Ministerul Educației și Cercetării
Universitatea Valahia din Târgoviște
Facultatea de Inginerie Electrică, Electronică și Tehnologia Informației
Departamentul de Electronică, Telecomunicații și Inginerie Energetică

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Valahia din Târgoviște
1.2 Facultatea	Inginerie Electrică, Electronică și Tehnologia Informației
1.3 Departamentul	Electronică, Telecomunicații și Inginerie Energetică
1.4 Domeniul de studii	Inginerie Electronică, Telecomunicații și Tehnologii Informaționale
1.5 Ciclul de studii	Licență
1.6 Programul de studii	Tehnologii și Sisteme de Telecomunicații

2. Date despre disciplină

2.1 Denumirea disciplinei				**Programare obiect-orientată**			
2.2 Titularul activităților de curs				Conf. dr. ing. Mihai BÎZOI			
2.3 Titularul activităților de seminar				As. dr. ing. Marius Adrian Păun			
2.4 Anul de studiu	II	2.5 Semestrul	I	2.6 Tipul de evaluare	C	2.7 Regimul disciplinei	B F

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1 Număr de ore pe săptămână	4	din care: 3.2 curs	2	3.3 seminar/laborator	2
3.4 Total ore din planul de învățământ	56	din care: 3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp					ore
Studiul după manual, suport de curs, bibliografie și notițe					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					6
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri (inclusiv mini-proiectul de laborator)					12
Tutoriat					2
Examinări					2
Alte activități (pregătirea testului grilă Moodle)					2
3.7 Total ore studiu individual					44
3.9 Total ore pe semestru					100
3.10 Numărul de credite					4

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	• Programarea calculatoarelor și limbaje de programare • Structuri de date și algoritmi
4.2 de competențe	Descrierea funcționării unui sistem de calcul, a principiilor de bază ale arhitecturii microprocesoarelor și microcontrolerelor de uz general, a principiilor generale ale programării structurate.

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	Sală de curs cu videoproiector și calculator sau tablă interactivă. Suportul de curs, prezentările și testele sunt disponibile pe platforma Moodle; la nevoie, cursurile se pot desfășura online, pe platformele Moodle și Teams, conform procedurilor universității.

5.2 de desfășurare a seminarului/laboratorului	Sală de laborator cu calculatoare și software specific: mediul Code::Blocks cu compilatorul MinGW (standardul C++17) și biblioteca wxWidgets pentru interfața grafică. La nevoie, laboratoarele se pot desfășura online, pe platformele Moodle și Teams, conform procedurilor universității.

6. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

6.1 Obiectivul general al disciplinei	Formarea conceptelor de bază din programarea orientată pe obiecte. Însușirea deprinderilor de a utiliza instrumentele și tehnologiile de proiectare și realizare a aplicațiilor orientate pe obiecte.
6.2 Obiectivele specifice	• Înțelegerea principiilor programării orientate pe obiecte. • Cunoașterea sintaxei și a modului de supraîncărcare a operatorilor. • Tratarea erorilor prin excepții și gestiunea resurselor prin RAII și pointeri inteligenți. • Utilizarea programării generice și a bibliotecii standard C++ (containere și algoritmi). • Înțelegerea modului de realizare a aplicațiilor orientate pe obiecte într-un limbaj hibrid. • Proiectarea aplicațiilor orientate pe obiecte cu interfață grafică, modelate în UML și organizate după arhitectura MVC.

7. Rezultatele învățării

7.1 Cunoștințe *(Rezultatul asimilării de informații prin învățare. Cunoștințele reprezintă ansamblul de fapte, principii, teorii și practici legate de un anumit domeniu de muncă sau de studiu. Pot fi teoretice și/sau faptice)*

