

Laboratori de Programació

Codi: 102767
Crèdits: 6

Titulació	Tipus	Curs	Semestre
2502441 Enginyeria Informàtica	OB	2	1

Professor 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: No

Equip docent

Ernest Valveny Llobet
Gabriel Villalonga Pineda
Joaquin Cerda Company
Aura Hernández Sabaté

Prerequisits

L'assignatura no té cap prerequisit oficial. De totes formes, s'assumeix que l'estudiant ha cursat 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 Mètodes Abstractes de Càlcul. 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'aprofundirà en el concepte d'algorisme recursiu amb algorismes recursius complexos. A més s'introduiran algorismes de cerca i ordenació eficients i 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 complexos.
- 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.
- Profunditzar en el concepte de recursivitat i la seva aplicació al recorregut d'estructures complexes, 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

1. Introducció

Objectius i presentació de l'assignatura. Tractament d'excepcions estructurat

2. Orientació a objectes

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

3. Estructures Lineals

Cues, com implementar-les. Memòria dinàmica i estàtica. Càlcul complexitat amortitzat.

4. Algorismes d'ordenació

Mètode de la bombolla, SelectionSort, InsertionSort, QuickSort. Recursivitat. Càlcul complexitat.

5. Estructures de dades no lineals. Arbres.

Arbres. Arbres binaris. Arbres de cerca binària. Cues de prioritat (Heaps). Recorreguts d'arbres.

6. Estructures de dades no lineals. Grafs

Representacions i recorreguts. Dijkstra, Kruskal.

7. Estructures de dades no lineals. Hash

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

8. Ús de llibreries C++ desde Python. Conceptes bàsics de Python.

Metodologia

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. Així, els dos principals resultats d'aprenentatge hauran de ser els següents. Per una banda, la realització d'un projecte de programació que s'haurà d'anar desenvolupant de forma principalment autònoma durant tot el curs (amb seguiment i control durant les sessions de pràctiques de laboratori) i que suposarà integrar de forma pràctica gairebé tots els conceptes i eines de programació introduïts a les classes de teoria en la resolució d'un problema real complex. Per altra banda, la resolució individual de tots els exercicis que es proposaran al llarg del curs (alguns dels quals es resoldran i discutiran a les sessions de problemes) que han de servir per comprendre, integrar i aplicar els conceptes explicats a classe de teoria. A totes les activitats del curs (teoria, problemes i pràctiques) es farà servir principalment el llenguatge de programació C++, i es donaran alguns conceptes de programació en python.

A nivell presencial, l'assignatura s'organitzarà en tres tipus d'activitats diferents: classes de teoria, sessions de problemes i pràctiques de laboratori.

Classes de teoria

Les classes de teoria serviran per introduir els conceptes que es detallen al temari de l'assignatura. Tot i basar-se en les explicacions del professor i de material complementari en forma transparències, les sessions de teoria també tindran un enfoc força pràctica amb exemples i petits exercicis que es plantejaran als alumnes per facilitar la comprensió i aprenentatge dels conceptes explicats. L'estudiant haurà de completar les classes de teoria amb el treball personal en la realització dels exercicis que es vagin proposant i que han de servir per acabar d'entendre els continguts de l'assignatura. Cal tenir present que el temari de l'assignatura té una continuïtat lògica al llarg del curs, de manera que per poder seguir correctament una classe cal haver assimilat el que s'ha explicat a les sessions anteriors.

Sessions de problemes

Les sessions de problemes (anomenades pràctiques d'aula en els horaris) es dedicaran bàsicament a la resolució i discussió de problemes que serveixin d'aplicació dels conceptes explicats a les classes de teoria. Cada grup de teoria es divideix en dos grups de problemes (grup A i grup B). La divisió dels estudiants entre els dos grups es farà al principi i serà fixa per tot el curs. A les sessions de problemes es treballaran exercicis basats en les explicacions de les sessions de teoria. Per cada tema, es proposarà un conjunt d'exercicis que els estudiants hauran de resoldre de forma autònoma i que hauran de lliurar de forma individual com a part de l'avaluació de l'assignatura.

Pràctiques de laboratori

Durant el curs hi haurà 6 sessions de pràctiques de laboratori que es dedicaran principalment al seguiment i avaluació del treball fet per l'alumne en el projecte de programació. El projecte de programació permet abordar un problema de programació d'una certa complexitat, que es va desenvolupant al llarg de tot el curs. Apart del treball que es faci a les sessions de pràctiques, la realització del projecte s'haurà de fonamentar principalment en el treball autònom dels estudiants. Abans de cada sessió de pràctiques, l'estudiant haurà d'haver realitzat de forma autònoma el treball que s'hagi plantejat, segons la planificació del projecte al llarg del curs. Durant la sessió el professor examinarà el treball realitzat pels estudiants, avaluarà el progrés, corregirà els errors i resoldrà els dubtes que se li plantegin. L'assistència a les sessions de control al laboratori és obligatòria i servirà per determinar part de la nota final del projecte.

El treball del projecte i de les sessions de laboratori s'haurà de fer en grups de 2 persones.

La gestió de la docència de l'assignatura es farà a través de Cerbero (<http://cerbero.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.

Competències Transversals

- **T01.02 Desenvolupar la capacitat d'anàlisi, síntesi i prospectiva.** Aquesta competència es desenvoluparà durant les sessions de teoria, problemes i pràctiques, 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			
Laboratori	12	0,48	1, 2, 3, 4, 6, 7
Problemes	12	0,48	2, 3, 5, 6, 7
Teoria	26	1,04	2, 3, 5, 6
Tipus: Supervisades			
Tutories	1	0,04	1, 2, 3, 4, 5, 6, 7
Tipus: Autònomes			
Laboratori: Implementació del projecte de programació	46	1,84	1, 2, 3, 4, 7
Problemes: Resolució de problemes de manera individual.	34	1,36	1, 2, 3, 4, 5, 6, 7
Teoria: Estudi Individual	13	0,52	2, 3, 5, 6

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.

S'haurà d'aconseguir una **nota mínima de 5** 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%**.

- **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 5** en cadascun dels dos parcials per poder aprovar l'assignatura.

La nota final serà la mitja dels dos parcials: **Avaluació Individual = (0.5 * Parcial1) + (0.5 * 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 de pràctiques de laboratori. 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.

Suspesos: 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 final serà suspès i es posarà un 4.5 a la nota de l'expedient de l'alumne.

Convalidacions: No hi haurà convalidacions de pràctiques.

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 teórico-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
Segón 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/>