



Ministerul Educației și Cercetării al Republicii Moldova
I.P. Centrul de Excelență în Energetică și Electronică

"Aprob"

Mariana BARLADEAN,

Directoare I.P. Centrul de Excelență în Energetică și
Electronică,

"17" septembrie 2025



Curriculum la modulul
S.07.O.021 Limbaje de programare

*Programul de studii: 0714.2 – Automatizarea și robotizarea proceselor de
producție*

*Calificarea: 0714.2.1 – Tehnician/tehniciană în automatizarea
proceselor de producție*

Chișinău 2025

Curriculumul a fost elaborat în baza ordinului nr. 822, din data de 10.06.2024, elaborat de Ministerul Educației și Cercetării al Republicii Moldova, Cu privire la aprobarea listei programelor de formare profesională tehnică postsecundar și postsecundar nonterțiară pentru care urmează să fie elaborate/actualizate planurile de învățământ.



Aprobat la :

Consiliul metodic-științific al Centrului de Excelență în Energetică și Electronică din 15
septembrie 2025, director adjunct pentru instruire Virgil BANTAȘ _____
Ședința Catedrei "Automatizări" a I.P. Centrul de Excelență în Energetică și Electronică din
05 septembrie 2025, șef catedră Veaceslav CEAUȘ _____

Autori:

Sergiu TINCOVAN, profesor discipline de specialitate, grad didactic doi, I.P. Centrul de
Excelență în Energetică și Electronică

Veaceslav CEAUȘ, profesor discipline de specialitate, grad didactic superior, șef catedră
"Automatizări", I.P. Centrul de Excelență în Energetică și Electronică

Recenzenți:

1. Sergiu GAUGAȘ, director tehnic, Asociația „RENAM”
2. Denis ȚAPOTEI, metrolog-șef aparate, control, măsurări și automatizări, Fabrica SA „Bucuria”

Adresa Curriculumului în Internet:

<https://ceee.md/programe-de-formare-profesionala/>

Cuprins

<i>I. Preliminarii</i>	<i>4</i>
<i>II. Motivația, utilitatea modulului pentru dezvoltarea profesională.....</i>	<i>4</i>
<i>III. Finalități de învățare specifice modulului.....</i>	<i>5</i>
<i>IV. Administrarea modulului.....</i>	<i>5</i>
<i>V. Unitățile de învățare.....</i>	<i>5</i>
<i>VI. Repartizarea orientativă a orelor pe unități de învățare</i>	<i>8</i>
<i>VII. Studiu individual ghidat de profesor.....</i>	<i>8</i>
<i>VIII. Lucrările de laborator recomandate</i>	<i>10</i>
<i>IX. Sugestii metodologice.....</i>	<i>10</i>
<i>X. Sugestii de evaluare a rezultatelor învățării.....</i>	<i>12</i>
<i>XI. Resursele necesare pentru desfășurarea procesului de studii.....</i>	<i>13</i>
<i>XII. Resursele didactice recomandate elevilor.....</i>	<i>13</i>

I. Preliminarii

Curriculumul la modulul *Limbaje de programare* este elaborat în baza *Standardului de calificare 0714.2.1 Tehnician/tehniciană în automatizarea proceselor de producție* și planului de învățământ, aprobat de Ministerul Educației și Cercetării al Republicii Moldova nr. de înregistrare SC – 57/24 din 03 septembrie 2024.. Modulul *Limbaje de programare* este un element formativ al pachetului curricular pentru calificările din domeniul de formare profesională *Electronică și automată* și face parte din componenta de specialitate a planului de învățământ la programul de studii *0714.2 Automatizarea și robotizarea proceselor de producție*.

Scopul modulului *Limbaje de programare* este formarea la elevi a cunoștințelor și abilităților de utilizare a limbajului C în programarea microprocesoarelor. Limbajul C este ușor de utilizat cu interfețe de lucru explicite. Modulul *Limbaje de programare* este axat pe noțiuni generale ale limbajului C, pe funcțiile și pe instrucțiunile de utilizare ale acestuia.

