

INFORMAȚII PUBLICE PRIVITOARE LA CONCURSURI

Denumire câmp	Descriere
Facultatea	Matematică și Informatică
Departamentul	Informatică
Poziția în statul de funcții	99
Funcția	Asistent universitar (pe perioadă determinată 3 ani)
Disciplinele din încărcătura postului/ ariile de cercetare, așa cum figurează în statul de funcții	Algoritmi si programare (lb. engleză) Programare orientata obiect (lb. engleză) Arhitectura sistemelor de calcul soft (lb. engleză) Arhitectura sistemelor de calcul soft (lb. română) Limbaje formale si tehnici de compilare (lb. română) Limbaje formale si tehnici de compilare (lb. engleză)
Domeniul științific	Informatică
Descrierea postului scos la concurs	Asistent, 99, 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. 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).
Atribuții	Activitatea didactică: seminar, laborator, proiecte, consultații, lucrări de control, examene, elaborarea de materiale didactice pentru disciplinele din post. Activitatea de cercetare științifică: • 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). 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 studențești. Servicii pentru comunitatea academică: participare la acțiunile

	desfășurate de departament, facultate și universitate (promovarea admiterii, colaborarea cu mediul economic, etc).
Data și ora susținerii probei orale	28.01.2020, ora 11:00
Locul susținerii probei orale	Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala C310
Probele de concurs, data, ora și locul de susținere a acestora	Pentru postul de ASISTENT UNIVERSITAR, concursul constă în: 1. evaluarea dosarului individual; 2. susținerea unei probe orale; 3. susținerea unei probe scrise. 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; Proba 1 – Probă scrisă (limba engleză): 28.01.2020, ora 8:00, Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala C335. Proba 2 – Probă orală (limba română): susținerea unui proiect de seminar/ laborator/ lucrări practice – 28.01.2020, ora 11:00, Departamentul de Informatică, str. Teodor Mihali nr. 58-60, sala C310. Dacă sunt mai mulți candidați, comisia va decide ordinea în care aceștia vor susține proba orală. 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;
Tematica și bibliografia probelor de concurs	Proba 1 – Probă scrisă Tematică: Fundamentele programării, Programare orientată pe obiecte, Structuri de date 1. Fundamentele programării 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. 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, Dictionar. 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. 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
8. Ierarhii de clase
9. Interfețe grafice utilizator (GUI)
10. Elemente de programare bazată pe evenimente (Evenimente: Semnale și sloturi Qt; Proiectare GUI; Callback/Observer)
11. Elemente de programare bazată pe evenimente (Componente grafice cu modele; Șablonul MVC; Modele predefinite)
12. Șabloane de proiectare

Bibliografie:

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.

C. Arhitectura sistemelor de calcul soft

1. Reprezentarea datelor: date elementare, reprezentari binare si ordini de plasare, organizarea si memorarea datelor
2. Codificarea caracterelor, reprezentarea cu semn si fara semn, cod complementar, conversii, conceptul de depasire
3. Arhitectura sistemelor de calcul: organizarea unui SC, unitatea centrala, ceasul sistem, calculator pe n biti, memoria, dispozitivele periferice

4. Performantele unui SC, arhitectura microprocesorului 80x86 – structura, registri, calculul de adresa, moduri de adresare, adrese far si near
5. Unitatea executiva (EU) a microprocesorului 80x86: rolul si functiile registrilor si al flagurilor. Clasificare (Registrii si Flaguri) si studii de caz.
6. Unitatea BIU a microprocesorului 80x86: registrii de adresa, registrii de segment, reprezentarea instructiunilor. Formula de specificare offset pe 32 biți si formula de specificare offset pe 16 biți.
7. Elementele limbajului de asamblare: formatul unei linii sursa, expresii, tipuri de accesare a operanzilor, operatori. Conversii nondistructive (si operatorii specifici)
8. Directive pentru definirea segmentelor, pt.definirea datelor, directivele EQU și INCLUDE, macrouri
9. Instructiuni ale limbajului de asamblare: instructiuni de transfer, conversii, operatii aritmetice cu semn si fara semn, operatii de deplasare si rotire de biti, operatii logice pe biti
10. Instructiuni de salt conditionat si neconditionat, instructiuni de ciclare, instructiuni pe siruri. Conceptul de depășire și modul în care arhitectura 80x86 se comportă
11. Implementarea apelului de subprograme si programare multimodul: convenții de apel: CDECL și STDCALL, cod de apel, cod de intrare, cod de iesire, directivele de import-export EXTRN și GLOBAL
12. Legarea de module NASM cu module scrise în limbaje de nivel înalt (studiu de caz – programarea C). Exemple și discuții pentru apeluri recursive
13. Biblioteci statice și dinamice din Windows: LIB vs. DLL. Formatele de fișier obiect de ieșire NASM și biblioteca suport. Biblioteci de sistem Win32: exemple de gestionare a fișierelor, de Expunerea, conversația, dezbateră, problematizarea, descoperirea gestionare a proceselor, de gestionare a memoriei. Implementare biblioteci utilizator.
14. Adresare reală vs. Adresare protejată. Interacțiunea dintre programele utilizatorului și nucleul OS. Conceptul de memorie virtuală. Prezentare generală a procesului de segmentare și paginare. Coduri de configurare și de protecție: de la modul real la modul protejat, tranziție pe 32 de biți.

