

FIȘA DISCIPLINEI

Universitatea Academiei de Științe a Moldovei Facultatea Științe exacte			Denumirea cursului: Programare orientată spre obiect Codul cursului în planul de studii: S.06.A.043				
Nivelul calificării ISCED: 6 Domeniul de formare profesională: 443 Matematică Specialitatea: 443.1 Matematică			Catedra responsabilă de curs: Matematică și Informatică Titular/Responsabil de curs: Țițchiev Inga, dr., conf. univ.				
Total ore			Număr de ore pe tipuri de activități			Forma de evaluare	Număr de credite
total	contact direct	studiu individual	curs	seminar	laborator		
120	42	78	14	0	28	E	4
<p>Descrierea succintă a corelării cursului cu programul de studii</p> <p>Pregătirea specialiștilor în informatică, necesită formarea și dezvoltarea competenței de programare. Cea mai modernă paradigmă de programare este cea orientată obiect, care se impune ca instrument eficient în programarea microcontrolerelor, în proiectarea protocoalelor de comunicații și chiar în verificarea proiectării corecte a circuitelor în tehnicile VLSI. În acest fel, cunoașterea unui limbaj de programare orientat obiect susține și contribuie la crearea și dezvoltarea competenței mai sus menționate. Cursul de lecții <i>Programarea orientată pe obiect</i> este predestinat studenților ciclului I (universitar) cu scopul familiarizării lor cu caracteristicile limbajului C++ și mecanismele POO: încapsularea, moștenirea, supraîncărcarea operatorilor, polimorfismul.</p>							
<p>Competențe dezvoltate în cadrul cursului</p> <p>Competențe generale:</p> <ul style="list-style-type: none"> - cunoștințe de bază în domeniul matematicii elementare și a matematicii superioare; - capacitate de analiză și sinteză a teoriilor, metodelor și datelor provenite din diverse compartimente ale matematicii sau din surse adiacente; - capacitate de a aplica cunoștințele teoretice la studiul problemelor practice; - abilitate de a studia independent diverse surse și capacitate de a continua studiile cu un grad sporit de autoinstruire; - capacitate de aplicare a tehnologiilor informaționale și a cel puțin unei limbi străine în studiu și cercetare; - capacitatea de a lucra atât independent, cât și în echipă, în funcție de cerințele de moment. <p>Competențe specifice:</p> <ul style="list-style-type: none"> - dezvoltarea capacității de memorare, generalizare și analiză critică a informației care permite viitorului specialist să-și asume responsabilități și să se adapteze exigențelor societății; - cunoașterea și aplicarea metodologiei contemporane de cercetare, a ideilor, algoritmilor sau modelelor matematice în soluționarea problemelor practice; - argumentarea importanței investigațiilor reieșind din direcțiile de cercetare în domeniul matematicii teoretice și aplicative; - analiza metodelor de rezolvare a problemelor și a soluțiilor în concordanță cu datele din literatura de specialitate; - comunicarea logică, coerentă, argumentată a informației, ideilor, soluțiilor în limba de stat și alte limbi, în medii profesionale și alte medii social-economice; - identificarea domeniilor și argumentarea avantajelor pe care le oferă matematica în rezolvarea unor probleme profesionale, relevante pentru societate; - formarea abilităților de a însuși metode noi și mecanisme eficiente de soluționare a problemelor și de implementare a rezultatelor în activitatea profesională; - dezvoltarea capacității de a studia individual diverse surse informaționale care contribuie la sporirea gradului de autoinstruire în domeniul științei. 							
<p>Finalități de studii ale cursului</p> <p>la nivel de aplicare studenții vor:</p> <ul style="list-style-type: none"> - determina rolului și funcțiile POO în activitatea intelectuală, - realiza ierarhii de clase, activitate fundamentală în conceperea unui program OO, în vederea modelării într-un mod adecvat a unor sisteme complexe, - identifica probleme și elaborează scenariile posibile de soluționare, - interpreta și prezenta rezultatele execuției, - transfera cele însușite în cadrul altor domenii, 							

- demonstra utilitatea practică a cunoștințelor însușite,
- aplica cunoștințele acumulate în activitatea de instruire în cadrul altor discipline.

la nivel de integrare studenții vor:

- expune punctul de vedere și argumenta poziția proprie,
- utilizează diverse medii de dezvoltare integrate pentru programarea în C++,
- scrie în C++ aplicații de complexitate medie,
- utilizează bibliotecile de clase pentru a implementa aplicații complexe,
- planifică etapele rezolvării problemelor,
- aplică limbajul de specialitate în formularea răspunsurilor la întrebări,
- dezvoltă abilități sociale de interacțiune cu ceilalți,
- selectează și aplică cunoștințele acumulate în soluționarea problemelor,
- formulează scopuri specifice și strategii de îndeplinire a acestora,
- estimează calitatea produselor realizate.

