

INFORMAȚII PUBLICE PRIVITOARE LA CONCURSURI

Denumire câmp	Descriere
Facultatea	Matematică și Informatică
Departamentul	Informatică
Poziția în statul de funcții	100
Funcția	Asistent universitar (pe perioadă determinată 3 ani)
Disciplinele din încărcătura postului/ ariile de cercetare, așa cum figurează în statul de funcții	Algoritmi și programare (lb. engleză) Fundamentele programării (lb. engleză) Programare orientată obiect (lb. engleză) Structuri de date și algoritmi (lb. engleză)
Domeniul științific	Informatică
Descrierea postului scos la concurs	<p>Asistent, 100, Departamentul de Informatică. Postul de asistent universitar presupune desfășurarea de activități didactice, de cercetare științifică și de îndrumare a studenților, precum și efectuarea de servicii pentru comunitatea academică.</p> <p>Candidații la ocuparea postului vacant de asistent universitar trebuie 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>
Atribuții	<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"> • participarea la cel puțin un seminar de cercetare în cadrul facultății; • participarea la competiții pentru obținerea de granturi de cercetare științifică; • publicarea, în fiecare perioadă de 4 ani, a cel puțin 5 articole/studii indexate BDI (Mathematical Reviews/ MathSciNet, ZMath (Emis), Computing Reviews, IEEE Xplore, DOAJ, SCOPUS, DBLP) din care cel puțin 2 indexate ISI sau SCOPUS sau în volumele unor conferințe internaționale relevante (ACM, IEEE, AMS, EMS). <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>

Data și ora susținerii probei orale	28.01.2020, ora 12:00
Locul susținerii probei orale	Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala C335
Probele de concurs, data, ora și locul de susținere a acestora	<p>Pentru postul de ASISTENT UNIVERSITAR, concursul constă în:</p> <ol style="list-style-type: none"> 1. evaluarea dosarului individual; 2. susținerea unei probe orale; 3. susținerea unei probe scrise. <p>Proba orală constă în prezentarea unui proiect de seminar/ laborator/ lucrări practice. Comisia stabilește, pe baza tematicii și bibliografiei de concurs, tema prezentării probei orale și o comunică candidaților cu 48 de ore înaintea susținerii probei prin email și prin afișarea la avizierul și pagina web a facultății, cu menționarea datei și orei afișării, sub semnătura președintelui comisiei de concurs. 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;</p> <p>Proba 1 – Probă scrisă (limba engleză): 28.01.2020, ora 8:00, Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala C335.</p> <p>Proba 2 – Probă orală (limba română): susținerea unui proiect de seminar/ laborator/ lucrări practice – 28.01.2020, ora 12:00, Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala C335. Dacă sunt mai mulți candidați, comisia va decide ordinea în care aceștia vor susține proba orală.</p> <p>Dosarul individual, proba orală și proba scrisă contează în proporții egale la nota finală acordată în referatul individual de apreciere întocmit de fiecare membru al comisiei de concurs;</p>
Tematica și bibliografia probelor de concurs	<p>Proba 1 – Probă scrisă</p> <p>Tematică:</p> <p>Fundamentele programării, Programare orientată pe obiecte, Structuri de date</p> <ol style="list-style-type: none"> 1. Fundamentele programării <p>Subalgoritmi: specificare, testare. Clase de algoritmi: căutare, sortare, interclasare. Metode de proiectare a algoritmilor: top-down, rafinare succesivă. Subprograme, apel și modalități de transmitere a parametrilor (prin valoare și referință). Tehnici de programare: Backtracking, Divide et impera, Greedy. Programare modulară: modul, interfață, implementare; concretizare în C/C++, Java, Python.</p> <ol style="list-style-type: none"> 2. Programare orientată pe obiecte

Clase, obiecte.
Moștenire, polimorfism.
Programare bazată pe interfețe.
3. Structuri de date
Tipuri abstracte de date (TAD).
TAD Mulțime, Colecție, Listă, Stivă, Coadă, Dicționar.
Specificarea TAD.
Implementări pentru TAD folosind: vectori, liste înlănțuite, arbori binari.