Pentru studierea cu succes a acestui modul este necesar de parcurs, în prealabil, următoarele unități de curs:

G.04.O.002 Tehnologia informației;

F.06.O.015 Analiza și sinteza circuitelor numerice.

II. Motivația, utilitatea modulului pentru dezvoltarea profesională

Curriculumul la modulul *Limbaje de programare* este un document normativ și obligatoriu pentru realizarea procesului de pregătire a tehnicienilor în învățământul profesional tehnic postsecundar, care vor elabora diverse aplicații de programare în conformitate cu sarcinile de lucru.

Modulul *Limbaje de programare* contribuie la formarea competențelor profesionale ale tehnicianului în domeniul automatizării, necesare pentru îndeplinirea atribuțiilor și sarcinilor de lucru stabilite în calificarea profesională. Astfel, prezentul modul va contribui la:

- dobândirea cunoștințelor factice, principiilor, proceselor și conceptelor generale din domeniul elaborării produselor de program;
- formarea abilităților cognitive și practice necesare pentru elaborarea produselor program conform sarcinii;
- asumarea responsabilității pentru mentenanța de aplicații.

Rezultatele învățării atinse în cadrul acestui modul vor fi necesare pentru studierea unităților de curs orientate spre elaborarea/dezvoltarea produselor program, precum:

S.07.O.022 Proiectarea asistată de calculator;

S.08.O.023 Sisteme de reglare automată;

S.08.O.024 Microprocesoare și microcontrolere;

S.08.O.025 Programarea microcontrolerelor;

S.08.O.027 Automatizarea și robotizarea proceselor de producție în industrie;

P.07.O.051 Practica tehnologică;

P.07.O.052 Practica ce anticipează probele de absolvire.

III. Finalități de învățare specifice modului

Modulul *Limbaje de programare* va contribui la formarea rezultatelor învățării descrise în *Standardul de calificare Tehnician/tehniciană în automatizarea proceselor de producție*, care, la rândul lor, vor asigura formarea competențelor profesionale specificate în standardul ocupațional:

Rezultate ale învățării (RÎ) din standardul de calificare	Competențe profesionale (CP) din standardul ocupațional
RÎ7. Absolventul poate stabili ordinea de executare a acțiunilor de depanare a problemelor identificate, în funcție de nivelul de risc.	CP8. Algoritmizarea soluționării problemelor tehnice
RÎ8. Absolventul poate soluționa probleme tehnice ale obiectului/sistemului prin testări și autotestări repetate, utilizând diferite metode de verificare.	CP9. Soluționarea problemelor tehnice

Pentru a atinge rezultatele învățării nr. 7 și 8 din Standardul de calificare *Tehnician/tehniciană în automatizarea proceselor de producție*, după studierea modului *Limbaje de programare*, elevii vor fi capabili:

1. să identifice elementele de bază ale limbajului C;
2. să elaboreze programe elementare în limbajul C;
3. să prelucreze date structurate, utilizând instrucțiunile de decizie și ciclice în limbajul C;
4. să elaboreze programe elementare în limbajul specific mediului Arduino IDE.

IV. Administrarea modului

Semestrul	Numărul de ore					Forma de evaluare	Numărul de credite
	Total	Contact direct			Studiu individual		
		Teorie	Practică	Laborator			
VII	60	20	8	12	20	examen	2

V. Unitățile de învățare

Rezultate ale învățării	Unități de conținut	Abilități
1. Elementele de bază ale limbajului C		
RÎ1. Identificarea elementelor de bază ale limbajului C	1.1. Structura generală a unui program. Vocabularul limbajului C: - cuvinte-chele; - identificatori 1.2. Unități lexicale: - separatori	1. Identificarea părților componente ale unui program 2. Identificarea cuvintelor-cheie ale limbajului de programare 3. Scrierea identificatorilor 4. Scrierea simbolurilor de separare

