

## ANEXA Nr. 4

### INFORMAȚII PUBLICE PRIVITOARE LA EXAMENELE DE PROMOVARE

Denumire câmp	Descriere
<b>Facultatea</b>	<b>Matematică și Informatică</b>
<b>Departamentul</b>	Informatică
<b>Poziția în statul de funcții</b>	Lector
<b>Funcția</b>	64
<b>Disciplinele din încărcătura postului</b>	Fundamentele programarii (în lb. română) Logica computationala (în lb. engleză) Structuri de date și algoritmi (în lb. engleză) Algoritmi fundamentali (în lb. engleză)
<b>Domeniul științific</b>	Informatică
<b>Descrierea postului</b>	<p>Lector, 64, Departamentul Informatică. Postul de lector universitar presupune desfășurarea de activități didactice, de cercetare științifică și de îndrumare a studenților, precum și efectuarea de servicii pentru comunitatea academică.</p> <p>La examenul de promovare poate participa doar cadrul didactic titular care a obținut la evaluările de către conducere, conform procedurilor operaționale de evaluare a activității profesionale, pe ultimii 3 ani în care s-a aflat în activitate, calificativul cel puțin „foarte bine” și care nu are o sancțiune disciplinară neradiată în condițiile legii.</p> <p>Pentru înscrierea la examenul de promovare în cariera didactică este necesară întrunirea condițiilor de vechime minimă în calitate de cadru didactic titular în învățământul superior în cadrul UBB după cum urmează:</p> <p>a) 3 ani pentru funcția de lector universitar/ șef de lucrări;</p> <p>Cerințe pentru candidați: Candidații la ocuparea postului vacant de lector universitar trebuie să dețină diploma de doctor în domeniul postului și să aibă palmaresul științific în concordanță cu standardele domeniului Informatică și cu disciplinele postului. De asemenea, candidații trebuie să facă dovada stăpânirii limbii engleze prin documente depuse la dosar (nivel C1 sau documente atestând studii sau stagii de cercetare cumulate de cel puțin 9 luni în străinătate, în instituții de învățământ sau cercetare unde comunicarea s-a făcut în limba engleză).</p>
<b>Atribuții</b>	<p>Activitatea didactică: seminar, laborator, proiecte, consultații, lucrări de control, examene, elaborarea de materiale didactice pentru disciplinele din post.</p> <p>Activitatea de cercetare științifică:</p> <ul style="list-style-type: none"> <li>• participarea la cel puțin un seminar de cercetare în cadrul facultății;</li> <li>• participarea la competiții pentru obținerea de granturi de cercetare</li> </ul>

