avaluació de l'assignatura tindrà en compte tres tipus d'activitats d'avaluació: lliurament de problemes, avaluació individual i projecte de programació. La nota final de l'assignatura s'obté combinant l'avaluació d'aquestes 3 activitats de la manera següent:

$$\text{Nota Final} = (0.2 * \text{Avaluació Problemes}) + (0.4 * \text{Projecte}) + (0.4 * \text{Avaluació Individual})$$

- Lliurament de problemes: en aquest apartat s'inclou el lliurament dels exercicis que es proposin al llarg del curs i altres activitats que es realitzin a les sessions de problemes.

No és necessari aconseguir una nota mínima en aquesta activitat per poder aprovar l'assignatura.

Els exercicis que es lliurin fora de termini o que tinguin una avaluació de suspès es podran recuperar i tornar a lliurar en qualsevol moment del curs abans de la data de l'examen final de l'assignatura, amb una reducció sobre la nota del 20%. Els problemes estaran ponderats segons el pes del tema al conjunt de l'assignatura, i el nombre de problemes que s'hagin de lliurar per cada tema.

- Avaluació individual: en aquest apartat s'inclou el resultat de les proves individuals que es faran al llarg del curs. Hi haurà dues proves parcials que es faran durant el període lectiu del curs en horaris de classe i una prova final durant el període oficial d'exàmens. Aquesta prova final serà de recuperació i només l'hauran de fer els estudiants que no hagin superat algun dels dos parcials. Si s'ha superat un dels dos parcials, però l'altre no, en aquesta prova només s'ha de recuperar la part de l'assignatura corresponent al parcial que no s'hagi superat.

S'haurà d'aconseguir una nota mínima de 4 en cadascun dels dos parcials i una nota promig mínima de 5 per poder aprovar l'assignatura .

La nota final serà la mitja dels dos parcials:
$$\text{Avaluació Individual} = (0.5 * \text{Parcial1}) + (0.5 * \text{Parcial2})$$

- Projecte: inclou tot el treball del projecte de programació. Inclou l'avaluació dels dos lliuraments del projecte (un lliurament parcial a meitat de curs i el lliurament final) i l'avaluació del seguiment del

projecte que es farà a les sessions presencials que correspongui. La nota final es calcularà de la forma següent:

$$\text{Projecte} = (0.2 * \text{Avaluació seguiment projecte}) + (0.3 * \text{Entrega Parcial 1}) + (0.5 * \text{Entrega Final})$$

- S'haurà d'aconseguir una nota mínima de 4 en l'avaluació del seguiment del projecte i una nota mínima de 5 en el lliurament final del projecte per poder aprovar el projecte.
- S'haurà d'aconseguir una nota mínima de 5 en el projecte per poder aprovar l'assignatura.
- La nota de l'entrega final del projecte es podrà recuperar si la nota del projecte és ≥ 3 i la nota de l'avaluació individual és ≥ 5 .

No avaluable: Un alumne es considerarà no avaluable (NA) si no fa com a mínim el 50% dels lliuraments d'exercicis i no fa cap de les proves d'avaluació següents: parcial 1, parcial 2, prova final de recuperació, lliurament final de la pràctica.

Suspensos: Si el càlcul de la nota final és igual o superior a 5 però no s'arriba al mínim exigít en alguna de les activitats d'avaluació, la nota finalserà suspès i es posarà un 4.5 a la nota de l'expedient de l'alumne.

Convalidacions: Pels alumnes repetidors es convalidarà la nota del projecte de l'any anterior (curs 2017-18) si es compleixen aquestes condicions:

a) La nota final del projecte del curs anterior és més gran o igual a 7

b) La nota de l'avaluació individual del curs anterior és més gran o igual a 3

MH: Es donaran tantes matrícules com es puguin dins de la normativa de la universitat, començant per les notes més altes i sempre i quan la nota mínima sigui un 9.

