

Guia docent

340371 - PRO1-I2023 - Programació I

Última modificació: 04/07/2024

Unitat responsable: Escola Politècnica Superior d'Enginyeria de Vilanova i la Geltrú
Unitat que imparteix: 723 - CS - Departament de Ciències de la Computació.

Titulació: GRAU EN ENGINYERIA INFORMÀTICA (Pla 2018). (Assignatura obligatòria).

Curs: 2024 **Crèdits ECTS:** 7.5 **Idiomes:** Català

PROFESSORAT

Professorat responsable: Bernardino Casas Fernández

Altres: Bernardino Casas Fernández
Alejandro Ríos Jerez

CAPACITATS PRÈVIES

Coneixement de les tècniques de programació imperativa:

- instruccions bàsiques: assignació, alternativa, iteració
- accions, funcions i pas de paràmetres
- vectors, tuples i seqüències
- esquemes de recorregut i cerca

Conèixer bé un llenguatge imperatiu, preferentment C++. Experiència en l'edició, compilació i execució de programes en C++ a l'entorn Linux.

REQUISITS

Haver aprovat FOPR o almenys estar-hi matriculat.

COMPETÈNCIES DE LA TITULACIÓ A LES QUALS CONTRIBUEIX L'ASSIGNATURA

Específiques:

1. CEFB3. Capacitat per a comprendre i dominar els conceptes bàsics de matemàtica discreta, lògica, algorítmica i complexitat computacional, i la seva aplicació per al tractament automàtic de la informació per mitjà de sistemes computacionals i la seva aplicació per a la resolució de problemes propis de l'enginyeria.
2. CEFB4. Coneixement dels fonaments de l'ús i programació dels computadors, els sistemes operatius, les bases de dades i, en general, els programes informàtics amb aplicació en enginyeria.
3. CEFB5. Coneixement de l'estructura, funcionament i interconnexió dels sistemes informàtics, així com els fonaments de la seva programació.
4. CEFC6. Coneixement i aplicació dels procediments algorítmics bàsics de les tecnologies informàtiques per a dissenyar solucions a problemes, analitzant la idoneïtat i complexitat dels algorismes proposats.
5. CEFC7. Coneixement, disseny i utilització de forma eficient els tipus i estructures de dades més adequades a la resolució d'un problema.

Transversals:

6. APRENTATGE AUTÒNOM - Nivell 1: Dur a terme les tasques encomanades en el temps previst, tot treballant amb les fonts d'informació indicades, d'acord amb les pautes marcades pel professorat.
7. COMUNICACIÓ EFICAÇ ORAL I ESCRITA - Nivell 1: Planificar la comunicació oral, respondre de manera adequada les qüestions formulades i redactar textos de nivell bàsic amb correcció ortogràfica i gramatical.
8. TREBALL EN EQUIP - Nivell 1: Participar en el treball en equip i col·laborar-hi, un cop identificats els objectius i les responsabilitats col·lectives i individuals, i decidir conjuntament l'estratègia que s'ha de seguir.
9. ÚS SOLVENT DELS RECURSOS D'INFORMACIÓ - Nivell 1: Identificar les pròpies necessitats d'informació i utilitzar les col·leccions, els espais i els serveis disponibles per dissenyar i executar cerques simples adequades a l'àmbit temàtic.

METODOLOGIES DOCENTS

L'assignatura consta de:

- 2 hores a la setmana de classes presencials a l'aula (grup gran) en les que es presenten, de forma participativa, els conceptes i procediments associats als continguts de l'assignatura, i
- 3 hores a la setmana a l'aula de laboratori (grup petit) en les que s'apliquen els conceptes i tècniques apresos.

La resolució de problemes es fa tant dins com fora de l'aula (treball personal).

OBJECTIUS D'APRENTATGE DE L'ASSIGNATURA

L'objectiu de l'assignatura és consolidar les tècniques bàsiques de disseny d'algorismes per a la resolució de problemes mitjançant l'ordinador, en els àmbits científic i tècnic, i aprendre els fonaments de tècniques avançades com la programació recursiva, el disseny modular i la programació basada en objectes.

En acabar el curs, l'estudiant ha de:

- Dominar els conceptes de classe, objecte, atribut i mètode.
- Conèixer els elements de l'especificació d'una classe de dades i els elements de la seva implementació.
- Saber construir programes que usin classes corresponents a estructures lineals simples (pila, cua, llista, vector) i arborescents (arbre binari, arbre general).
- Conèixer les fases del disseny modular.
- Comprendre la recursivitat lineal i la múltiple, i la seva relació amb la iterativitat.
- Dissenyar programes recursius i iteratius que siguin correctes i eficients.
- Saber explicar els diferents tipus d'immersions d'especificació i les seves característiques.
- Implementar una estructura de dades amb requeriments específics sobre les seves operacions i l'eficiència de les mateixes usant tipus recursius de dades (o punters).