	<p>științifică;</p> <ul style="list-style-type: none"> <li>• publicarea, în fiecare perioadă de 3 ani, a cel puțin 3 articole/studii indexate BDI (Mathematical Reviews/ MathSciNet, ZMath (Emis), Computing Reviews, IEEE Xplore, DOAJ, SCOPUS, DBLP);</li> <li>• îndeplinirea cerințelor din anexa la fisa postului, cel puțin la nivelul calificativului -Satisfăcător-.</li> </ul> <p>Activitatea de îndrumare a studenților: îndrumare de lucrări de diplomă, tutore la o formație de studenți, îndrumarea acestora pentru participarea la activitatea cercurilor științifice și la concursuri studentești.</p> <p>Servicii pentru comunitatea academică: participare la acțiunile desfășurate de departament, facultate și universitate (promovarea admiterii, colaborarea cu mediul economic, etc).</p>
<b>Data și ora susținerii prelegerii/ probei orale</b>	13.07.2026 ora 10.00
<b>Locul susținerii prelegerii/ probei orale</b>	Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala L404
<b>Probele de examen, data, ora și locul de susținere a acestora</b>	<p>Proba orală constă în prezentarea unui curs, în funcție de natura postului. Comisia de concurs stabilește titlul și îl anunță candidatului/candidaților cu 48 de ore înaintea prelegerii, fiind urmată obligatoriu de o sesiune de întrebări din partea comisiei și/sau a publicului.</p> <p>Durata minimă a probei orale susținute de către candidat este de 30 de minute; proba conține în mod obligatoriu și o sesiune de întrebări din partea comisiei și/ sau a publicului. Dacă sunt mai mulți candidați, comisia va decide ordinea în care aceștia vor susține proba orală.</p>
<b>Tematica și bibliografia probelor de examen</b>	<p><b>Proba 1 –Probă orală: susținerea unui curs</b></p> <p><b>Tematică:</b></p> <p><b>A. Fundamentele programarii (în lb. română)</b></p> <ol style="list-style-type: none"> <li>1. Introducere in procese de dezvoltare software : Ce este programarea - algoritmi, program, elemente de baza Python, interpretor Python, roluri în ingineria software ; Cum scriem programe - enunț problema , cerințe, proces de dezvoltare dirijat de functionalitati (FDD) ; Exemple - calculator</li> <li>2. Programare procedurală : Tipuri structurate - liste, tuple, dicționare ; Funcții - cazuri de testare, definiție, variabile, apel ; Transmiterea parametrilor ; Funcții anonime ; Cum scriem funcții - programare dirijată de teste, refactorizări</li> <li>3. Programare modulară : Ce este un modul - modul Python, domeniul variabilelor, pachete, module standard, distribuie module ; Cum organizăm codul sursă - responsabilități, single responsibility principle, separation of concerns, dependency, coupling, cohesion ; Arhitecturi software stratificate ; Eclipse+PyDev</li> <li>4. Tipuri definite de utilizator : Cum definim tipuri noi ; Incapsulare, ascunderea informației, tipuri abstracte de date</li> <li>5. Principii de proiectare și programare : Problema - program cu operații CRUD pe entități de un tip dat ; Arhitectura stratificată - UI, Domeniu, Infrastructura ; Sabloane GRASP ; Sabloane DDD - entity, validator,</li> </ol>

repository, controller ; Principii - Information Expert, Low Coupling, High Cohesion, Protected Variation, Single responsibility, Dependency Injection

6. Programare orientată pe obiecte : Obiecte si clase ; Diagrame UML; Mos tenere ; Exceptii
7. Proiectarea programelor : Top down and bottom up strategies ; Organizarea elementelor UI si relat ia cu alte straturi
8. Testarea si inspectarea programelor : Black box testing, white box testing ; Unit testing, integration testing ; Program inspection: coding style, refactoring
9. Recursivitate : Recursivitate directa si indirecta - Exemple Complexitatea algoritmilor ; Notat ia asimptotica - big-o, little-o, bigomega, little-omega, theta ; Comparat ii algoritmi
10. Algoritmi de căutare : ca utare secventiala ; ca utare binara Algoritmi de sortare ; BubbleSort ; SelectionSort ; InsertionSort ; QuickSort ; MergeSort ; Complexitatea algoritmilor
11. Backtracking : Algoritmul Backtracking ; Extensii ale algoritmului ; Exemple
12. Divide et Impera - Descriere Metoda - Exemple ; Greedy - Descriere Metoda ; Exemple
- 13 Programare dinamică - Descriere Metoda ; Exemple.

#### *Bibliografie*

1. Kent Beck. Test Driven Development: By Example. Addison-Wesley Longman, 2002. See also Test-driven development.  
[http://en.wikipedia.org/wiki/Test-driven\\_development](http://en.wikipedia.org/wiki/Test-driven_development)
2. Martin Fowler. Refactoring. Improving the Design of Existing Code. Addison-Wesley, 1999. See also <http://refactoring.com/catalog/index.html>
3. Frentiu, M., H.F. Pop, Serban G., Programming Fundamentals, Cluj University Press, 2006
4. The Python language reference.  
<http://docs.python.org/py3k/reference/index.html>
5. The Python standard library.  
<http://docs.python.org/py3k/library/index.html>
6. The Python tutorial. <http://docs.python.org/tutorial/index.html>

