



Universitatea din Craiova
FACULTATEA DE AUTOMATICĂ, CALCULATOARE ȘI
ELECTRONICĂ
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA
INFORMAȚIEI

Bvd. Decebal 107
CRAIOVA, ROMANIA

Tel. 40 - (0)251 - 438198
Fax 40 - (0)251 - 438198

Tematica de concurs

pentru ocuparea postului de asistent, pe perioadă determinată, poz. 51, din Statul de funcții al Departamentului de Calculatoare și Tehnologia Informației, anul universitar 2022-2023

A. Tematica pentru proba scrisă și proba orală

Proiectarea algoritmilor

1. Introducere în analiza și proiectarea algoritmilor
2. Metoda „divide-et-impera”
3. Corectitudinea și testarea algoritmilor
4. Algoritmi de sortare. Complexitatea sortării
5. Tipuri abstracte de date. Liste
6. Stive și cozi. Alocarea dinamică a memoriei
7. Tabele de dispersie
8. Grafuri și arbori. Reprezentarea arborilor
9. Reprezentarea și parcurgerea grafurilor
10. Metoda programării dinamice
11. Explorare exhaustivă și „backtracking”
12. Metoda „greedy”
13. Introducere în complexitatea calculului

Programare orientată pe obiecte

1. Paradigme de programare. Avantajele programării orientate pe obiecte
2. Definirea și utilizarea claselor. Funcții constructor și funcția destructor
3. Compunerea obiectelor
4. Ierarhii de clase. Moștenire
5. Clase imbricate. Funcții și clase prietene
6. Supraîncărcarea operatorilor
7. Polimorfism și funcții virtuale
8. Clase și funcții parametrizate. Mecanismul template
9. Ierarhii de clase pentru tratarea excepțiilor
10. Ierarhii de clase pentru operațiile de intrare/ieșire
11. Elemente de programare generică
12. Șabloane de proiectare în C++

Informatica aplicată II

1. Introducere în limbajul Java și mediul de programare integrată Eclipse
2. Fundamente ale proiectării orientate pe obiect
3. Concepte generale ale proiectării orientate pe obiecte
4. Interfețe, legare dinamică și polimorfism
5. Șabloane de proiectare orientată pe obiect. Șabloane creaționale

6. Șabloane de proiectare orientată pe obiect. Șabloane structurale
7. Șabloane de proiectare orientată pe obiect. Șabloane comportamentale
8. Principii de proiectare orientată pe obiect (SOLID)
9. Principii de proiectare orientată pe obiect (GRASP)

B. Tematica lucrărilor de laborator

Proiectarea algoritmilor

1. Experimente cu algoritmi elementari
2. Analiza experimentală a algoritmilor
3. Experimente cu algoritmi „divide et impera”
4. Experimente cu algoritmi de sortare
5. Tipuri abstracte de date. Implementarea listelor
6. Implementarea stivelor și cozilor
7. Implementarea tabelor de dispersie
8. Implementarea și parcurgerea arborilor.
9. Implementarea și parcurgerea grafurilor
10. Experimente cu algoritmi de programare dinamică.
11. Experimente cu algoritmi combinatoriali
12. Experimente cu algoritmi „backtracking”
13. Experimente cu algoritmi „greedy”

Programarea orientată pe obiecte

1. Programare modulară în limbajul C
2. Programare modulară în limbajul C++. Definirea și utilizarea claselor
3. Clase compuse
4. Moștenirea și ierarhii de clase
5. Clase prietene și clase imbricate
6. Supraîncărcarea funcțiilor și a operatorilor
7. Clase abstracte. Funcții virtuale și polimorfism
8. Mecanismul template. Clase și funcții parametrizate
9. Tratarea excepțiilor
10. Ierarhii de clase pentru operații de intrare/ieșire
11. Biblioteca STL. Programare generică
12. Șabloane de proiectare în C++

Informatica aplicată II

1. Introducere în limbajul Java. Instrumente Java
2. Obiecte, clase de obiecte și ierarhii de clase în limbajul Java
3. Abstractizarea, încapsulare, reutilizare și persistența cu limbajul Java
4. Interfețe, legarea dinamică și implementarea polimorfismului în limbajul Java
5. Utilizarea șabloanelor creaționale în limbajul Java
6. Implementări ale șabloanelor creaționale în limbajul Java
7. Utilizarea șabloanelor structurale în limbajul Java
8. Implementări ale șabloanelor structurale în limbajul Java
9. Utilizarea șabloanelor comportamentale în limbajul Java
10. Implementări ale șabloanelor comportamentale în limbajul Java
11. Aplicarea SOLID în limbajul Java
12. Aplicarea SOLID în limbajul Java
13. Aplicarea GRASP în limbajul Java
14. Aplicarea GRASP în limbajul Java

Bibliografie

Proiectarea algoritmilor

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, *Introduction to Algorithms, 3rd Edition*, MIT Press, 2009
2. Steven S. Skiena, *The Algorithm Design Manual (2nd ed.)*, Springer, 2011.
<http://www.algorist.com/>
3. Antti Laaksonen, *Guide to Competitive Programming. Learning and Improving Algorithms Through Contests*, Springer, 2020, <https://link.springer.com/book/10.1007/978-3-030-39357-1>

Programare orientata pe obiecte

1. Marius Brezovan, 2008, Programare orientata pe obiecte in limbajul C++, Editura SITECH, Craiova, ISBN 978-606-530-093-4
2. Bruce Eckel, 2000, Thinking in C++, Prentice Hall
3. Bjarne Stroustrup, 1997, The C++ Programming Language, Addison-Wesley
4. Scott Meyers, 1996, Effective C++, Addison-Wesley
5. Stanley Lippman, Josee Lajoie, 1998, C++ Primer, Addison-Wesley
6. Andrei Alexandrescu, 2002, Programarea modernă în C++, Programare generică si modele de proiectare aplicate, Teora
7. E. Gamma, R. Helm, R. Johnson, J. Vlissides, 2002, Design Patterns. Sabloane de proiectare, Teora

Informatica aplicata II

1. E. Gamma, R. Helm, R. Johnson, J. Vlissides, 2002, Design Patterns. Sabloane de proiectare, Teora
2. Robert, Martin (2002). Agile Software Development: Principles, Patterns, and Practices. Editura Prentice Hall.
3. Schmidt, Douglas C.; Michael Stal; Hans Rohnert; Frank Buschmann (2000). Pattern-Oriented Software Architecture, Volume 2: Patterns for Concurrent and Networked Objects. John Wiley & Sons.
4. Fowler, Martin (2002). Patterns of Enterprise Application Architecture. Addison-Wesley.
5. Hohpe, Gregor; Bobby Woolf (2003). Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley.
6. Freeman, Eric T; Elisabeth Robson; Bert Bates; Kathy Sierra (2004). Head First Design Patterns. O'Reilly Media