

SYLLABUS
pentru disciplina:

“PROGRAMARE ORIENTATĂ PE OBIECTE”

FACULTATEA AUTOMATICĂ ȘI CALCULATOARE
DOMENIUL/SPECIALIZAREA INGINERIA SISTEMELORAnul de studii: II
Semestrul I

Titularul cursului: prof. dr. ing. Stoicu-Tivadar Vasile					
Colaboratori: (Nume și prenume, titul științific, grad didactic; departamentul de care aparține)	Berian Dorin	dr. ing.	Ș.l.	AIA	
	Pintea Romina	drd. ing.	drd. cu frecvență (prep.)	AIA	
Număr de ore/săptămână / Verificarea / Credite					
Curs	Seminar	Laborator	Proiect	Examinare	Credite
2,5	0	1,5	0,5	E	5
Statul disciplinei	Fundamentală <input type="checkbox"/>	În domeniu <input type="checkbox"/>	De specialitate <input checked="" type="checkbox"/>	Complementară <input type="checkbox"/>	
	Obligatorie: Impusă <input checked="" type="checkbox"/>	Opțională <input type="checkbox"/>		Facultativă <input type="checkbox"/>	

A. OBIECTIVELE DISCIPLINEI

Cursul furnizează abilitățile necesare pentru ca studenții să poată dezvolta programe folosind sisteme de calcul moderne și o abordare orientată pe obiecte, plecând de la cunoștințele de bază de programare. Cursul are un caracter pronunțat aplicativ. **Rezultatele învățării** vor fi reflectate în cunoașterea unui limbaj de programare de nivel înalt cu ajutorul cărora să creeze programe. De asemenea, studenții vor dezvolta **abilități** de rezolvare a problemelor, abilitatea de a analiza o problemă de programare de nivel mediu, dezvoltarea unei soluții prin proiectare în programare de nivel înalt și implementarea proiectului într-un limbaj de programare, toate acestea folosind abordarea specifică, orientată pe obiecte. Studenții sunt ghidați pentru a avea o **atitudine** corectă față de viitoarea profesie, prin antrenarea **valorilor** specifice muncii în echipă și eticii profesionale.

Disciplina contribuie la dezvoltarea **competenței** Proiectarea asistată de calculator și utilizarea de software dedicat - 100 %, respectiv 1,9 % din total.

Obiective:

Transmiterea cunoștințelor de bază despre paradigma programării orientate pe obiecte, crearea deprinderilor teoretice și practice necesare analizei, proiectării, codificării și testării aplicațiilor de complexitate medie în limbajul C++, inclusiv utilizarea șabloanelor.

Înțelegerea modului în care cunoștințele despre programarea orientată pe obiecte pot fi aplicate în programarea Windows (studiu de caz).

B. SUBIECTELE CURSULUI

Paradigma programării orientate pe obiecte (2 h).

Limbajul C++. Generalități. Programarea în stilul C++, operatori de I/E, variabile locale, clase, supraîncărcare, moștenire, constructori/destructori (3 h)

Clase și obiecte. Clase, structuri, uniuni, funcții prietene, funcții inline, constructori, membri statici, manipularea obiectelor (4 h)

Matrice, pointeri și referințe. Matrice de obiecte (2 h), pointeri către obiecte, referințe (2 h), alocarea dinamică (2 h)

Supraîncărcarea. Supraîncărcarea funcțiilor (2 h) supraîncărcarea operatorilor (2 h)

Moștenirea. Tipuri de moștenire. Polimorfismul cu funcții virtuale (proprietăți, utilizare). (4 h)

Operații de I/E în C++. Stream-uri, clase utilizate, variabila de stare, formatarea datelor, supraîncărcarea operatorilor << și >>, lucrul cu fișiere. (4 h)

Studiu de caz: introducerea în programarea Windows. Stilul windows, funcții API, bucle de mesaje, ierarhia MFC. (4 h)

Tipare. Containere, iteratori, algoritmi, colecții. Standard Template Library. (4 h)

C. SUBIECTELE APLICATIILOR (laborator, seminar, proiect)

Laborator:

Completări aduse de C++ față de limbajul C (2 h)

Încapsularea prin intermediul claselor (2 h)

Pointeri la metode. (2 h)

Constructorii, destructorii. (2 h)

Funcții și clase friend. (2 h)

Derivarea claselor. (2 h)

Metode virtuale. (3 h)

Utilizarea listelor eterogene. (4 h)

Moștenirea multiplă. (2 h)

Proiect: Teme individuale de complexitate medie incluzând utilizarea tiparelor (7 h)

Exemplu: Aplicație de editare grafică prin crearea și utilizarea unei ierarhii de clase specifice.

D. METODE DIDACTICE FOLOSITE

- *Curs* - Tehnicile didactice folosite sunt: expunere, conversație, explicație, exemplu, demonstrație, analiză comparativă, studiu de caz, problematizare, metoda proiectelor. Cursul este prezentat pe videoprojector. Programele sunt rulate în direct și se discută cazuri pe modificări de parametri sau diverse situații care pot genera erori.
- *Laborator* - Tehnicile didactice folosite sunt: expunere, conversație, explicație, exemplu, demonstrație, brainstorming. La începutul laboratorului asistentul face o expunere cu exemple de 10 minute, urmată de discuții, apoi studenții rezolvă aplicațiile din laborator sub îndrumarea cadrului didactic, în final sunt notați.
- *Proiect* - Tehnicile didactice folosite sunt explicație, exemplu, demonstrație, brainstorming. Studenții trebuie să studieze documentația necesară, să împartă proiectul în task-uri, să își rezolve fiecare task-urile și apoi să le includă în proiectul final. Studenții trebuie să colaboreze în echipe mici (2-3 persoane) pentru realizarea proiectelor, exersându-și astfel abilitățile de lucru în echipă și de comunicare. Totodată trebuie să dezvolte și o documentație a proiectului. Acesta trebuie susținut în final. Toate acțiunile studenților sunt realizate sub îndrumarea cadrului didactic.

E. PROCEDURA DE EVALUARE

Nota de activitate pe parcurs este obținută prin ponderarea notelor de la laborator și proiect conform timpului alocat. Notele vor ține cont de implicarea în activitate, de calitatea acesteia, de modul de prezentare.

Examinarea conține 2 probe: test-grilă (scris, 60 întrebări referitoare la conținutul cursului, din care cel puțin 20 cu conținut preponderent practic - cu cod sursă) cu pondere 40 % din nota de examen și proba pe calculator (problemă de C++, 1 h 30 min) cu pondere de 60 % din nota de examen.

Nota de examen are pondere 2/3 din nota finală.

F. BIBLIOGRAFIE

1. Stoicu-Tivadar, V., *Programare orientată pe obiecte*, Editura Orizonturi Universitare, Timisoara, 2000 (la biblioteca UPT)
2. Schildt, H., "C++ manual complet", Editura Teora, București, 1997 (la biblioteca UPT)
3. Somnea, D., Turturea, D., "Inițiere în C++". *Programarea orientată pe obiecte*, Editura Tehnică, București, 1993

G. COMPATIBILITATE INTERNACIONALA

Toate universitățile cu profil informatic sau știința calculatoarelor, exemple:

Technische Universität Wien

UMIST Manchester

California Institute of Technology

Universitatea Tehnică și Economică din Budapesta (Budapest University of Technology and Economics) etc.

Data: 30.03.2009

ȘEF DEPARTAMENT
Prof. dr. ing. SILEA Ioan

TITULAR DE DISCIPLINĂ,
Prof. dr. ing. STOICU-TIVADAR Vasile