

Programació Avançada

Codi: 104338
Crèdits: 6

Titulació	Tipus	Curs	Semestre
2503758 Enginyeria de Dades	FB	1	2

Professor/a de contacte

Nom: Ernest Valveny Llobet

Correu electrònic: Ernest.Valveny@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

Prerequisits

L'assignatura no té cap prerequisit oficial. De totes formes, s'assumeix que l'estudiant ha cursat l'assignatura prèvia de Fonaments de programació i, per tant, està familiaritzat amb les estructures bàsiques de la programació.

Objectius

Aquesta assignatura s'ha de veure com la continuació lògica de l'assignatura de Fonaments de Programació. L'objectiu bàsic és ampliar les nocions bàsiques de programació introduïdes a Fonaments de Programació i completar-les els principis de la programació orientada a objectes. La programació orientada a objectes és una de les metodologies més exteses de disseny i programació. A la programació orientada a objectes el software s'organitza en classes que contenen mètodes (procediments) i atributs (dades). Les instàncies o objectes d'aquestes classes s'envien missatges les unes a les altres (fan "trucades" a mètodes d'altres objectes) aconseguint així la funcionalitat desitjada. A part del concepte de classe, tres altres elements fonamentals són l'herència, la composició i la separació de les parts pública i privada de les classes. Aquests elements són doncs necessaris per a la programació OO, però no suficients: cal saber com utilitzar-los per arribar a solucions que siguin fàcilment extensibles en el futur, ja que una llei del software és que els canvis són inevitables. Existeixen una sèrie de principis de disseny o heurístiques que ens diuen com s'han d'emprar aquests elements per tal d'aconseguir-ho. I sobre ells encara, es construeixen els patrons de disseny, que són solucions a nivell de disseny OO a problemes recurrents, que apareixen sovint en resoldre problemes aparentment independents. En aquesta assignatura s'introdueixen i es practiquen els conceptes OO així com els principis i patrons de disseny.

D'aquesta forma, els objectius formatius de l'assignatura són els següents:

- Entendre i aplicar correctament els principis bàsics de la programació orientada a objectes: concepte de classe, objecte i encapsulament de dades.
- Entendre i aplicar a problemes reals els principis d'herència i composició de classes.

- Conèixer i aplicar els principis i patrons de disseny a la resolució de problemes de programació.
- Adquirir la capacitat de disseny i programació d'algorismes per a la resolució de problemes complexos,
- Utilitzar les estructures de programació i de representació de la informació més adients per la implementació d'un algorisme.
- Entendre el cicle de vida del software per la resolució de problemes de programació: anàlisi del problema, disseny, implementació i prova.
- Desenvolupar els programes seguint unes normes d'estil tendents a aconseguir programes de qualitat.

Competències

- Avaluar de manera crítica el treball realitzat.
- Buscar, seleccionar i gestionar de manera responsable la informació i el coneixement.
- Demostrar sensibilitat cap als temes ètics, socials i mediambientals.
- Dissenyar solucions algorítmiques eficients per a problemes computacionals, implementar-les en forma de desenvolupaments de programari robust, estructurat i fàcil de mantenir, i verificar-ne la validesa.
- Planificar i gestionar el temps i els recursos disponibles.
- Que els estudiants hagin demostrat que comprenen i tenen coneixements en una àrea d'estudi que parteix de la base de l'educació secundària general, i se sol trobar a un nivell que, si bé es basa en llibres de text avançats, inclou també alguns aspectes que impliquen coneixements procedents de l'avantguarda d'aquell camp d'estudi.

Resultats d'aprenentatge

1. Aplicar els principis bàsics de l'estructura i la programació dels computadors.
2. Aplicar estratègies de depuració, prova i correcció de programes.
3. Avaluar de manera crítica el treball realitzat.
4. Buscar, seleccionar i gestionar de manera responsable la informació i el coneixement.
5. Demostrar sensibilitat cap als temes ètics, socials i mediambientals.
6. Desenvolupar programes amb un bon estil de programació i ben documentats.
7. Implementar en un llenguatge de programació problemes algorítmics de dificultat mitjana.
8. Planificar i gestionar el temps i els recursos disponibles.
9. Que els estudiants hagin demostrat que comprenen i tenen coneixements en una àrea d'estudi que parteix de la base de l'educació secundària general, i se sol trobar a un nivell que, si bé es basa en llibres de text avançats, inclou també alguns aspectes que impliquen coneixements procedents de l'avantguarda d'aquell camp d'estudi.

Continguts

1. Conceptes d'orientació a objecte: encapsulament, herència, composició, interfície. Diagrames de classe UML
2. Programació orientada a objecte en Python3 i Java
3. Principis generals de disseny orientat a objecte
4. Patrons de disseny
5. Estil de codificació, assercions i missatges de logging

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. Les sessions presencials de classe s'organitzaran per introduir els continguts teòrics de l'assignatura, des d'una perspectiva molt pràctica a partir d'exemples i d'exercicis i problemes de programació que s'hauran de resoldre a classe directament amb l'ordinador.