Bibliografie:

1. M. Frențiu, B. Pârv, Elaborarea programelor. Metode și tehnici moderne, ProMedia, Cluj-Napoca, 1994
2. M. Frențiu, H.F. Pop, G. Șerban, Programming fundamentals, Cluj University Press, 2006
3. T. Cormen, C. Leiserson, R. Rivest: Introducere în algoritmi. Cluj-Napoca: Editura Computer Libris Agora, 2000
4. B. Eckel, Thinking in C++, vol I și II, <http://www.mindview.net>
5. B. Eckel, Thinking in Java, <http://www.mindview.net>
6. M.A. Ellis, B. Stroustrup, The annotated C++ reference manual, Addison-Wesley, 1994
7. The Python language reference.
<http://docs.python.org/py3k/reference/index.html>
8. R.S. Pressman, Software engineering. A practitioner's approach, 6th ed., McGraw-Hill, 2005

Proba 2 – Probă orală: susținerea unui proiect de seminar/
laborator/ lucrări practice

Tematică:

A. Algoritmi și programare

1. Introducere în procese de dezvoltare software
2. Programare procedurală
3. Programare modulară
4. Tipuri definite de utilizator
5. Principii de proiectare și programare
6. Programare orientată pe obiecte
7. Proiectarea programelor
8. Testarea și inspectarea programelor
9. Recursivitate
10. Complexitatea algoritmilor
11. Algoritmi de căutare și sortare
12. Metode de rezolvare a problemelor (I) – Backtracking, Greedy
13. Metode de rezolvare a problemelor (II) - Divide & Conquer, Programare dinamică

Bibliografie:

1. M.L. Hetland, Beginning Python: From Novice to Professional, Apress, 2005.
2. M. Frențiu, H.F. Pop, Fundamentals of Programming, Cluj University Press, 2006.

3. K. Beck, Test Driven Development: By Example. Addison-Wesley Longman, 2002.

http://en.wikipedia.org/wiki/Test-driven_development

4. M. Fowler, Refactoring. Improving the Design of Existing Code, Addison-Wesley, 1999.

<http://refactoring.com/catalog/index.html>

5. The Python Programming Language - <https://www.python.org/>

6. The Python Standard Library -

<https://docs.python.org/3/library/index.html>

7. The Python Tutorial - <https://docs.python.org/3/tutorial/>

B. Fundamentele programarii

1. Introducere in procese de dezvoltare software

2. Programare procedurala

3. Programare modulara

4. Tipuri definite de utilizator

5. Principii de proiectare si programare

6. Programare orientata pe obiecte

7. Proiectarea programelor

8. Testarea si inspectarea programelor

9. Recursivitate

10. Complexitatea algoritmilor

11. Algoritmi de cautare și sortare

12. Metode de rezolvare a problemelor (I) – Backtracking, Greedy

13. Metode de rezolvare a problemelor (II) - Divide & Conquer, Programare dinamica

Bibliografie:

1. Kent Beck - Test Driven Development: By Example. Addison-Wesley Longman, 2002.

2. Kleinberg and Tardos – Algorithm Design. Pearson Educational, 2014

(<http://www.cs.princeton.edu/~wayne/kleinberg-tardos/>)

3. Martin Fowler - Refactoring. Improving the Design of Existing Code. Addison-Wesley, 1999.

(<http://refactoring.com/catalog/index.html>)

4. Frentiu, M., H.F. Pop, Serban G. - Programming Fundamentals, Cluj University Press, 2006

5. The Python language reference.

(<https://docs.python.org/3/reference/index.html>)

6. The Python standard library.

(<https://docs.python.org/3/library/index.html>)

7. The Python tutorial.

(<https://docs.python.org/3/tutorial/index.html>)