	- numere - simboluri speciale - șiruri de caractere	5. Scrierea simbolurilor speciale în limbajul de programare 6. Alcătuirea șirurilor de caractere
2. Structura și componentele limbajului C		
RÎ2. Elaborarea programelor elementare în limbajul C	2.1. Structura limbajului C: - conceptul de date - tipuri de date (întregi; reale; caractere; booleene) - operatori 2.2. Componentele limbajului C: - variabile, constante (declarare, inițializare) - succesiunea operatorilor - operații de intrare la nivel de consolă sau fișier - operații de ieșire la nivel de consolă sau fișier - interfața mediului de programare - tipuri de instrucțiuni (vide; simple; compuse)	1. Diferențierea tipurilor de date 2. Identificarea operatorilor și operanzilor 3. Selectarea operatorilor în funcție de specificul tipului de date utilizat 4. Declararea variabilelor/constantelor conform tipului de dată solicitat 5. Inițializarea variabilelor/constantelor cu valori prestabilite 6. Scrierea expresiilor aritmetice în limbajul de programare C 7. Utilizarea operatorilor și operanzilor la prelucrarea datelor conform specificațiilor 8. Citirea datelor de intrare la nivel de consolă sau fișier 9. Afișarea datelor de ieșire la nivel de consolă sau fișier 10. Determinarea compatibilității componentelor instrucțiunilor
3. Structuri de control ale limbajului C		
RÎ3. Prelucrarea datelor structurate, utilizând instrucțiunile de decizie și ciclice în limbajul C	3.1. Structuri cu instrucțiuni de decizie: - instrucțiuni de decizie singulară - instrucțiuni de decizie multiplă 3.2. Structuri cu instrucțiuni ciclice și particularitățile lor: - structuri cu instrucțiunea ciclică FOR - structuri cu instrucțiunea ciclică WHILE - structuri cu instrucțiunea ciclică DO WHILE	1. Identificarea părților componente ale unei instrucțiuni de decizie 2. Elaborarea algoritmilor de decizie conform specificațiilor 3. Translarea algoritmilor cu decizie singulară sau cu decizie multiplă în limbajul de programare 4. Implementarea algoritmilor de decizie în limbajul de programare 5. Identificarea părților

	3.3. Tipuri de date structurate: - tablou unidimensional - tablou bidimensional - șir de caractere - funcții standard pentru prelucrarea acestora - funcții pentru conversii de date	componente ale unei instrucțiuni de ciclice 6. Elaborarea algoritmilor ciclici cu un număr cunoscut de pași 7. Elaborarea algoritmilor ciclici cu un număr necunoscut de pași 8. Translarea algoritmilor ciclici în limbajul de programare 9. Implementarea algoritmilor ciclici în limbajul de programare 10. Interpretarea rezultatelor unei aplicații de consolă 11. Testarea unei aplicații de consolă 12. Corectarea erorilor depistate în cadrul unei aplicații de consolă 13. Declararea tipurilor de date tablou unidimensional 14. Declararea tipurilor de date ale tablou bidimensional 15. Declararea tipurilor de date șir de caractere 16. Inițializarea elementelor unui tablou unidimensional, bidimensional, șir de caractere 17. Aplicarea instrucțiunilor de scriere/citire a elementelor tablourilor unidimensionale, bidimensionale, șirului de caractere 18. Aplicarea instrucțiunilor pentru prelucrarea datelor de tip tablou unidimensional, bidimensional, șir de caractere 19. Aplicarea instrucțiunilor pentru realizarea conversiei de date conform specificațiilor
4. Aplicații speciale ale limbajului pentru Arduino IDE		
RÎ4. Elaborarea programelor elementare în	4.1. Mediul de programare Arduino IDE: - lexicul și sintaxa mediului de	1. Descrierea lexicului și sintaxei programelor elaborate în mediul Arduino IDE