#### **B. Logica computatională (în lb. engleză)**

1. Baze de numerație:
  - 1.1. Definiții, reprezentare și operații (algoritmi de comparare, adunare, înmulțire, împărțire) cu numere într-o bază dată.
  - 1.2. Conversiile numerelor întregi și raționale între baze de numerație utilizând o bază intermediară.
  - 1.3. Conversii rapide: bazele 2,4,8,16.
2. Reprezentarea internă a numerelor
  - 2.1. Reprezentarea numerelor întregi fără semn, operații, noțiunea de depășire.
  - 2.2. Reprezentarea numerelor întregi cu semn folosind codurile: direct, invers și complementar, operații, depășire.

2.3. Reprezentarea numerelor reale: virgulă fixă, virgula mobilă.

3. Logica propozițiilor – sintaxa și semantica

3.1. Sintaxa: conective, formule.

3.2. Semantica: interpretarea unei formule, model, formulă consistentă (realizabilă), formulă inconsistentă (contradictorie), tautologie, relația de consecință logică. Tabela de adevăr a unei formule.

3.3. Echivalențe logice (legi): DeMorgan, absorbția, comutativitatea, asociativitatea, distributivitatea, idempotența.

3.4. Clauze și forme normale: forma normală conjunctivă (FNC) și forma normală disjunctivă (FND), algoritmul de aducere a unei formule la FNC și FND.

4. Logica propozițiilor – sistemul formal

4.1. Sistemul formal (axiomatic, deductiv) al logicii propoziționale, deducție, teoremă.

4.2. Teorema de deducție și consecințele sale.

4.3. Proprietățile logicii propozițiilor

5. Logica predicatelor de ordinul I

5.1. Sintaxa logicii predicatelor: conective, cuantificatori, termeni, atomi, formule, literal, clauze. Sistemul formal (axiomatic) asociat logicii predicatelor.

5.2. Semantica logicii predicatelor: interpretare, model, formulă validă, formulă consistentă, formulă contradictorie, relația de consecință logică.

5.3. Proprietățile logicii predicatelor.

6. Metoda tabelor semantice în logica propozițiilor și logica predicatelor

6.1. Clase de formule, reguli de descompunere a formulelor, ramură (deschisă, închisă), construcția tabelii semantice asociate unei formule.

6.2. Construire de modele și anti-modele pentru o formulă din tabelul semantice

7. Metoda rezoluției în logica propozițiilor

7.1. Sistemul formal al rezoluției.

7.2. Strategii ale rezoluției: strategia saturării pe nivele, strategia mulțimii suport, strategia.

7.3. Rafinări ale rezoluției: rezoluția blocării, rezoluția liniară.

8. Rezoluția în logica predicatelor

8.1. Forme normale ale formulelor predicative: premexa, Skolem și clauzala.

8.2. Substituții și unificatori.

8.3. Rezoluția predicativă – sistemul formal.

8.4. Strategii și rafinări ale rezoluției predicative.

9. Modelarea raționamentului matematic și uman folosind logica propozițiilor și cea a predicatelor.

10. Algebre booleene și funcții booleene.

10.1. Algebre booleene: definiții, proprietăți, principiul dualității, exemple.

10.2. Funcții booleene: definiții, maxtermi, mintermi, forma canonică conjunctivă, forma canonică disjunctivă.

10.3. Monoame maximale și centrale, factorizare.

11. Simplificarea funcțiilor booleene (I)

11.1. Metoda diagramelor Veitch-Karnaugh pentru funcții cu 2-3-4

variabile;