C. Programare orientata obiect

1. Elemente de bază ale limbajului C

2. Programare modulară în C/C++

3. Programare orientată obiect în C++.

4. Tipuri de date derivate și tipuri definite de utilizator, alocare dinamică în C++. Elemente de programare generica

5. Moștenire

6. Polimorfism

	<p>8. Ierarhii de clase</p> <p>9. Interfețe grafice utilizator (GUI)</p> <p>10. Elemente de programare bazată pe evenimente (Evenimente: Semnale si sloturi Qt; Proiectare GUI; Callback/Observer)</p> <p>11. Elemente de programare bazată pe evenimente (Componente grafice cu modele; Șablonul MVC; Modele predefinite)</p> <p>12. Șabloane de proiectare</p> <p>Bibliografie:</p> <ol style="list-style-type: none"> 1. B. Stroustrup. The C++ Programming Language, Addison Wesley, 1998. 2. Bruce Eckel. Thinking in C++, Prentice Hall, 1995. 3. A. Alexandrescu. Programarea moderna in C++: Programare generica si modele de proiectare aplicate, Editura Teora, 2002. 4. S. Meyers. Effective C++: 55 Specific Ways to Improve Your Programs and Designs (3rd Edition), Addison-Wesley, 2005. 5. S. Meyers. More effective C++: 35 New Ways to Improve Your Programs and Designs, Addison-Wesley, 1995. 6. B. Stroustrup. A Tour of C++, Addison Wesley, 2013. 7. C++ reference (http://en.cppreference.com/w/). 8. Qt Documentation (http://doc.qt.io/qt-5/). 9. E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Longman Publishing, 1995. <p>D. Structuri de date si algoritmi</p> <ol style="list-style-type: none"> 1. Introducere. Structuri de date. Structuri statice, semistatice si dinamice. 2. Tipuri de date: domeniu, operatii si reprezentarea datelor 3. Tabloul 4. Abstract Data Types - TAD Colectie, TAD Dictionar, TAD Stiva si Coadă, TAD Coadă cu prioritati 5. Linked Lists - TAD Lista, Lista înlantuita 6. Ansamblul (heap) 7. Hash Table - Tabela de dispersie 8. TAD Arbore - Arborele binar 9. Arborele binar de cautare 10. Arbori binari de cautare echilibrati 11. aplicații și structuri de date în diferite limbaje de programare (Python, C++, Java, C#) <p>Bibliografie:</p> <ol style="list-style-type: none"> 1. T. Cormen, C. Leiserson, R. Rivest, C. Stein: Introduction to algorithms, Third Edition, The MIT Press, 2009 2. S. Skiena: The algorithms design manual, Second Edition, Springer, 2008 3. N. Karumanchi: Data structures and algorithms made easy, CareerMonk Publications, 2016 4. M. A. Weiss: Data structures and algorithm analysis in Java, Third Edition, Pearson, 2012 5. R. Sedgewick: Algorithms, Addison-Wesley Publishing, 1984
Descrierea procedurii de concurs	Comisia de concurs evaluează candidații ținând cont de următoarele criterii:

- Conținutul dosarului individual;
- Proba orală (proba 1)
- Proba scrisă (proba 2).

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.

Candidații eligibili pentru ocuparea postului scos la concurs trebuie să obțină:

- cel puțin nota 6 (șase) la fiecare criteriu;
- nota finală cel puțin 7 (șapte) dată de fiecare referent;
- media generală cel puțin 8,50 (opt și 50%).

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 asupra concursului votului secret al membrilor comisiei. În urma exercitării votului secret, președintele constată rezultatul votului, îl comunică membrilor comisiei și îl menționează în încheierea raportului asupra concursului, cu precizarea numărului de voturi "pentru", respectiv "contra", votul fiind menținut secret. În cazul în care votul "pentru" nu este acordat de majoritatea membrilor comisiei, postul scos la concurs nu este ocupat de niciun candidat. Raportul asupra concursului este semnat de fiecare dintre membrii comisiei de concurs și de către președintele comisiei.

Director departament,

Prof. univ. dr. Laura DIOȘAN

