UNIVERSITATEA "DUNĂREA DE JOS" DIN GALAŢI
FACULTATEA STIINTA CALCULATOARELOR

SPECIALIZAREA : CALCULATOARE

FORMA DE ÎNVĂŢĂMÂNT
ZI
ANUL DE STUDIU: I
SEMESTRUL : II

FISA DISCIPLINEI

LIMBAJE DE PROGRAMARE ORIENTATE PE OBIECTE
UG-FSC-C COBF108

Număr credite: 5

1. Obiectivele disciplinei

Studierea principiilor programarii orientate pe obiecte prin intermediul limbajului de programare C++. Insusirea de catre studenti a principiilor programarii orientate pe obiecte. Dezvoltarea deprinderilor de utilizare a limbajului C++ pentru programarea orientata obiect.

2. Rezultatele invatarii (*se exprima in obiective masurabile ce fac subiectul evaluarii*)

a. Cunostintele generale

La sfarsitul cursului, studentii trebuie sa stapaneasca principiile de baza si mecanismele specifice programarii orientate obiect (notiunile de clasa, obiect, mostenire). Trebuie sa fie capabil sa realizeze intr-un mod profesionist aplicatii orientate pe obiecte (in limbajul C++), pentru rezolvarea unor probleme practice.

b. Cunostintele de specialitate

Studentul trebuie sa isi insuseasca notiunile specifice programarii orientate obiect (notiunea de clasa, obiect, metoda, functie prietena, mostenire, legare dinamica) si mecanismele specifice acesteia.

Studentul trebuie sa detina deprinderi de programare in limbajulul C++.

Studentul trebuie sa fie capabil sa conceapa si sa implementeze o ierarhie de clase folosita in rezolvarea unor probleme concrete.

Studentul trebuie sa isi insuseasca un stil de programare profesionist.

c. Competentele generale

Nu este cazul

d. Competentele de specialitate

Competente de dezvoltare: notiuni de limbaje orientate obiect, notiuni de algoritmi, notiuni despre metodele specifice, normele si instrumentele programarii orientate obiect, notiuni despre tehnici de dezvoltare a aplicatiilor orientate obiect.

e. Abilitatile cognitive specifice

Nu este cazul.

3. Concordanţa cu obiectivele planului de învăţământ/specializării

a. Contributia rezultatelor invatarii disciplinei la formarea competentelor specializarii
   La aceasta disciplina studentii invata principiile de baza al programarii orientate obiect si dobandesc deprinderea de dezvolta aplicatii intr-un limbaj de programare orientat obiect. Acestea contribuie la formarea unui specialist capabil sa programeze intr-un stil profesionist.
b. Cerintele disciplinare prealabile
   Cunostintele dobandite la disciplina „Programarea calculatoarelor si limbaje de programare I" (AOBF103).

4. Structura activitatii didactice

CURS	28 ore
Seminar	0 ore
Lucrări practice	28 ore
Proiect	0 ore

5. Prezentarea conţinutului disciplinei

a. Curs	Continutul activitatii	Nr. ore
1: Concepte de baza ale programarii orientate obiect: Abstractizarea datelor. Mostenirea. Incapsularea (ascunderea) informatiei. Legarea dinamica. Alte aspecte.		*(2 ore)*
2: Clase si obiecte: Definirea claselor si accesul la membri. Instantierea claselor. Membrii unei clase. Pointerul this. Domeniul unui nume, vizibilitate si timp de viata. Functii inline. Tablouri de obiecte. Functii prietene		*(6 ore)*
3: Supraincarcarea operatorilor: Moduri de supraincarcare a operatorilor. Restrictii la supraincarcarea operatorilor. Membrii constanti ai unei clase. Supraincarcarea operatorilor insertor, extractor, de atribuire, de indexare, new, delete, ( ), ->. Conversii		*(8 ore)*
4: Crearea ierarhiilor de clase: Mecanismul mostenirii. Clase derivate si clase de baza. Relatia dintre constructorii si destructorii claselor de baza si ai clasei derivate. Mostenirea simpla si mostenirea multipla. Redefinirea datelor membru ale unei clase de baza intr-o clasa derivata. Supraincarcarea functiilor membru ale unei clase de baza intr-o clasa derivata. Pointeri si referinte la clasa de baza si la clasele derivate. Clase virtuale. Metode virtuale si polimorfism. Metode virtuale pure si clase abstracte.		*(6 ore)*
5: Operatii de intrare/iesire in limbajul C++: Principii de baza. Testarea si modificarea starii unui flux. Formatarea datelor din fluxurile de intrare/iesire. Metodele clasei istream. Metodele clasei ostream. Manipulatori creati de utilizator. Fluxuri de date pentru fisiere. Fisiere binare.		*(6 ore)*