11.2. Algoritm de simplificare dual pentru forma canonica conjunctiva.

12. Simplificarea funcțiilor booleene (II)

12.1. Metoda analitică a lui Quine-Mc'Clusky

12.2. Metoda algebrică a lui Moisisil

13. Circuite logice

13.1. Definiții, reprezentarea circuitelor poartă de bază și derivate.

13.2. Exemple de circuite logice simple care intră în componenta hard a calculatoarelor: „decodorul” „circuitul comparator”, circuitul „sumatorul binar”

14. Circuite logice combinacionale-exemple

*Bibliografie:*

1. M. Ben-Ari: Mathematical Logic for Computer Science, Ed. Springer, 2001. 2. F. Boian, Bazele Matematice ale Calculatoarelor, Editura Presa Universitara Clujeana, 2002 – library.

3. M. Cocan, B. Pop: Bazele matematice ale sistemelor de calcul, Editura Albastra, Cluj-Napoca, 2001 – UBB library.

4. M.Fitting: First-order logic and Automated Theorem Proving, Ed.Springer Verlag, 1996.

5. M.Huth, M. Ryan: Logic in Computer Science - Modelling and Reasoning about Systems, 2nd edition, Cambridge, 2004.

6. D.W. Loveland: Automated Theorem Proving: a Logical Basis, North Holland, 2014.

7. M. Lupea, A. Mihis: Logici clasice și circuite logice. Teorie și exemple, ediția 3, Editura Albastra, Cluj-Napoca, 2011.

8. M. Lupea, A. Mihis: A Computational Approach to Classical Logics and Circuits, Editura Presa Universitara Clujeana, Cluj-Napoca, 2016.

9. Mihaela Malita, Mircea Malita, Bazele Inteligentei Artificiale, Vol. I, Logici propozitionale, Ed. Tehnica, Bucuresti, 1987.

10. L.C. Paulson: Logic and Proof, Univ. Cambridge, 2000, on-line course.

11. M. Possega: Deduction Systems, Inst. of Informatics, 2002, on-line course.

12. A.Wasilewska: Logics for Computer Science: Classical and Non-Classical, Springer, 2021, e-book.

### **C. Structuri de date si algoritmi (în lb. engleză)**

1. Structuri de date. Tipuri abstracte de date. Analiza algoritmilor: Tipuri abstracte de date și structuri de date ; Convenții de pseudocod ;Complexități

2. Vector. Iteratori: Vector dinamic ; Analiza complexității amortizate ; Interfața unui iterator

3. Liste înlanțuite: Lista simplu înlanțuită: reprezentare și operații ; Listă dublu înlanțuită: reprezentare și operații ; Iterator pentru liste înlanțuite

4. Tipuri abstracte de date: TAD Mulțime: descriere, domeniu, interfață și reprezentări posibile ; TAD Dicționar: descriere, domeniu, interfață și reprezentări posibile ; TAD Matrice: descriere, domeniu, interfață și

reprezentări posibile

5. Liste înlănțuite II: Liste înlănțuite ordonate: reprezentare și operații ; Liste înlănțuite pe tablou: reprezentare și operații

6. Tipuri abstracte de date II: TAD Lista (descriere, domeniu, interfață și reprezentări posibile) ; TAD Stivă (descriere, domeniu, interfață și reprezentări posibile) ; TAD Coadă (descriere, domeniu, interfață și reprezentări posibile)

7. Tabela de dispersie: Tabela de adresare directă ; Tabela de dispersie (descriere, proprietăți) ; Rezolvarea coliziunilor prin liste independente

8. Tabela de dispersie II: Rezolvarea coliziunilor prin liste întrepătrunse ; Rezolvarea coliziunilor prin adresare deschisă

9. Arbori. Arbori binari: Concepte legate de arbori ; Aplicații ale arborilor ; Reprezentări posibile ; Traversarea arborilor ; Descrierea și proprietățile arborilor binari ; Domeniul și interfața TAD Arbore binar

10. Arbori binari II: Reprezentări posibile ale TAD Arbore binar ; Traversarea arborelui binar (algoritmi recursivi/non-recursivi)

