

Titulació	Tipus	Curs	Semestre
2502441 Enginyeria Informàtica	OB	2	1

Professor de contacte

Nom: Ricardo Juan Toledo Morales

Correu electrònic: Ricardo.Toledo@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

Prerequisits

Càlcul: és convenient que l'estudiant hagi adquirit els coneixements sobre funcions i límits.

Fonaments de Informàtica: és convenient que l'estudiant ho tingui aprovat i ho repassi.

Metodologia de la Programació: és convenient que l'estudiant ho tingui aprovat i ho repassi.

Objectius

Coneixement de les estructures de dades (ED) en un context d'orientació a objectes i utilitzant el llenguatge PYTHON com element que permeti expressar ambdós coneixements (els objectes i les estructures de dades).

Definir el concepte de tipus abstracte de dades TAD i la seva implicació al disseny del software.

Estudiar les estructures de dades més rellevants, analitzar la complexitat espacial i temporal i les seves aplicacions en algorismes que les fan servir habitualment, per tal que l'estudiant sigui capaç d'escollir, dissenyar i implementar un TAD davant d'un problema que se li plantegi.

Aprofundir en la programació en PYTHON.

Coneixement de les diferents formes d'organitzar i representar la informació en un ordinador juntament amb la definició d'algorismes òptims per gestionar i processar les dades.

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 el pensament sistèmic.
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. Breu introducció a la història de la programació orientada a l'objecte i PYTHON.

2. Orientació a objectes i PYTHON.

Introducció al paradigma d'objectes. Abstracció de dades. Ocultació. Llenguatge PYTHON. Complexitat.

3. Estructures Lineals

Orientació a objectes: herència. Seqüències. Vectors simples. Relació amb referències en PYTHON. TAD Vector. Matrius: representació i operacions. Llistes. Deque. Adaptadors de seqüències: Piles, Cues. Algorismes de cerca i ordenació amb seqüències. Recursivitat. Els alumnes han d'escollir seqüències i aplicar-les per a solucionar problemes.

4. Estructures de dades no lineals. Arbres.

Arbres. Arbres binaris. Arbres de cerca binària. Els alumnes han d'implementar un arbre de cerca binària per optimitzar les cerques. Cues de prioritat (Heaps).

5. Altres estructures de dades no lineals.

Tècniques de "Hashing". Matrius hash i llistes hash. Funcions hash. Grafs. Problemes reals que necessiten solucions amb algorismes de grafs.

Metodologia

Metodologia docent:

La metodologia docent estarà orientada cap a la motivació de la iniciativa i el treball individual i en grup de l'alumne, a aconseguir un procés d'aprenentatge on la figura de l'alumne és la figura principal i el professor en lloc d'un lector passa a ser més bé un tutor. La idea principal és que l'alumne vagi adquirint els coneixements i les habilitats de forma contínua i motivada, ajudat pel professor i els seus companys. Durant tot el semestre l'alumne treballarà en un projecte que ha de dissenyar, implementar, analitzar i presentar amb els seus companys. Aquest projecte serà el punt de referència per explicar els diferents temes vistos durant les classes de teoria. L'alumne, des del principi, coneixerà i definirà el seu projecte i anirà desenvolupant-lo de forma contínua durant tot el semestre. En aquesta línia i d'acord amb els objectius proposats a l'assignatura, el desenvolupament del curs es basarà en diferents activitats:

a) **Classes magistrals (T):** L'objectiu de la classe magistral és donar a conèixer els principals conceptes, coneixements i procediments dintre de l'assignatura, així com ensenyar a raonar de forma crítica respecte les formes de programar amb diferents estructures de dades. S'exposaran de forma clara i molt concisa els algorismes i les estructures bàsiques dins el temari de l'assignatura, s'ensenyarà com i per què s'han utilitzat

las diferents formes d'estructurar les dades i s'ensenyarà a comparar entre les diferents estructures de dades emfatitzant els seus avantatges i desavantatges. Encara que aquest tipus és la forma menys interactiva per ensenyar, el seu paper és molt important per a donar els conceptes bàsics com un punt de partida per treballar a l'assignatura.

b) **Problemes (P)**: Discutirem els problemes i les seves solucions. D'aquesta forma assegurarem que els conceptes bàsics de teoria s'han assolit i que s'ha fet el disseny de la part corresponent del projecte.