HORES TOTALES DE DEDICACIÓ DE L'ESTUDIANTAT

Tipus	Hores	Percentatge
Hores grup petit	45,0	24.00
Hores aprenentatge autònom	112,5	60.00
Hores grup gran	30,0	16.00

Dedicació total: 187.5 h

CONTINGUTS

1.- Disseny modular i disseny basat en objectes

Descripció:

Abstracció i la seva necessitat. Descomposició funcional i per dades. Mòduls. Ocultació de la informació. Encapsulat. Fases del disseny modular: distinció entre especificació i implementació. Tipus de mòduls i el seu ús. Biblioteques. Principis bàsics de disseny basat en objectes: classes i objectes; camps i mètodes. Implementació de dissenys modulars en C++. Compilació separada i muntatge. Depuració, prova i documentació de programes modulars.

Objectius específics:

- Diferenciar els rols d'usuari, especificador i implementador de classes de dades. Enumerar els elements de l'especificació d'una classe de dades. Enumerar els elements de la implementació d'una classe de dades.
- Dissenyar una classe de dades amb una clara independència entre la seva especificació i la seva implementació. Justificar que l'única forma de crear, consultar o modificar un objecte d'una classe de dades sigui a través de les operacions de l'especificació de la mateixa.
- Resoldre en C++ qualsevol exercici basat en l'aplicació d'un esquema algorísmic senzill sobre un vector format per elements d'una classe de dades.
- Donada una implementació per a una classe de dades senzilla, introduir millores a la representació i a les seves operacions.
- Dissenyar de forma individual un programa modular en C++ a partir de les abstraccions de dades extretes del enunciat d'un problema. Afegir o modificar alguna funcionalitat d'un programa modular en C++ donat.
- Preparar un programa en C++ que faci servir tipus simples i classes de dades, algunes d'elles predefinides i d'altres definides pel propi alumne, per poder ser executat.

Activitats vinculades:

Activitat 1 i Control Parcial.

Dedicació: 20h

Grup gran/Teoria: 2h

Grup mitjà/Pràctiques: 3h

Grup petit/Laboratori: 2h

Activitats dirigides: 1h

Aprenentatge autònom: 12h

2.- Estructures de dades lineals i arborescents

Descripció:

Piles, cues i llistes: especificació i ús (operacions de cerca i recorregut). Iteradors: definició i ús.

Arbres binaris, n-aris i generals: especificació i ús (operacions de cerca i recorregut).

Objectius específics:

- Conèixer els tipus de dades utilitzats habitualment per representar i manegar estructures de dades lineals i la seva especificació. Dissenyar algorismes iteratius i recursius per resoldre problemes de cerca i recorregut en piles, cues i llistes, usant les operacions del tipus de dades corresponent i iteradors (quan sigui recomanable).
- Conèixer tipus de dades utilitzats per representar i manegar estructures de dades arborescents i la seva especificació. Dissenyar algorismes recursius per resoldre problemes de cerca i recorregut en arbres binaris usant les operacions del tipus de dades corresponent.

Activitats vinculades:

Activitat 2, Control Parcial i Control Final.

Dedicació: 26h

Grup gran/Teoria: 3h

Grup mitjà/Pràctiques: 4h

Grup petit/Laboratori: 2h

Activitats dirigides: 1h

Aprenentatge autònom: 16h

3.- Programació recursiva metòdica

Descripció:

Justificació de la correctesa d'algorismes recursius. Immersió (o generalització) d'una funció. Immersions d'especificació: per afebliment de la postcondició i per reforçament de la precondició. Relació entre algorismes recursius lineals finals i algorismes iteratius.

Objectius específics:

- Descriure els passos principals del disseny d'algorismes recursius. Justificar la correctesa d'algorismes recursius relativament senzills.
- Conèixer el concepte d'immersió d'una funció i explicar la diferència entre immersions d'especificació i immersions d'eficiència. Conèixer els diferents tipus d'immersions d'especificació i els diferents tipus d'immersions d'eficiència.
- Donat un algorisme recursiu, determinar si existeix una manera senzilla d'obtenir un algorisme iteratiu equivalent, i si és així, escriure'l.

Activitats vinculades:

Control Parcial, Control Final i Pràctica.

Dedicació: 39h 30m

Grup gran/Teoria: 4h

Grup mitjà/Pràctiques: 6h

Grup petit/Laboratori: 4h

Activitats dirigides: 1h

Aprenentatge autònom: 24h 30m

4.- Programació iterativa metòdica

Descripció:

Invariant d'un bucle. Justificació de la correctesa d'algorismes iteratius.

Objectius específics:

- Descriure els passos principals del disseny d'algorismes iteratius. Justificar la correctesa d'algorismes iteratius relativament senzills.

Activitats vinculades:

Control Parcial.

Dedicació: 24h 30m

Grup gran/Teoria: 2h

Grup mitjà/Pràctiques: 3h 30m

Grup petit/Laboratori: 3h

Activitats dirigides: 1h

Aprenentatge autònom: 15h

5.- Milllores d'eficiència en programes recursius i iteratius

Descripció:

Detecció de la repetició de càlculs en programes recursius i iteratius. Immersions d'eficiència: noves (paràmetres d'entrada) i/o resultats (valors retornats o paràmetres de sortida) en operacions recursives per millorar l'eficiència. Noves variables locals que usen el seu valor en iteracions anteriors per millorar l'eficiència d'operacions iteratives.

