



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	Electronică aplicată

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	<ul style="list-style-type: none">Programarea calculatoarelor și limbaje de programareStructuri 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	Însușirea conceptelor, principiilor și tehnicilor programării orientate pe obiecte și dezvoltarea aptitudinilor de proiectare, implementare, testare și depanare a aplicațiilor software în limbajul C++, utilizând mecanismele programării orientate pe obiecte și bibliotecile software specifice.
6.2 Obiectivele specifice	<ul style="list-style-type: none"> - Cunoașterea principiilor programării orientate pe obiecte: abstractizare, încapsulare, moștenire și polimorfism. - Înțelegerea mecanismelor de definire și utilizare a claselor și obiectelor, constructorilor, destructorilor și gestionării resurselor în C++. - Aplicarea mecanismelor de moștenire, polimorfism, supraîncărcare a operatorilor, tratare a excepțiilor și programare generică. - Utilizarea bibliotecii standard C++ pentru gestionarea colecțiilor de date, prelucrarea informațiilor și realizarea operațiilor de intrare-ieșire. - Dezvoltarea aptitudinilor de modelare a aplicațiilor prin diagrame UML și organizarea codului conform arhitecturii Model-View-Controller. - Proiectarea, implementarea, testarea și depanarea unor aplicații software de complexitate medie, inclusiv cu interfață grafică, utilizând instrumente și biblioteci specifice.