limbajul specific mediului Arduino IDE	programare Arduino IDE - funcții și operatori 4.2. Formarea intervalelor de timp cu ajutorul temporizatoarelor: - configurarea fuse-biților pentru temporizator - programarea ieșirilor temporizatoarelor în regim PWM 4.3. Regimurile de schimb al datelor: - regim digital de citire și înscrisere a datelor - regim analogic de citire și înscrisere a datelor	2. Aplicarea funcțiilor și operatorilor în mediul de programare Arduino IDE 3. Stabilirea valorilor fuse-biților 4. Setarea fuse-biților 5. Utilizarea regimului digital de citire și înscrisere a datelor pentru operațiile I/O 6. Utilizarea regimului analogic de citire și înscrisere a datelor pentru operațiile I/O
--	--	--

VI. Repartizarea orientativă a orelor pe unități de învățare

Nr. crt.	Unități de învățare	Numărul de ore				
		Total	Contact direct			Studiu individual
			Teorie	Practică	Laborator	
1.	Elementele de bază ale limbajului C	10	4	2	-	4
2.	Structura și componentele limbajului C	10	4	2	-	4
3.	Structuri de control ale limbajului C	22	6	2	8	6
4.	Aplicații speciale ale limbajului pentru Arduino IDE	18	6	2	4	6
	Total	60	20	8	12	20

VII. Studiu individual ghidat de profesor

Materii pentru studiul individual	Produse de elaborat	Modalități de evaluare	Număr de ore
1. Elementele de bază ale limbajului C			
1.1. Constante întregi și reale	Referat	Prezentarea referatului	2 ore
1.3. Vocabularul și sintaxa limbajului C	PPP cu exemple de aplicare	Derularea prezentării	2 ore
2. Structura și componentele limbajului C			

2.1. Utilizarea instrucțiunilor simple și a operatorilor de intrare la elaborarea unui program elementar în limbajul C	Prezentare PPP	Derularea prezentării	2 ore
2.1. Utilizarea instrucțiunilor simple și a operatorilor de ieșire la elaborarea unui program elementar în limbajul C	Prezentare PPP	Derularea prezentării	2 ore
3. Structuri de control ale limbajului C			
3.1. Operatori de decizie: - construcțiile de decizie de tip IF-ELSE - construcțiile de decizie de tip ELSE-IF - construcțiilor de decizie de tip SWITCH	Exerciții individuale la calculator	Prezentarea rezultatelor exercițiilor	2 ore
3.2. Operatori ciclici: - cicluri de tip FOR - cicluri de tip WHILE - cicluri de tip DO WHILE	Exerciții individuale la calculator	Prezentarea rezultatelor exercițiilor	2 ore
2.2. Tipuri de date structurate: - tipuri de date tablou unidimensional - tipuri de date tablou bidimensional - tipuri de date șir de caractere	Program pentru prelucrarea datelor structurate	Prezentarea programului și a rezultatelor	2 ore
4. Aplicații speciale ale limbajului pentru Arduino IDE			
5.1. Bazele programării Arduino IDE: - structura programului pentru Arduino IDE - instrucțiunile pentru Arduino IDE - tipurile de date și variabile - operatori - directive	Exerciții individuale la calculator	Prezentarea rezultatelor exercițiilor	2 ore
5.2. Construcțiile de decizie ale programului pentru Arduino IDE: - ciclul FOR - ciclul WHILE - ciclul DO WHILE	Exerciții individuale la calculator	Prezentarea rezultatelor exercițiilor	2 ore
5.3. Funcții și proceduri: - subrutine - funcții de convertire a tipului de date - funcții matematice - funcții I/O	Exerciții individuale la calculator	Prezentarea rezultatelor exercițiilor	2 ore

Notă: Rezultatele exercițiilor executate la calculator se includ în portofoliu.