Bibliografie:

1. Al. Vancea, F. Boian, D. Bufnea, A. Andreica, A. Darabant, A. Navroschi – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2014.
 2. Al. Vancea, F. Boian, D. Bufnea, A. Gog, A. Darabant, A. Sabau – Arhitectura calculatoarelor. Limbajul de asamblare 80x86., Editura Risoprint, Cluj-Napoca, 2005.
 3. A. Gog, A. Sabau, D. Bufnea, A. Sterca, A. Darabant, Al. Vancea – Programarea în limbaj de asamblare 80x86. Exemple si aplicatii., Editura Risoprint, Cluj-Napoca, 2005.
 4. Randal Hyde – The Art of Assembly Programming, No Starch Press, 2003.
- <http://homepage.mac.com/randyhyde/webster.cs.ucr.edu/www>.

artofasm.com/DOS/index.html)

5. Boian F.M. Vancea A. Arhitectura calculatoarelor, suport de curs. Facultatea de Matematica si Informatica, Centrul de Formare Continua si Invatamânt la Distanta,. Ed. Centrului de Formare Continua si Invatamânt la Distanta, Cluj, 2002
6. Irvine, K.R., 2015. Assembly language for x86 processors.
7. Kusswurm, D., 2014. Modern X86 Assembly Language Programming. Springer.
8. Carter, P.A., 2004. PC Assembly Language. Github: <http://pacman128.github.io/static/pcasm-book.pdf>
9. Cavanagh, J., 2013. X86 Assembly Language and C Fundamentals. CRC Press.
10. Guide, P., 2011. Intel® 64 and ia-32 architectures software developer's manual. Volume 3B: System programming Guide, Part, 2, p.11. (<http://www.facweb.iitkgp.ac.in/~goutam/compiler/readingMaterial/intelXeon/253665.pdf>)
11. BitDefender internal documentations – materiale postate pe pagina cursului

D. Limbaje formale si tehnici de compilare (lb. română)

1. Compilator. Introducere.
2. Analiza lexicala. Limbaje. Introducere
3. Gramatica. Clasificarea Chomsky. Automate finite.
4. Limbaje regulate. Generatoare de analizoare lexicale
5. Proprietati de inchidere ale limbajelor regulate.
6. Gramatici independente de context
7. Generatoare de analizoare sintactice. Automate Push Down
8. Gramatici de atribute
9. Analiza sintactica
10. Cod intermediar. Modele de calcul

Bibliografie:

1. A.V. AHO, D.J. ULLMAN - Principles of computer design, Addison-Wesley, 1978.
2. A.V. AHO, D.J. ULLMAN - The theory of parsing, translation and compiling, Prentice-Hall, Engl. Cliffs., N.J., 1972, 1973.
3. D. GRIES - Compiler construction for digital computers,, John Wiley, New York, 1971.
4. MOTOGNA, S. – Metode de proiectare a compilatoarelor, Ed. Albastra, 2006
5. SIPSER, M., Introduction to the theory of computation, PWS Pulb. Co., 1997.
6. L.D. SERBANATI - Limbaje de programare si compilatoare, Ed. Academiei RSR, 1987.
7. <http://www.cs.ubbcluj.ro/~dana/2018-2019/LFTC/ResurseCurs/>

E. Limbaje formale si tehnici de compilare (lb. engleză)

1. Compilator. Introducere
2. Analiza lexicala. Limbaje. Introducere
3. Gramatica. Clasificarea Chomsky. Automate finite
4. Limbaje regulate. Generatoare de analizoare lexicale
5. Proprietati de inchidere ale limbajelor regulate.
6. Gramatici independente de context

	7. Generatoare de analizoare sintactice. Automate Push Down 8. Gramatici de atribute 9. Analiza sintactica 10. Cod intermediar . Modele de calcul 11. Aspecte recapitulative teoretice si practice. Aplicativitatea mecanismelor formale studiate in proiectarea compilatoarelor. Bibliografie 1. A.V. AHO, D.J. ULLMAN - Principles of computer design, Addison-Wesley, 1978. 2. A.V. AHO, D.J. ULLMAN - The theory of parsing, translation and compiling, Prentice-Hall, Engl. Cliffs., N.J., 1972, 1973. 3. D. GRIES - Compiler construction for digital computers,, John Wiley, New York, 1971. 4. MOTOGNA, S. – Metode de proiectare a compilatoarelor, Ed. Albastra, 2006 5. SIPSER, M., Introduction to the theory of computation, PWS Pulb. Co., 1997 6. CSÖRNYEI ZOLTÁN, Bevezetés a fordítóprogramok elméletébe, I, II., ELTE, Budapest, 1996 7. L.D. SERBANATI - Limbaje de programare si compilatoare, Ed. Academiei RSR, 1987. 8. CSÖRNYEI ZOLTÁN, Fordítási algoritmusok, Erdélyi Tankönyvtanács, Kolozsvár, 2000. 9. DEMETROVICS JÁNOS-DENEV, J.-PAVLOV, R., A számítástudomány matematikai alapjai, Nemzeti Tankönyvkiadó, Budapest, 1999 10. GRUNE, DICK - BAL, H. - JACOBS, C. - LANGENDOEN, K.: Modern Compiler Design, John Wiley, 2000
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