7. Rezultatele învățării

7.1 Cunoștințe (<i>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</i>)	<ul style="list-style-type: none"> - C1. Studentul/absolventul descrie și explică principiile programării orientate pe obiecte, mecanismele de definire și utilizare a claselor și obiectelor, moștenirea, polimorfismul, gestionarea resurselor, programarea generică și utilizarea bibliotecilor standard C++ în dezvoltarea aplicațiilor software.
7.2 Aptitudini (<i>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)</i>)	<ul style="list-style-type: none"> - A1. Studentul/absolventul analizează cerințele unor probleme de programare, identifică soluții adecvate și evaluează structurile de clase, mecanismele și bibliotecile software necesare implementării acestora. - A2. Studentul/absolventul proiectează, implementează, testează și depunează aplicații orientate pe obiecte în limbajul C++, utilizând clase, moștenire, polimorfism, tratarea excepțiilor, programarea generică și biblioteca standard. - A3. Studentul/absolventul realizează aplicații software de complexitate mică și medie, utilizând modelarea UML, arhitectura MVC și biblioteci pentru interfețe grafice, verifică funcționalitatea acestora și interpretează rezultatele obținute.
7.3 Responsabilitate și autonomie (<i>Capacitatea cursantului de a aplica în mod autonom și responsabil cunoștințele și aptitudinile sale</i>)	<ul style="list-style-type: none"> - RA1. Studentul/absolventul manifestă un comportament profesional responsabil și etic în dezvoltarea aplicațiilor software, respectă cerințele de calitate, regulile de utilizare a resurselor informatice și principiile integrității academice.
7.4 Arii de învățare EUR-ACE	<ul style="list-style-type: none"> - CI – Cunoaștere și înțelegere: cunoașterea principiilor programării orientate pe obiecte, a mecanismelor limbajului C++ și a metodelor de structurare și dezvoltare a aplicațiilor software. - AI – Analiză inginerască: analiza cerințelor aplicațiilor, evaluarea alternativelor de implementare și identificarea structurilor de clase, algoritmilor și bibliotecilor software adecvate. - PrI – Practică inginerască: utilizarea mediilor de dezvoltare, a instrumentelor de compilare și depanare și a bibliotecilor software pentru implementarea, testarea și verificarea aplicațiilor 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; Comparatie 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 între obiecte; Evaluarea metodelor; Principiile programării orientate pe obiecte)	Expunerea, prezentarea, conversația	2 ore
C3. Constructori și destructori (Caracteristici și tipuri de constructori; Lista de inițializare; Destructorii; RAI; 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, <i>Programare obiect-orientată – suport de curs și prezentări</i> , https://moodle.valahia.ro ; 2. Bjarne Stroustrup, <i>A Tour of C++ (Third Edition)</i> , Pearson Education, 2023. 3. Bjarne Stroustrup, <i>Programming: Principles and Practice Using C++ (Third Edition)</i> , Pearson Education, 2024. 4. Marc Gregoire, <i>Professional C++ (Sixth Edition)</i> , John Wiley & Sons, 2024. 5. John Lakos, Vittorio Romeo, Rostislav Khlebnikov, Alisdair Meredith, <i>Embracing Modern C++ Safely</i> , Pearson Education, 2022. 6. J. Guy Davidson, Kate Gregory, <i>Beautiful C++</i> , Pearson Education, 2022. 7. Martin Fowler, <i>UML Distilled (Third Edition)</i> , Addison-Wesley, 2004. 8. Bjarne Stroustrup, Herb Sutter (ed.), <i>C++ Core Guidelines</i> , https://isocpp.github.io/CppCoreGuidelines . 9. <i>cppreference.com – C++ reference</i> , 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., <i>Programare obiect-orientată – lucrări de laborator</i> , https://moodle.valahia.ro 2. Bjarne Stroustrup, <i>A Tour of C++ (Third Edition)</i> , Pearson Education, 2023. 3. Bjarne Stroustrup, <i>Programming: Principles and Practice Using C++ (Third Edition)</i> , Pearson Education, 2024. 4. Marc Gregoire, <i>Professional C++ (Sixth Edition)</i> , 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 disciplinei sunt corelate cu cerințele pregătirii ingineresti în domeniul electronicii aplicate, oferind cunoștințe și aptitudini privind proiectarea și dezvoltarea aplicațiilor software utilizând principiile programării orientate pe obiecte. Studiul mecanismelor limbajului C++, al bibliotecilor standard și al metodelor de modelare și structurare a aplicațiilor contribuie la dezvoltarea capacității de realizare a soluțiilor software modulare, reutilizabile și fiabile. Activitățile de laborator, inclusiv realizarea unui mini-proiect cu interfață grafică, dezvoltă aptitudini de analiză a cerințelor, implementare, testare și depanare a aplicațiilor, relevante pentru dezvoltarea și integrarea componentelor software în sisteme electronice și aplicații ingineresti, în concordanță cu cerințele mediului profesional.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoașterea și explicarea principiilor programării orientate pe obiecte, a mecanismelor limbajului C++ și a metodelor de proiectare și structurare a aplicațiilor software. (C1, A1)	Examinare scrisă (lucrare scrisă sau test online)	50%
10.5 Seminar/laborator	Înțelegerea și aplicarea conceptelor de programare orientată pe obiecte; corectitudinea soluțiilor pentru probleme de programare și interpretarea rezultatelor. (C1, A1–A2)	Test grilă pe Moodle	20%
	Corectitudinea implementării și funcționării mini-proiectului, utilizarea mecanismelor orientate pe obiecte, organizarea codului, testarea, depanarea și justificarea soluției realizate. (A1–A3, RA1)	Verificare practică pe mini-proiect, după barem (în laborator sau pe MS Teams)	30%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> - Identificarea și explicarea conceptelor fundamentale ale programării orientate pe obiecte: clase, obiecte, încapsulare, moștenire și polimorfism. (C1) - Analizarea cerințelor unei probleme simple de programare și identificarea unei structuri de clase și a unei soluții de implementare adecvate. (A1) - Implementarea, compilarea, executarea și depanarea unei aplicații C++ care utilizează clase, metode și mecanisme fundamentale ale programării orientate pe obiecte. (A2) - Realizarea și verificarea funcționalității unui mini-proiect software, cu respectarea cerințelor de bază privind organizarea codului, corectitudinea implementării și utilizarea responsabilă a resurselor informatice. (A3, RA1) <p>Obține minimum 50% din punctajul total cumulat și minimum 50% din punctajul aferent fiecărei probe scrise.</p>			

Fișa disciplinei corespunde planului de învățământ care se aplică pentru anul I începând cu anul universitar 2025-2026, fiind valabilă pentru anul universitar 2026-2027

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