Objectius específics:

- Determinar si el cost d'un algoritme iteratiu o recursiu senzill donat que treballi sobre vectors, piles, cues, llistes o arbres és lineal o si és quadràtic (suposant que el cost sigui d'un d'aquests dos tipus).
- Determinar si es pot millorar l'eficiència d'un algoritme recursiu senzill donat i, en cas de que sigui possible, dissenyar un algoritme recursiu alternatiu més eficient usant immersions d'eficiència.
- Determinar si es pot millorar l'eficiència d'un algoritme iteratiu senzill donat i, en cas de que sigui possible, dissenyar un algoritme iteratiu alternatiu més eficient.

Activitats vinculades:

Control Final i Pràctica.

Dedicació: 20h

Grup gran/Teoria: 2h

Grup mitjà/Pràctiques: 3h

Grup petit/Laboratori: 2h

Activitats dirigides: 1h

Aprenentatge autònom: 12h

6.- Tipus recursius de dades

Descripció:

Introducció a l'ús de tipus recursius de dades. El constructor de tipus punter i la gestió de memòria dinàmica. Implementació d'estructures de dades enllaçades mitjançant tipus recursius de dades. Algorismes iteratius i recursius per resoldre problemes de cerca i recorregut en estructures de dades enllaçades accedint directament a la representació basada en nodes i punters a nodes.

Objectius específics:

- Implementar una estructura de dades amb requeriments específics sobre les seves operacions i/o l'eficiència de les mateixes usant tipus recursius de dades (o estructures de dades enllaçades).
- Dissenyar algorismes iteratius i recursius per resoldre problemes de cerca i recorregut en estructures de dades enllaçades utilitzant directament la seva representació.

Activitats vinculades:

Control Final.

Dedicació: 20h

Grup gran/Teoria: 2h

Grup mitjà/Pràctiques: 3h

Grup petit/Laboratori: 2h

Activitats dirigides: 1h

Aprenentatge autònom: 12h

SISTEMA DE QUALIFICACIÓ

C1 = Control Parcial. Prova individual (2 hores).

C2 = Control Final. Prova individual (màx. 3 hores) de caràcter global.

A1 = Activitat individual (proposta d'activitat o resolució de problemes de la plataforma Jutge).

A2 = Activitat individual (resolució de problemes de verificació).

Pràctica = Nota resultant corresponent als lliuraments de la pràctica.

Qualificació Final = $(A1*0,5 + A2*0,5)*0,15 + \max((C1*0,5 + C2*0,5), C2) *0,60 + Pràctica*0,25$

Mòdul de pràctiques de laboratori:

Es compon del treball pràctic que l'alumnat realitza per l'assignatura. En concret, la realització d'una pràctica de programació en equips de dues persones on es fan servir els coneixements que s'aprenen a l'assignatura. La pràctica s'avalua a partir de: l'execució del programa realitzat, el codi presentat i una Prova de Validació Individual (PVI). La Prova de Validació Individual pot realitzar-se mitjançant una entrevista, el seguiment a les sessions de laboratori o bé en el Control Final de l'assignatura. La realització i presentació de la pràctica (que recull el treball de pràctiques realitzat durant el curs) serà condició necessària per a la superació de l'assignatura. En cas contrari, la qualificació final de tota l'assignatura serà de 'No Presentat'.

La reavaluació, prova de màxim 3 hores, substitueix la nota dels dos controls, per tant correspon a un 60% de la qualificació final.

NORMES PER A LA REALITZACIÓ DE LES PROVES.

Les proves Control Parcial, Control Final i Reavaluació, són presencials i individuals.

Les proves Activitat 1 i Activitat 2 són no presencials i individuals.

La Pràctica es realitza en equips de dues persones. Es lliura de forma no presencial i s'avalua tant de forma presencial com de forma no presencial a partir de la documentació presentada.

BIBLIOGRAFIA

Bàsica:

- Musser, David R.; Derge, Gillmer J.; Saini, Atul. STL tutorial and reference guide : C++ programming with the standard template library. 2nd ed. Boston [etc.]: Addison-Wesley publishing company, 2000. ISBN 9780321702128.
- Weiss, Mark Allen. Data structures and problem solving using C++. 2a ed. Upper Saddle River: Pearson Education International, 2003. ISBN 0321205006.

Complementària:

- Balcázar, José Luis. Programación metódica. Madrid [etc.]: McGraw-Hill, 1993. ISBN 8448119576.
- Castro, Jorge. Curs de programació. Madrid [etc.]: McGraw-Hill, 1992. ISBN 844810031X.

RECURSOS

Enllaç web:

- <http://c.conclase.net/curso/index> . Curs de C++. Per ser usat com a manual de referència.
- <http://www.cplusplus.com/reference/stl/>. Manual de referència sobre els containers de la llibreria STL de C++.