11. Ansamblu binar: Definiție, reprezentări, operații specifice ; HeapSort az

12. TAD Coadă cu priorități: Descriere, domeniu și interfață ; Reprezentări posibile ; Implementare pe ansamblu

13. Aplicații ale structurilor de date studiate

#### *Bibliografie:*

1. T. Cormen, C. Leiserson, R. Rivest, C. Stein: Introduction to algorithms, Third Edition, The MIT Press, 2009
2. Clifford A. Shaffer, A Practical Introduction to Data Structures and Algorithm Analysis, Third Edition, 2010
3. N. Karumanchi: Data structures and algorithms made easy, CareerMonk Publications, 2016
4. Narasimha Karumanchi, Data Structures and Algorithms Made Easy: Data Structures and Algorithmic Puzzles, Fifth Edition, 2016
5. M. A. Weiss: Data structures and algorithm analysis in Java, Third Edition, Pearson, 2012

#### **D. Algoritmi fundamentali (în lb. engleză)**

1. Introducere în procesele de dezvoltare software (Ce este programarea: algoritm, program, elemente de bază ale limbajului Python, interpretorul Python, roluri de bază în ingineria software ; Cum scriem programe: enunțul problemei, cerințe, procese de dezvoltare ghidate de funcționalități)

2. Programare procedurală (Tipuri de date: listă, tuplu, dicționar ; Funcții: cazuri de test, definiție, scopul variabilelor, transmiterea parametrilor ; Dezvoltare ghidată de testare, refactorizare)

3. Programare modulară (Ce este un modul: definirea unui modul Python, scopul variabilelor într-un modul, pachete, librării standard ; PyCharm)

4. Tipuri definite de utilizator (Cum definim tipuri noi de date: încapsulare, ascunderea datelor în Python ; Introducere în programarea orientată obiect ; Excepții)

5. Principii de proiectare software (Arhitectura stratificată: interfața

	<p>utilizator, aplicație, domeniu, infrastructură ; Cum organizăm codul sursă: responsabilități, principiul responsabilității unice, separarea grijilor, dependențe, cuplaj, coeziune)</p> <p>6. Programare orientată obiect (Tipuri abstracte de date ; Implementarea claselor în Python ; Obiecte și clase: clase, obiecte, atribute, metode, scop și namespace în Python )</p> <p>7. Aspecte legate de proiectare (Strategii de tip top-down și bottom-up ; Elemente de interfață utilizator (UI) )</p> <p>8. Testarea și inspecția programelor (Metode de testare: testare exhaustivă, testare de tip black box, testare de tip white box ; Testare automată, TDD ; Operații cu fișiere în Python)</p> <p>9. Recursivitate (Noțiunea de recursivitate ; Recursivitate directă și indirectă ; Exemple)</p> <p>10. Complexitatea algoritmilor (Algoritmi de căutare - Definirea problemei ; Metode de căutare: secvențială, binară ; Complexitatea algoritmilor ; Algoritmi de sortare - Definirea problemei ; Metode de sortare: Bubble Sort, Selection Sort, Insertion Sort, Quick Sort ; Complexitatea algoritmilor )</p> <p>11. Backtracking (Prezentarea generală a metodei ; Complexitatea algoritmilor ; Exemple )</p> <p>12. Divide et impera, Greedy (Prezentarea generală a metodei ; Complexitatea algoritmilor ; Exemple )</p> <p>13. Programare dinamică (Descrierea metodei ; Exemple)</p> <p><i>Bibliografie</i></p> <p>1. M. Frentiu, H.F. Pop, Fundamentals of Programming, Cluj University Press, 2006.</p> <p>2.K.Beck,TestDrivenDevelopment:ByExample.Addison-WesleyLongman,2002. <a href="http://en.wikipedia.org/wiki/Test-driven_development">http://en.wikipedia.org/wiki/Test-driven_development</a></p> <p>3.M.Fowler,Refactoring.ImprovingtheDesignofExistingCode,Addison-Wesley,1999. <a href="http://refactoring.com/catalog/index.html">http://refactoring.com/catalog/index.html</a></p> <p>4.ThePythonProgrammingLanguage-<a href="https://www.python.org/">https://www.python.org/</a></p> <p>5.ThePythonStandardLibrary-<a href="https://docs.python.org/3/library/index.html">https://docs.python.org/3/library/index.html</a></p> <p>6.ThePythonTutorial-<a href="https://docs.python.org/3/tutorial/">https://docs.python.org/3/tutorial/</a></p>
<p><b>Descrierea procedurii de examen</b></p>	<p>Comisia de concurs evaluează candidații ținând cont de următoarele aspecte:</p> <ul style="list-style-type: none"> <li>• Conținutul dosarului individual (pondere 80%);</li> <li>• probă orală publică (susținerea unui curs), cu tema stabilită în condițiile art. 17, alin. (1), lit. b) din metodologia de promovare de către comisia de examen (pondere 20%)</li> </ul> <p>În cazul tuturor posturilor durata minimă a probei orale/ prelegerii susținute de către candidat este de 30 de minute; proba conține în mod obligatoriu și o sesiune de întrebări din partea comisiei și/ sau a publicului.</p> <p>În cazul unui post cu încărcătură integrală într-o singură limbă străină,</p>