RiC1. Explică paradigmele de programare și conceptele fundamentale ale programării orientate pe obiecte, cu accent pe limbajul C++ și diferențele față de alte limbaje orientate pe obiecte.
RiC2. Descrie mecanismele de definire și utilizare a claselor și obiectelor, inclusiv mesaje, metode, constructori, destructori și principiile încapsulării, precum și mecanismul excepțiilor și gestiunea resurselor prin RAII și pointeri inteligenți.
RiC3. Explică moștenirea simplă și multiplă, polimorfismul, funcțiile virtuale și clasele abstracte, precum și implicațiile acestora în proiectarea aplicațiilor.
RiC4. Prezintă regulile de supraîncărcare a operatorilor, utilizarea funcțiilor și claselor prieten și conceptele de programare generică prin template-uri.
RiC5. Descrie mecanismele de intrare–ieșire în C++, principalele componente ale bibliotecii standard (STL), rolul diagramelor de clase UML și al arhitecturii Model–View–Controller (MVC) în dezvoltarea aplicațiilor orientate pe obiecte.

7.2 Aptitudini *(Capacitatea de a aplica cunoștințe și de a utiliza know-how pentru a duce la îndeplinire sarcini și a rezolva probleme. Aptitudinile sunt descrise ca fiind cognitive (implicând utilizarea gândirii logice, intuitive și creative) sau practice (implicând dexteritate manuală și utilizarea de metode, materiale, unelte și instrumente))*

RiA1. Proiectează și implementează aplicații orientate pe obiecte de complexitate redusă, utilizând clase, moștenire și polimorfism în C++.
RiA2. Utilizează corect constructori, destructori, supraîncărcarea operatorilor și mecanismele de copiere și atribuire; tratează erorile prin excepții și gestionează memoria dinamică fără pierderi de resurse, folosind pointeri inteligenți.
RiA3. Aplică funcții și clase template, precum și containere și algoritmi din biblioteca standard C++.
RiA4. Implementează mecanisme de intrare–ieșire pentru tipuri predefinite și definite de utilizator.
RiA5. Modelează aplicații simple utilizând diagrame de clase UML și le transpune în cod C++.
RiA6. Dezvoltă și evaluează aplicații orientate pe obiecte simple, cu interfață grafică, organizate după arhitectura MVC, în cadrul activităților de laborator.

7.3 Responsabilitate și autonomie *(Capacitatea cursantului de a aplica în mod autonom și responsabil cunoștințele și aptitudinile sale)*

RiR1. Rezolvă autonom sarcini de programare orientată pe obiecte, respectând cerințele tehnice și bunele practici de dezvoltare software.
RiR2. Utilizează responsabil resursele software, documentația tehnică și bibliotecile asociate limbajului C++.
RiR3. Colaborează eficient în activitățile de laborator și respectă normele etice și academice în realizarea aplicațiilor software.
RiR4. Manifestă interes pentru perfecționarea continuă a competențelor în domeniul programării orientate pe obiecte.

8. Conținuturi

8.1 Curs	Metode de predare	Observații
C1. Introducere în programarea orientată pe obiecte (Paradigme de programare; Evoluția limbajelor de programare orientate pe obiecte; Obiect, mesaje, metode; Comparație limbaj pur versus hibrid; C++ ca „C mai bun”: referințe, const, string, namespace, supraîncărcarea funcțiilor)	Expunerea, prezentarea, conversația	2 ore
C2. Clase și instanțe (Clase și instanțe; Specificatori de acces și încapsulare; Metode, pointerul this, metode const; Membri statici; Mesaje	Expunerea, prezentarea, conversația	2 ore