Per una altra banda, es realitzarà un projecte de programació que s'haurà d'anar desenvolupant de forma principalment autònoma durant tot els curs (amb seguiment i control per part del professor en sessions puntuals) i que suposarà integrar de forma pràctica gairebé tots els conceptes i eines de programació introduïts a les sessions presencials en la resolució d'un problema real complex. A més a més, es proposarà un conjunt d'exercicis que s'hauran de resoldre de forma individual al llarg del curs (alguns dels quals es resoldran i discutiran a les sessions presencials) que han de servir per comprendre, integrar i aplicar els conceptes desenvolupats a les sessions presencials. A les activitats del curs (sessions presencials, problemes i pràctiques) es farà servir principalment el llenguatge de programació Python, però també s'utilitzarà Java en alguns exercicis per introduir millor alguns dels conceptes de la programació orientada a objectes.

A nivell presencial, les sessions de classe s'organitzaran en quatre hores setmanals i es faran en una aula amb ordinadors per facilitar el treball pràctic de l'estudiant. S'encoratja que l'alumne porti el seu propi portàtil a classe si en disposa d'un. A les sessions presencials s'aniran introduint els conceptes que es detallen al temari de l'assignatura. En alguns casos, es podran posar a disposició de l'estudiant vídeos explicatius o altre material complementari que l'estudiant haurà de visionar abans de la sessió de classe. Les sessions de classe tindran un enfoc força pràctic amb exemples i exercicis que es plantejaran als alumnes per facilitar la comprensió i aprenentatge dels conceptes explicats. Aquests exercicis es realitzaran i discutiran durant la sessió i serviran per anar introduint els continguts de l'assignatura i veure'n la seva aplicació pràctica.

L'estudiant haurà de completar les classes presencials amb el treball personal autònom 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. Alguns d'aquests exercicis s'hauran de lliurar de forma individual com a part de l'avaluació de l'assignatura.

A més a més, els estudiants hauran de fer en grups de 2 un projecte de programació que es desenvoluparà de forma autònoma durant tot el curs fora de les sessions presencials. El projecte de programació permetrà abordar un problema de programació d'una certa complexitat que integri la majoria dels conceptes explicats durant el curs. Durant el curs, es dedicaran algunes sessions presencials al control, seguiment i avaluació del treball fet per l'alumne en el projecte de programació.

Activitats formatives

Títol	Hores	ECTS	Resultats d'aprenentatge
Tipus: Dirigides			
Sessions presencials	50	2	1, 3, 4, 5, 7, 9
Tipus: Supervisades			
Implementació projecte programació	50	2	1, 2, 6, 7, 8
Tipus: Autònomes			
Estudi individual	14	0,56	4, 8
Resolució de problemes	32	1,28	1, 2, 6, 7, 8

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:

$$\text{Nota Final} = (0.3 * \text{Avaluació Problemes}) + (0.4 * \text{Projecte}) + (0.3 * \text{Avaluació Individual})$$

Lliurament de problemes: en aquest apartat s'inclou el lliurament dels exercicis que es proposin al llarg del curs.

- 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 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:

$$\text{Avaluació Individual} = (0.5 * \text{Parcial 1}) + (0.5 * \text{Parcial 2})$$

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 al llarg del curs. La nota final es calcularà de la forma següent:

$$\begin{aligned} \text{Projecte} = & (0.15 * \text{Avaluació seguiment projecte}) + (0.25 * \text{Entrega Parcial 1}) \\ & + (0.6 * \text{Entrega Final}) \end{aligned}$$

- S'haurà d'aconseguir 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 la nota global del **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 evaluable: Un alumne es considerarà no evaluable (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.

Nota important: 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.

Activitats d'avaluació

Títol	Pes	Hores	ECTS	Resultats d'aprenentatge
Avaluació individual	30%	4	0,16	1, 9
Lliurament de problemes	30%	0	0	1, 2, 3, 6, 7
Projecte de programació	40%	0	0	2, 4, 5, 6, 7, 8

Bibliografia

- Python: master the art of design patterns. D. Phillips, C. Giridhar, S. Kasampalis. Packt Publishing, 2016.
- Mastering object-oriented Python. Steven F. Lott. Packt publishing, 2014.
- Head first object oriented analysis and design. B.D. McLaughlin, G. Pollice, D. West. O'Reilly, 2007.
- Design patterns explained simply. Alexander Shvets. <https://sourcemaking.com/design-patterns-ebook>.
- Head first design patterns. E. Freeman, E. Freeman, K. Sierra, B. Bates. O'Reilly, 2004.
- Clean code: a handbook of agile software craftsmanship. R.C. Martin. Prentice Hall, 2008. O la versió en castellà, Código limpio, d'Anaya Multimedia 2012.