Condiții prerechizit: pentru studierea cursului *Programare orientată pe obiect* este necesară inițierea în Analiza matematică, Teoria mulțimilor, Tehnologii informaționale și bazele programării, Algoritmi și programare, Logica matematică, Teoria probabilităților, Limbaje formale și automate, Algoritm și structuri de date.

Teme de bază: Introducere în programarea orientată obiect. Paradigme de programare: procedurală, modulară, abstractizarea datelor, programare orientată pe obiecte. Avantaje ale programării orientate obiect, comparații între diverse limbaje orientate obiect. Crearea de obiecte. Diferențe față de limbajul C, parcurgerea elementelor comune. Completări aduse de C++. Clase și obiecte (moduri de stocare a informației, constructori, metode). Incapsulare. Tipuri "class". Tipuri struct și union. Definierea claselor, folosirea unei clase, accesul la membrii unei clase. Modificatori de protecție: public, protected, private. Operatorul de rezoluție. Accesul la date și funcțiile membru. Folosirea membrilor dată a unei clase. Pointeri la membrii dată a unei clase. Autoreferința. Pointerul *this. Funcții inline. Crearea și distrugerea obiectelor (inițializare folosind constructori, inițializarea datelor membre). Destructorii. Funcții prietene și clase prietene ale unei clase. Exemple. Exploatarea claselor. Membrii statici ai unei clase. Modificatorul const.

Supraîncărcarea operatorilor. Operatori supraîncărcați ca funcții membru. Operatori supraîncărcați ca funcții friend. Restricții ale supraîncărcării operatorilor. Supraîncărcarea funcțiilor. Interfețe și clase incluse (moștenire simplă). Interfețe și clase incluse (moștenire multiplă). Compatibilitatea între o clasă derivată și clasa de bază. Conversii de tip. Constructori și destructori în clasele derivate. Drepturi de acces în clase derivate. Limitările legăturii statice. Pointeri către clasele derivate. Clasele derivate și supra definirea operatorilor. Clase virtuale. Funcții virtuale. Clase abstracte și polimorfism. Tratarea erorilor. Propagarea erorilor. Ierarhia claselor de erori. Alocarea dinamică a obiectelor. Tablouri de obiecte. Clase cu membri obiecte. Tablouri statice. Tablouri dinamice. Clase derivate, abstracte. Aplicații. Lista simplu înlănțuită. Clase derivate, abstracte. Aplicații. Lista dublu înlănțuită. Clase pentru desenarea în modul grafic. Fișiere, operații cu fișiere. Operații de formatare în memorie. Programarea generică. Clase și funcții șablon. Funcții template. Exemple. Necesitatea utilizării claselor și a funcțiilor șablon. Clase template. Exemple. Declararea și utilizarea funcțiilor șablon, claselor șablon. Biblioteca de șabloane standard (STL), containeri, iteratori, algoritmi.

Strategii de predare-învățare: prelegeri interactive, lucrări de laborator, lucrări individuale, platforma MOODLE, proiecte, consultații.

Strategii de evaluare: teste de evaluare, prezentări, rapoarte, dezbateri, elaborarea portofoliilor, teze/proiecte etc. Nota finală se constituie din rezultatul evaluării finale (40%), curente (40 %) și calității lucrului individual al studentului pe parcursul semestrului (20%).

Bibliografie selectivă:

1. Bereanu C., Programare orientata pe obiect, Editura SITECH, Craiova, 1999.
2. Braicov A., Gîncu S., Borland C++ Builder. Ghid de inițiere, Editura Tipografia Centrală, Chișinău, 2009.
3. Jurca I., Programarea orientata pe obiecte in limbajul C++, Ed. Eurobit, Timișoara, 1992.
4. Muslea I., C++ pentru avansați, Microinformatica, Cluj, 1994.
5. Muslea I., C++ Programare orientata pe obiecte, Microinformatica, Cluj, 1992.
6. Negrescu L., Limbajul C++, Microinformatica, Cluj, 1994.
7. Popovici D.M., Popovici I.M., Tănase I., C++ Tehnologie orientata pe obiecte. Aplicații, Ed.Teora, 1996.

Data

Semnătura