VIII. Lucrările de laborator recomandate

Nr. d/o	Unități de învățare	Lista lucrărilor de laborator	Ore
1.	Elementele de bază ale limbajului C	Introducerea și extragerea datelor la nivel de consolă (LP)	2
2.	Structura și componentele limbajului C	Elaborarea algoritmilor de conversie a datelor întregi și reale din/în șir de caractere (LP)	2
3.	Structuri de control ale limbajului C	Elaborarea algoritmilor de decizie (LP)	2
		Implementarea algoritmilor de decizie singulară (LL)	2
		Implementarea algoritmilor de decizie multiplă (LL)	2
		Implementarea algoritmilor ciclici cu un număr cunoscut de pași (LL)	2
		Implementarea algoritmilor ciclici cu un număr necunoscut de pași (LL)	2
5.	Aplicații speciale ale limbajului pentru Arduino	Elaborarea programului pentru dispozitivul „Lumini alergătoare” (LP)	2
		Elaborarea programului pentru dispozitivul „Semafor compus” (LL)	2
		Elaborarea programului pentru dispozitivul „Lacăt electronic cu cod” (LL)	2

IX. Sugestii metodologice

Modulul *Limbaje de programare* este un curs de formare și dezvoltare a fundamentului competențelor profesionale ale viitorilor specialiști care trebuie să dețină cunoștințe și abilități de elaborare a programelor pentru procesele de automatizare.

Strategii didactice

Pentru a atinge rezultate ale învățării care să contribuie la dezvoltarea competențelor profesionale, strategiile didactice la modulul *Limbaje de programare* vor fi axate preponderent pe activizarea structurilor funcțional-acționare ale elevilor și pe exersarea potențialului psihofizic al acestora.

Metode didactice

Dat fiind caracterul practic al procesului de instruire, profesorul va selecta metode și tehnici didactice axate pe formarea aptitudinilor și deprinderilor praxiologice necesare. Un rol important la

formarea deprinderilor practice, dar și la dezvoltarea gândirii creative a elevilor în lucrările de laborator și lucrările practice, axate pe dezvoltarea următoarelor abilități:

- crearea algoritmului programului conform sarcinii;
- aplicarea mediilor de programare în limbajul C;
- alcătuirea textului inițial al programului în limbajul C;
- efectuarea compilării programului în limbajul C conform sarcinii;
- testarea și depanarea programului în limbajul C.

Pentru a atinge rezultatele învățării scontate, se recomandă aplicarea metodelor și tehnologiilor de predare care fac parte din:

- instruirea problematizată;
- instruirea euristică;
- instruirea demonstrativă;
- instruirea prin proiecte;
- simularea.

Ținând cont de specificul modulului dat, precum și în scopul formării competențelor profesionale, se recomandă aplicarea în procesul de predare-învățare nu numai a metodelor algoritmice (algoritmizarea și metoda exercițiului), dar și a unor strategii de tip euristic.

Algoritmizarea este o metodă de predare-învățare care constituie un ansamblu de operații executate într-o ordine relativ constantă, prin parcurgerea cărora poate fi rezolvat un set extins de probleme de același tip (gen, categorie).

Exercițiul presupune efectuarea unor operații mintale sau motrice în mod conștient și repetat, în vederea interiorizării unui model de acțiune și a formării sau îmbunătățirii unei priceperi. Aceasta este o metodă cu caracter algoritmic, în sensul că presupune repetarea identică a unor acțiuni până la formarea unei abilități. Cu toate acestea, exercițiul nu vizează doar formarea de abilități practice, ci poate fi aplicat și în domeniul teoretic, având ca scop: adâncirea înțelegerii unor noțiuni, reguli, principii și teorii, dezvoltarea operațiilor mintale și constituirea lor în structuri operaționale, sporirea capacității operatorii a cunoștințelor, dar și dezvoltarea unor trăsături morale, de voință de a atinge un scop.

Proiectul reprezintă o modalitate de instruire/autoinstruire grație căreia elevii efectuează o cercetare care este orientată spre obiective practice și este finalizată cu un produs, care poate fi: o schiță, o schemă, o prezentare, un text de program, o culegere tematică informațională despre părți componente, o culegere de imagini etc.