între obiecte; Evaluarea metodelor; Principiile programării orientate pe obiecte)		
C3. Constructori și destructori (Caracteristici și tipuri de constructori; Lista de inițializare; Destructori; RAII; Constructorul de copiere și operatorul de atribuire; Copiere superficială și profundă; Regula celor trei)	Expunerea, prezentarea, conversația	2 ore
C4. Moștenirea în limbajul C++ (Avantajele moștenirii; Sintaxa definirii claselor derivate; Accesul asupra membrilor moșteniți; Tipuri de moștenire; Constructori și destructori în procesul de moștenire; Moștenire versus compoziție)	Expunerea, prezentarea, conversația	2 ore
C5. Derivarea multiplă (Derivarea multiplă; Ambiguități de nume; Problema diamantului; Clase de bază virtuale; Constructorul de copiere și operatorul de atribuire în clasele derivate)	Expunerea, prezentarea, conversația	2 ore
C6. Polimorfismul (Ce este polimorfismul; Tipuri de polimorfism; Legare statică și dinamică; Funcții virtuale; Destructorul virtual; Mecanismul vtable; Clase abstracte și interfețe)	Expunerea, prezentarea, conversația	2 ore
C7. Excepții și gestiunea resurselor (throw, try, catch; Ierarhia std::exception; Derularea stivei; Excepții în constructori; Garanții de siguranță; Pointeri inteligenți; Regula lui zero)	Expunerea, prezentarea, conversația	2 ore
C8. Diagrame de clase UML. Arhitectura MVC (Diagrame UML; Reprezentarea claselor; Specificarea atributelor și metodelor; Relațiile dintre clase; Reutilizarea codului; Arhitectura Model–View–Controller)	Expunerea, prezentarea, conversația	2 ore
C9. Supraîncărcarea operatorilor. Funcții și clase prieten (Supraîncărcarea operatorilor: sintaxă și reguli; Operator membru versus funcție globală; Funcții prieten: globale și membre ale altor clase; Clase prieten)	Expunerea, prezentarea, conversația	2 ore
C10. Funcții și clase template (șablon) (Funcții template: sintaxă, deducerea argumentelor, specializare; Clase template: sintaxă, parametri non-tip; Instanțierea; Polimorfism parametric)	Expunerea, prezentarea, conversația	2 ore
C11. Sistemul de intrare-ieșire în C++ (Ierarhia fluxurilor; Funcții de I/O pentru tipurile predefinite și pentru tipuri definite de utilizator; Starea fluxurilor și validarea intrării; Formatarea I/O; Fișiere și fluxuri în memorie)	Expunerea, prezentarea, conversația	2 ore
C12. Biblioteca standard: containere (Containere, iteratori, algoritmi; Containere secvențiale; Containere asociative și neordonate; Adaptori; Complexitate și alegerea containerului)	Expunerea, prezentarea, conversația	2 ore
C13. Biblioteca standard: algoritmi (Intervale de iteratori; Funcții lambda; Algoritmi care modifică sau nu secvența; Sortare și căutare; Minim și maxim)	Expunerea, prezentarea, conversația	2 ore
C14. Dezvoltarea aplicațiilor orientate obiect. Recapitulare (Evoluția proceselor de dezvoltare software; Modelarea folosind cazuri de utilizare; De la cazuri de utilizare la clase; Scrierea codului; Subiecte posibile la verificarea scrisă)	Expunerea, prezentarea, conversația	2 ore

Bibliografie

1. Mihai Bîzoi, *Programare obiect-orientată – suport de curs și prezentări*, https://moodle.valahia.ro;
2. Bjarne Stroustrup, *A Tour of C++ (Third Edition)*, Pearson Education, 2023.
3. Bjarne Stroustrup, *Programming: Principles and Practice Using C++ (Third Edition)*, Pearson Education, 2024.
4. Marc Gregoire, *Professional C++ (Sixth Edition)*, John Wiley & Sons, 2024.
5. John Lakos, Vittorio Romeo, Rostislav Khlebnikov, Alisdair Meredith, *Embracing Modern C++ Safely*, Pearson Education, 2022.
6. J. Guy Davidson, Kate Gregory, *Beautiful C++*, Pearson Education, 2022.
7. Martin Fowler, *UML Distilled (Third Edition)*, Addison-Wesley, 2004.
8. Bjarne Stroustrup, Herb Sutter (ed.), *C++ Core Guidelines*, https://isocpp.github.io/CppCoreGuidelines.
9. *cppreference.com – C++ reference*, https://en.cppreference.com.