Revisions: Per a cada activitat d'avaluació, s'indicarà un lloc, data i hora de revisió en la que l'estudiant podrà revisar l'activitat amb el professor. En aquest context, es podran fer reclamacions sobre la nota de l'activitat, que seran avaluades pel professorat responsable de l'assignatura. Si l'estudiant no es presenta a aquesta revisió, no es revisarà posteriorment aquesta activitat.

Nota important sobre còpies i plagis:

Sense perjudici d'altres mesures disciplinàries que s'estimin oportunes, i d'acord amb la normativa acadèmica vigent, les irregularitats comeses per un estudiant que puguin conduir a una variació de la qualificació es qualificaran amb un zero (0). Les activitats d'avaluació qualificades d'aquesta forma i per aquest procediment no seran recuperables. Si és necessari superar qualsevol d'aquestes activitats d'avaluació per aprovar l'assignatura, aquesta assignatura quedarà suspesa directament, sense oportunitat de recuperar-la en el mateix curs. Aquestes irregularitats inclouen, entre d'altres:

- la còpia total o parcial d'una pràctica, informe, o qualsevol altra activitat d'avaluació;
- deixar copiar;
- presentar un treball de grup no fet íntegrament pels membres del grup;
- presentar com a propis materials elaborats per un tercer, encara que siguin traduccions o adaptacions, i en general treballs amb elements no originals i exclusius de l'estudiant;
- tenir dispositius de comunicació (com telèfons mòbils, smart watches, etc.) accessibles durant les proves d'avaluació teòric-pràctiques individuals (exàmens).

En aquests casos, la nota numèrica de l'expedient serà el valor menor entre 3.0 i la mitjana ponderada de les notes (i per tant no serà possible l'aprovat per compensació).

En l'avaluació dels lliuraments de problemes i pràctiques s'utilitzaran eines de detecció de còpia del codi del programa.

Nota sobre la planificació de les activitats d'avaluació:

Les dates d'avaluació continuada i lliurament de treballs es publicaran al principi de curs i poden estar subjectes a canvis de programació per motius d'adaptació a possibles incidències. Sempre s'informarà a Cerbero sobre aquests canvis ja que s'entén que aquesta és la plataforma habitual d'intercanvi d'informació entre professors i estudiants.

Activitats d'avaluació

Títol	Pes	Hores	ECTS	Resultats d'aprenentatge
Examen de recuperació	veure la descripció del mètode d'avaluació	2	0,08	1, 2, 3, 4, 5, 6, 7
Lliurament problemes	20% Nota final	0	0	2, 3, 5, 6, 7
Primer parcial	50% Nota av. individual	2	0,08	1, 2, 3, 4, 5, 6, 7
Projecte Programació	40% Nota final	0	0	1, 3, 4, 6, 7
Segon parcial	50% Nota Av. individual	2	0,08	1, 2, 3, 4, 5, 6, 7

Bibliografia

- <http://www.cplusplus.com/> : The C++ Resources Network
- https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_C%2B%2B: Programación en C++ - Wikilibros
- Mark Allen Weiss. Data Structures and Data Analysis in C++. Pearson. 2014.
- B. Eckel. Thinking in C++, Volume 1: Introduction to Standard C++, Prentice-Hall, 1999
- B. Eckel. Thinking in C++, Volume 2: Standard Libraries and Advanced Topics, Prentice-Hall, 1999
- F. Xhafa, P. Vázquez, J. Marco, X. Molinero, A. Martín: Programación en C++ para ingenieros. Thomson, 2006
- Thinking in PYTHON Bruce Eckel (se puede descargar de <http://www.bruceeckel.com>).
- Learning PYTHON 2nd Edition. Mark Lutz and David Ascher, Safari Tech Books Online.
- Manuals de Python (de la pagina web oficial).
- Llibres electronics interactius de python:
 - <http://interactivepython.org/runestone/static/thinkcspy/toc.html#t-o-c>
 - <http://interactivepython.org/runestone/static/pythonds/index.html>
 - <http://www.pythontutor.com/>