Procesul de studiu la unitatea de curs *Limbaje de programare* va deveni unul eficient dacă va fi însoțit de întocmirea portofoliului activităților individuale ale elevului. Se recomandă ca, de-a lungul instruirii la acest curs, elevii să țină un portofoliu, structurat pe două secțiuni. În partea I a portofoliului, elevul colectează și analizează informații, știri și inovații în domeniul tehnologiilor de programare.

În partea a II-a, elevul va reflecta asupra învățării prin efectuarea activităților practice și individuale propuse la fiecare unitate de învățare. Această secțiune a portofoliului ar trebui să reprezinte o colecție de activități pe care elevul le-a efectuat în cadrul orelor de activitate individuală pe tot parcursul semestrului.

X. Sugestii de evaluare a rezultatelor învățării

Noile direcții de dezvoltare a evaluării în pedagogia modernă reclamă transferul de la evaluarea axată pe inițiativa profesorului de a controla, la evaluarea centrată pe învățare, adică pe inițiativa elevului de a reflecta asupra propriului proces de cunoaștere, învățând din analiza și corectarea greșelilor. Strategiile de evaluare centrate pe învățare și formare se regăsesc în categoria strategiilor de evaluare interactivă care presupune existența unui parteneriat între profesor și elev, care are la bază procese de colaborare și care vizează responsabilizarea elevului în cadrul proceselor de învățare și evaluare.

Astfel, activitățile de evaluare vor fi orientate spre motivarea elevilor prin oferirea unui feedback continuu (permanent), fapt ce permite corectarea operativă a procesului de învățare, precum și prin evidențierea și accentuarea succeselor la fiecare etapă de învățare.

Metode de evaluare formativă recomandate:

- *observarea sistematică* a comportamentului elevilor, care permite evaluarea înțelegerii conceptelor, a capacităților, dar și a atitudinilor față de o sarcină dată;
- *autoevaluarea*, prin care elevul compară nivelul la care a ajuns cu obiectivele și standardele educaționale;
- *evaluarea reciprocă*, metodă care contribuie la responsabilizarea, conștientizarea, implicarea elevilor în evaluare, la socializarea grupului, la coparticiparea elevilor la formarea semenilor;
- *metoda lucrărilor de laborator*, prin care se evaluează materia însușită la lecțiile de teorie, utilizarea corespunzătoare a bibliografiei studiate, a materialelor și a instrumentarului, inclusiv a celor pentru programare.

Pentru a eficientiza procesele de evaluare, înainte de a începe evaluările, cadrul didactic va aduce la cunoștința elevilor tematica lucrărilor, modul de evaluare (bareme/grile/criterii de notare) și condițiile de realizare a fiecărei evaluări.

Evaluarea sumativă va fi realizată sub formă de examen la finele modulului. Scopul evaluării sumative este evidențierea *rezultatelor învățării*, și nu a procesului de învățare. Pentru a fi admis la examen, elevul:

- va prezenta portofoliul completat cu produsele studiului individual, prin care se evaluează experiența de învățare individuală a elevului;
- va efectua toate lucrările practice și de laborator, prin care se evaluează cunoștințele dobândite la lecțiile teoretice, precum și abilitățile practice de elaborare a programelor elementare în limbajul C.

Produsele recomandate pentru evaluarea rezultatelor învățării sunt:

- textul programului pentru algoritmul alcătuit;
- rezultatele compilării programului;
- rezultatele testării programului.

Evaluarea produselor elaborate de elev se va face în baza următoarelor criterii:

- Produsul corespunde specificațiilor prestabilite.
- Algoritmul programului este funcțional.
- Programul este compilat fără erori.
- Rezultatele testării programului corespund specificațiilor din sarcină.

Evaluarea proceselor executate de elev se va efectua în baza următoarelor criterii:

- completitudine;
- corectitudine;
- relevanță (comportament pe potrivă, adecvat);
- autonomie și responsabilitate;
- capacitatea de a acționa eficient în situații neprevăzute.