c) **Pràctiques (L)**: Durant la sessió de pràctiques els grups d'alumnes han d'anar amb la primera versió del codi fet i tindran l'oportunitat de comprovar que funciona, millorar-lo, depurar-lo i lliurar-lo. Durant aquesta sessió NO s'explicarà cap concepte teòric, ni com s'ha d'iniciar la implementació de la pràctica, donat que tots aquests dubtes s'han d'haver aclarit durant els problemes o les tutories.

d) **Preparació dels problemes (PP)**: L'objectiu d'aquesta activitat és treballar els conceptes vistos a teoria i poder trobar la seva rellevància en l'objectiu del projecte de l'alumne. Durant la preparació del seminari, els alumnes treballaran sobre problemes que permetran veure el significat i la utilitat dels conceptes vistos a les classes de teoria.

e) **Preparació de pràctiques (PL)**: Una vegada fet el disseny de la part del projecte, durant la preparació de pràctiques s'han de preparar els programes que l'implementin. Aquesta activitat és molt important perquè durant el laboratori els alumnes només tindran temps per depurar, aclarir dubtes, optimitzar el seu codi amb o sense l'ajuda del professor i, al final de la sessió, l'han de lliurar.

f) **Estudi (E)**: Les hores d'estudi es refereixen bàsicament a repassar els coneixements vists a classes de teoria, a cercar nova informació i, sobre tot, a preparar l'examen de l'assignatura.

g) **Altres activitats (AA)**: Considerem que un enginyer informàtic no només ha de poder resoldre un problema real, dissenyar i optimitzar la seva solució i treballar en equip, sinó que també ha de poder aprendre de forma autònoma un nou concepte/tema teòric i poder exposar-lo davant els seus companys. Per aquest motiu, dedicarem una part dels seminaris a fer exposicions de temes i problemes nous on els equips d'alumnes han d'aprendre, preparar i exposar de forma clara i concisa un tema nou (per exemple, trobar camins òptims en un graf) i fer una demostració davant els seus companys. Aquesta activitat ha donat molt bons resultats els anys passats encara que es pot aplicar només per una part dels equips d'alumnes per qüestions de disponibilitat horària. Per altra banda, també es passaran enquestes sobre la realització de les tasques del curs. La realització d'aquestes enquestes serà obligatòria per poder fer els lliuraments dels diferents treballs.

h) **Desenvolupament del pensament sistèmic**. En moltes activitats els alumnes veuran els problemes i solucions des d'un punt de vista sistèmic.

Activitats formatives

Títol	Hores	ECTS	Resultats d'aprenentatge
Tipus: Dirigides			
Avaluacions	10	0,4	1, 2, 3, 5, 6, 7
Classes magistrals	26	1,04	1, 2, 3, 4, 5, 6, 7
Laboratori	12	0,48	1, 3, 4, 6, 7
Problemes	12	0,48	1, 2, 3, 5, 6, 7
Tipus: Supervisades			
Tutories	15	0,6	1, 2, 3, 4, 5, 6, 7
Tipus: Autònomes			

Avaluació

Sistema d'avaluació

Aquesta assignatura és presencial. No es podrà al·legar desconeixement dels continguts i avisos desenvolupats a classe, per no assistència.

Les dates d'avaluació continuada i lliurament de treballs es publicaran al gestor documental Cerbero (<http://cerbero.uab.cat>) i poden estar subjectes a canvis de programació per motius d'adaptació a possibles incidències. Sempre s'informarà al Cerbero sobre aquests canvis ja que s'enten que és el mecanisme habitual d'intercanvi d'informació entre professor i estudiants.

Per demostrar que s'han assolit els objectius de l'assignatura s'han de demostrar tant els coneixements com les habilitats adquirides. Es tindrà en compte: la qualitat del treball de laboratori implementat, la implicació dels alumnes als problemes i laboratoris, i els coneixements científic-tècnics adquirits.

Aquell alumne que no ha participat en cap activitat d'avaluació es considera no avaluable (NA).

Aquells alumnes que es presentin a al menys una activitat d'avaluació, no poden tenir una nota final de no presentat. En aquest cas la nota, es calcularà comptant un 0 en les avaluacions no realitzades.