	<p>proba de examen va fi susținută în limba respectivă în fața comisiei de examen; în cazul unui post cu încărcătură în mai multe limbi străine, prin anunțul de examen va fi stabilit modul în care proba orală/ prelegerea va fi susținută în limbile respective.</p> <p>Nota finală a fiecărui candidat se calculează ca medie aritmetică a notelor obținute la criteriile de mai sus. Fiecare membru al comisiei (inclusiv președintele) întocmește un referat individual de apreciere care propune o notă finală pentru fiecare candidat.</p> <p>Pentru a se califica în vederea ocupării postului, candidatul trebuie să fi obținut la fiecare probă cel puțin nota 7,00 și să obțină media generală în raportul de sinteză asupra examenului cel puțin 8,50.</p> <p>Președintele comisiei de concurs întocmește un raport asupra concursului în care prezintă notele finale atribuite candidaților de către membrii comisiei și indică media generală obținută de fiecare candidat, calculată ca medie aritmetică a notelor finale din referatele individuale. Media generală astfel obținută reprezintă rezultatul concursului pentru fiecare candidat. Pe baza mediei generale, comisia de concurs decide ierarhia candidaților și nominalizează candidatul eligibil care a întrunit cel mai bun rezultat în concurs. Președintele comisiei de concurs supune raportul de sinteză asupra concursului votului deschis al membrilor comisiei. În urma exercitării votului, președintele constată rezultatul votului, îl comunică membrilor comisiei și îl menționează în încheierea raportului asupra concursului, cu precizarea numărului de voturi "pentru", respectiv "contra". În cazul în care votul "pentru" nu este acordat de majoritatea membrilor comisiei, postul scos la concurs nu este ocupat de niciun candidat. Ierarhia candidaților stabilită prin media generală nu poate fi modificată prin votul comisiei. Raportul asupra concursului este semnat de fiecare dintre membrii comisiei de concurs și de către președintele comisiei.</p>
<b>Perioada de comunicare a rezultatelor</b>	<i>În aceeași zi cu încheierea lucrărilor comisiei.</i>
<b>Perioada de depunere a contestațiilor</b>	<i>În termen de 3 zile lucrătoare de la comunicarea rezultatului evaluării la probele de examen.</i>
<b>Salariul minim de încadrare a postului la momentul angajării</b>	<i>Salariul minim de bază: lector univ. 7699 lei</i>

**Director Departament,  
Conf. dr. Adrian STERCA**