Pentru proba practică, se recomandă efectuarea unor operații cu ajutorul mediilor de programare, prin care elevii vor demonstra capacitățile lor de aplicare în practică a cunoștințelor dobândite. Probele practice răspund unui deziderat al pedagogiei moderne, și anume cel de „a ști să faci”.

XI. Resursele necesare pentru desfășurarea procesului de studii

Cerințe față de sălile de curs	
Pentru orele teoretice	Calculator, tablă interactivă sau proiector, conexiune la Internet
Pentru orele de laborator	Laborator de informatică cu 15-16 calculatoare, care asigură fiecărui elev un calculator, conexiune la Internet.
Cerințe tehnice	
Parametrii tehnici minimi ai calculatorului	Procesor: 2 GHz Memorie operativă: 4 GB Unitate de stocare: 500 GB Afișaj și grafică: size: 22", resolution: 1366 x 768 Network: Ethernet, 100 Mbps
Software	Sistem de operare Microsoft Windows și pachetul Office. Convertoarele de format al codului executabil, aplicația DOSbox, pachetul de programe Turbo C, Arduino IDE

XII. Resursele didactice recomandate elevilor

Nr. crt.	Denumirea resursei	Locul unde poate fi consultată, accesată sau procurată resursa	Numărul de exemplare disponibile
1.	Ștefan Buzurniuc, <i>Inițiere în limbajul C</i> , Chișinău, Evrica, 2004. ISBN 9975-942-71-7	Biblioteca	20
2.	Ян Белецкий, <i>Turbo Assembler. Версия 2.0. Учебное пособие для вузов</i> . Москва, 1994. ISBN 5-217-02037-7	Catedra (pe purtător electronic)	-

3.	Aplicații cu Microcontrolere / Gavril Todorean, Ovidiu Buza, Andras Balogh, Gabriel Oncea, Radu Voina, Vladimir Voicu, Alexandru Ricobon - Cluj-Napoca editura Risoprint, 2016	Format electronic Biblioteca IPCEEE	
4.	Valentina Tîrșu, <i>Limbajul C în cadrul disciplinei „Limba de programare”</i> . Suport de curs. Chișinău, CEP USM, 2014. ISBN 978-9975-71-556-0	Catedra (pe purtător electronic)	-
5.	Brian W. Kernighan, Dennis M. Richie, <i>Limbajul C</i> , București, Teora, 2003.	http://www.seap.usv.ro/~valeriu/lupu/Cap2.pdf	-
6.	Florian Moraru, <i>Programarea calculatoarelor în limbajul C</i> , 2008.	http://andrei.clubicisco.ro/cursuri/f/f-sym/1pc/2009CC/cursuri/Moraru-	-
7.	Cristian Marinoiu, <i>Programarea în limbajul C</i> . Ploiești, Editura Universității din Ploiești, 2000. ISBN 973-99506-8-x	Catedra (pe purtător electronic)	-
8.	В. В. Подбельский, <i>Программирование на языке C и учебное пособие для вузов</i> . Москва, Финансы и статистика, 2004.	Catedra (pe purtător electronic)	-
9.	Джон Хофман, <i>Освоение Arduino. Проектный подход к электронике, схемам и программированию</i> .	Suport didactic. Catedra (pe purtător electronic)	-
10.	Майкл МакРобертс, <i>Основы Arduino</i> . 2013.	Suport didactic. Catedra (pe purtător electronic)	-
11.	У. Соммер. <i>Программирование микроконтроллерных плат Arduino/Freduino</i> , пер. с нем. 2-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2017. ISBN 978-5-9775-3680-6	Catedra (pe purtător electronic)	-
12.	С. Монк. <i>Программируем Arduino. Профессиональная работа со скетчами</i> . Изд-во Питер, 2017. ISBN 978-5-496-02385-6	Catedra (pe purtător electronic)	-
13.	G. Vasilache, S. Gîncu, <i>Culegere de probleme la informatică</i> , Chișinău, 2012.	https://www.calameo.com/read/002801569a611d413be1c	-