b. Seminar

---

c. Lucrări practice

Continutul activitatii	Nr. ore
Aplicatii in care se utilizeaza clase si obiecte, metodele si functiile prietene ale unei clase.	*(4 ore)*
Implementarea de destructori si a diferitelor tipuri de constructori. Aplicatii in care obiectele constituie parametri in functii.	*(2 ore)*
Aplicatii la supraincarcarea operatorilor: implementarea claselor complex si fractie. Dezvoltarea exemplelor prezentate la curs prin adaugarea de noi functii membre si supraincarcarea unor operatori.	*(8 ore)*
Implementarea clasei vector. Dezvoltarea exemplului prezentat la curs prin adaugarea de noi functii membre si supraincarcarea unor operatori. Completarea aplicatiei prin implementarea metodelor de sortare.	*(4 ore)*
Implementarea clasei matrice. Dezvoltarea exemplului prezentat la curs prin adaugarea de noi functii membre si supraincarcarea unor operatori	*(2 ore)*
Aplicatii in care se implementeaza ierarhii de clase. Completarea exemplelor (din curs) de ierarhii de clase care ilustreaza mostenirea simpla si multipla. Aplicatii complexe, lucrul cu clase abstracte.	*(4 ore)*
Aplicatii in care se utilizeaza operatiile de intrare/iesire din C++.	*(4 ore)*

d. Proiect

---

6. Invatare

a. Forme de invatare/predare

Conversatia euristica, expunerea, problematizarea, invatarea prin rezolvarea de probleme, studiul de caz, dialogul, inductia.

b. Resurse educationale

1. Stefanescu D., Curs C/C++, I si II, www.ariadne.ugal.ro
2. Stefanescu D., Indrumar de laborator la disciplina limbaje de programare, I si II, www.ariadne.ugal.ro

c. Bibliografie disponibila

1. Catrina O, Cojocaru I, Turbo C++, Ed. Teora, Bucuresti, 1993
2. Jamsa K, Klander L, Totul despre C si C++ - Manualul fundamental de programare, Ed. Teora, Bucuresti, 2005
3. Negrescu L., Limbajul C si C++ pentru incepatori, vol. I si II, Ed. Teora, Bucuresti, 2003
4. Stefanescu D., Programarea in limbajele C/C++, MATRIXRom, Bucuresti, 2002
5. Stefanescu D., Segal C., Initiere in limbajele C/C++, Editura Fundatiei Universitare "Dunarea de Jos", Galati, 2000
6. Stroustrup B., C++, Ed. Teora, Bucuresti, 2001

d. Alte resurse

7. Studiu individual: **100 ore**

8. Evaluare

a. Forme de evaluare

Evaluare formativa si cu caracter de diagnosticare, realizata in cadrul activitatii de laborator, prin: teste teoretice periodice si discutarea rezultatelor, urmarirea modului in care studentii rezolva problemele propuse cu scopul de a depista si corecta greselile acestora (nota N11).

Evaluare sumativa, realizata prin verificarea practica finala de la laborator (N12) si examenul scris, final (nota N2).

b. Principii de notare

- Notarea activitatii din timpul semestrului: N1
  - Verificarea sistematica a programelor elaborate de studenti in timpul lucrarilor practice: N11
  - Verificarea finala lab.: N12
  - N1=0.3*N11+0.7*N12
- Proba de examinare ***(scris si oral)***: N2
  - Teorie – test grila *(scris)*- punctaj maxim 10 p, punctaj min promovare 5 p, nota N21
  - Probleme, proba practica *(oral)* – punctaj maxim 10 p, punctaj min promovare 5 p, nota N22
  - N2=0.5*N21+0.5*N22
- Nota finala: NF=0.25*N1 + 0.75*N2

  OBS:
  Se vor putea prezenta la examen doar studentii care au promovat disciplina A103.
  La proba practica a examenului final, vor putea participa doar acei strudenti care au promovat estul grila scris din cadrul examenului final.
  La examenul final, promoveaza doar studentii care la proba practica obtin min. 5 puncte.

c. Informarea studentilor cu privire la evaluarea asociata disciplinei
La primul curs, titularul prezinta studentilor modul in care vor fi evaluati si notati la disciplina respectiva.

9. Responsabil de curs

Nume : s.l.ing. STEFANESCU DIANA
Date de contact : Diana.Stefanescu@ugal.ro
Facultatea de Stiinta Calculatoarelor, Str. Domneasca 111, Corpul G, et. III, G 310, 800201 Galati, Tel./fax: +40 236 460182
Program de contact: luni 7:30-14, marti 8-19, miercuri, joi 8-14

Responsabil de curs, **Sef Departament / Catedra,**