Totes les activitats d'avaluació tenen una nota mínima de u (1). Treure una nota inferior a u (1) en una activitat d'avaluació no recuperable implica el suspens de l'assignatura.

Per aprovar és necessari que l'avaluació de cadascuna de les parts superi el mínim exigít i que l'avaluació total superi els 5 punts. En cas de no superar l'assignatura, la nota numèrica de l'expedient serà el valor menor entre 4.5 i la mitjana ponderada de les notes.

Matrícules d'Honor: per cada assignatura d'un mateix pla d'estudis, esconcediran globalment les matrícules d'honor resultants de calcular el cinc per cent o fracció dels alumnes matriculats en tots els grups de docència de l'assignatura. Només es podran atorgar a estudiants que hagin obtingut una qualificació final igual o superior a 9. Si els candidats superen el 5% es procedirà a ordenar per notes.

A) Avaluació continuada.

B) Examen de recuperació. Pels alumnes que no han assolit els mínims a l'avaluació continuada.

A) Avaluació continuada

L'assistència a les avaluacions de teoria i seminaris, i a les sessions i avaluacions de pràctiques és **obligatòria**.

Teoria i Seminaris: Es realitzaran dues avaluacions parcials sobre els continguts de teoria i seminaris. Aquestes proves podran contenir també alguna pregunta/problema referent a les pràctiques.

Laboratoris: La no assistència a dues sessions implica un suspès. Es realitzaran dues avaluacions parcials. Aquestes avaluacions sumaran el 60% de la nota final de pràctiques. A cada sessió es realitzarà algun tipus d'avaluació individual sobre el contingut de la pràctica d'aquella sessió. La suma d'aquestes avaluacions individuals de sessions i lliuraments sumaran el 40% de la nota final de pràctiques.

Per a aprovar sense examen de recuperació s'ha de treure una nota mínima de 5 a cadascuna de les dues avaluacions parcials de teoria i seminaris, i tenir una nota de pràctiques igual o superior a 5. Per

a la resta d'activitats d'avaluació la nota mínima serà un 1. Si es compleixen aquestes condicions, la nota final es calcularà segons la fórmula:

Nota Final = 0.55 x Mitjana Parcial + 0.45 x Nota de Pràctiques

Si l'alumne ha fet almenys una entrega (parcials o laboratoris) la nota final no pot ser no avaluable. En el cas de no arribar al mínim exigít en alguna de les activitats d'avaluació, si el càlcul de la nota final és igual o superior a 5, es posarà un 4,5 de nota a l'expedient.

B) Examen de recuperació dels parcials. Els laboratoris no es poden recuperar.

Els alumnes que no superin la teoria i els problemes de l'assignatura per avaluació continuada, segons els criteris indicats anteriorment, hauran de fer un examen de recuperació.

En aquest cas es recupera la nota més baixa dels parcials i la nota de recuperació reemplaça la nota del parcial.

La nota mínima de recuperació és un cinc (5).

En aquesta assignatura no es convaliden activitats de cursos anteriors.

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). Per exemple, plagiar, copiar, deixar copiar, ..., una activitat d'avaluació, implicarà suspendre aquesta activitat d'avaluació 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.

Activitats d'avaluació

Títol	Pes	Hores	ECTS	Resultats d'aprenentatge
Examen de recuperació	veure la descripció del mètode d'avaluació	0	0	1, 2, 3, 4, 5, 6, 7
Primer parcial	50% de la nota final de teoria	0	0	2, 3, 5, 6, 7
Primer parcial de laboratori	veure la descripció del mètode d'avaluació	0	0	1, 2, 3, 4, 6, 7
Segon parcial	50% de la nota final de teoria	0	0	2, 3, 5, 6, 7
Segon parcial de laboratori	veure la descripció del mètode d'avaluació	0	0	1, 2, 3, 4, 6, 7

Bibliografia

Bibliografia bàsica:

Fundamentals of data structures in C++. E.Horowitz, S. Sahani, D. Mehta. Computer Science Press, 1995

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/>

Bibliografia complementària:

Estructuras de Datos, Algoritmos y Programación Orientada a Objetos. G. L. Heileman. McGraw-Hill, 1998.

The art of computer programming: sorting and searching. D. Knuth. Addison-Wesley, 1976

Enllaços web:

<http://www.python.org>

<http://www.diveintopython.net>