8.2 Seminar/laborator	Metode de predare	Observații
L1. Mediul de lucru. C++ ca „C mai bun” (Code::Blocks și MinGW, verificarea suportului C++17, depanarea pas cu pas).	Expunerea, demonstrația, lucrări practice	2 ore
L2. Clase și obiecte (Încapsulare și invarianți, metode const, membri statici, separarea în .h și .cpp).	Expunerea, demonstrația, lucrări practice	2 ore
L3. Constructori și destructori (Tipuri de constructori, lista de inițializare, RAII).	Expunerea, demonstrația, lucrări practice	2 ore
L4. Copierea obiectelor (Constructorul de copiere, operatorul de atribuire, regula celor trei).	Expunerea, demonstrația, lucrări practice	2 ore
L5. Derivarea (moștenirea) claselor.	Expunerea, demonstrația, lucrări practice	2 ore
L6. Derivarea multiplă (Problema diamantului, baze virtuale; copierea în clase derivate).	Expunerea, demonstrația, lucrări practice	2 ore
L7. Polimorfism și excepții (Funcții virtuale, clase abstracte, vector<unique_ptr>; tratarea excepțiilor).	Expunerea, demonstrația, lucrări practice	2 ore

L8. Interfețe grafice cu wxWidgets. Mini-proiect: fereastra principală a aplicației.	Expunerea, demonstrația, lucrări practice	2 ore
L9. Supraîncărcarea operatorilor. Funcții prieten. Mini-proiect: modelul și scheletul MVC.	Expunerea, demonstrația, lucrări practice	2 ore
L10. Funcții și clase template. Mini-proiect: istoricul măsurătorilor.	Expunerea, demonstrația, lucrări practice	2 ore
L11. Intrare-ieșire. Mini-proiect: salvarea și încărcarea datelor (CSV).	Expunerea, demonstrația, lucrări practice	2 ore
L12. Biblioteca standard: containere. Mini-proiect: modelul pe containere STL.	Expunerea, demonstrația, lucrări practice	2 ore
L13. Biblioteca standard: algoritmi. Mini-proiect: statistici, sortare, filtrare.	Expunerea, demonstrația, lucrări practice	2 ore
L14. Evaluare practică (o cerință nouă, implementată individual pe propriul mini-proiect).	Verificare practică	2 ore

Bibliografie

1. Bîzoi M., *Programare obiect-orientată – lucrări de laborator*, https://moodle.valahia.ro
2. Bjarne Stroustrup, *A Tour of C++ (Third Edition)*, Pearson Education, 2023.
3. Bjarne Stroustrup, *Programming: Principles and Practice Using C++ (Third Edition)*, Pearson Education, 2024.
4. Marc Gregoire, *Professional C++ (Sixth Edition)*, John Wiley & Sons, 2024.
5. *wxWidgets – Cross-Platform GUI Library, documentation*, https://docs.wxwidgets.org.

9. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

Conținuturile urmăresc practicile actuale din dezvoltarea software (C++17, biblioteca standard, gestiunea resurselor prin RAII, modelare UML, arhitectura MVC) cerute de întreprinderile și firmele de profil din zona Târgoviște și din județele limitrofe.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Testarea noțiunilor teoretice fundamentale	Examinare scrisă (lucrare scrisă sau test online)	50%
10.5 Seminar/laborator	Testarea noțiunilor teoretice fundamentale	Test grilă pe Moodle	20%
	Realizarea de aplicații funcționale; extinderea mini-proiectului MVC cu o cerință nouă	Verificare practică pe mini-proiect, după barem (în laborator sau pe MS Teams)	30%

10.6 Standard minim de performanță

- Efectuarea lucrărilor de laborator și realizarea mini-proiectului;
- Însușirea limbajului de specialitate;
- Cunoașterea tehnicilor de proiectare și realizare a aplicațiilor POO utilizând arhitectura MVC;
- Obținerea a cel puțin 50% din punctajul de la evaluare.

Data completării
15.09.2026

Titularul de curs
Conf. dr. ing. Mihai BÎZOI

Titularul de aplicații
As. dr. ing. Marius Adrian Păun

Data avizării în departament
23.09.2026

Director de departament
Conf. univ. dr. ing. Dan Constantin PUCHIANU

Data avizării în Consiliul Facultății
24.09.2026

Decan
Conf. univ. dr. ing. Nicoleta ANGELESCU