**UNIVERSITATEA TEHNICA “GHEORGHE ASACHI” DIN IAŞI**

**Facultatea de Automatica şi Calculatoare**

**Departamentul de Calculatoare**

Concurs pentru ocuparea postului de asistent universitar, poz. 59

Disciplinele postului: Programare orientată pe obiecte,

Programare II,

Sisteme distribuite

**Tematica de concurs**

**pentru ocuparea postului de asistent poz. 59, din Statul de**

**funcții al Departamentului de Calculatoare**

**pe anul universitar 2019-2020**

**Pentru proba scrisă, proba orală şi proba practică**

1. **Programare orientată pe obiecte şi Programare II - pentru proba orală şi proba practică**
2. Introducere C++:
   1. tipuri de data în C++
   2. operaţii intrare ieşire (cin, cout) C++
   3. funcţii membre ale structurilor
   4. operatorul de rezoluţie
3. Tipul referinţă în C++
4. Alocarea şi dealocarea dinamică de memorie (pentru variabile simple şi pentru tablouri)
5. Funcţii cu parametrii impliciţi
6. Supraîncărcarea funcţiilor
7. Constante
   1. Variabile constante
   2. Funcţii constante
   3. Pointeri constanţi
   4. Funcţii membre constante
8. Namespace-uri
9. Principiile de baza ale POO:
   1. Abstractizarea datelor
   2. Incapsularea
   3. Polimorfismul
   4. Moştenirea
10. Clase
    1. specificatorii de acces (teorie, exemple, specificatori impliciţi)
    2. Constructori (constructorul de copiere (apelare implicită), liste de iniţializare, constructori expliciţi, constructor implicit, constructor fără lista de argumente, constructor de iniţializare, constructor de copiere)
    3. Destructor
    4. Pointerul this
11. Supraîncărcarea operatorilor
    1. prin funcții membre
    2. prin funcții globale
    3. exemplu operatorul +
    4. operatorii impliciţi = și &
    5. Operatorul ++ forma post şi prefixată
12. Membri statici
    1. Variabile membre statice
    2. Funcţii membre statice
13. Funcţii inline
14. Compoziţia
15. Agregarea
16. Template-uri
    1. Conversii int-float
    2. Templateuri de funcții
17. Standard C++ library
    1. STD vs STL
    2. STD (Fişiere header STD, Iteratori, <vector>, <string>, std::sort – descriere, exemplu de utilizare)
18. Funcţii prietene
19. Clase prietene
20. Moştenirea
    1. Ierarhii de tipuri
    2. Specificatori de acces
    3. Suprascrierea metodelor din clasa de bază
    4. Ordinea de apelare a constructorilor şi a destructorilor
    5. Constructori apelaţi implicit
    6. Rescrierea functiilor din clasa de bază
    7. Moştenirea multiplă
21. Polimorfism
    1. Pointeri la clasa de bază
    2. Operatorul de referenţiere &
    3. Tip declarat / tip real
    4. Apelarea unei funcţii suprascrise
    5. Funcţii virtuale
       1. Tabela virtuală
    6. Apelare constructori şi destructori in cazul polimorfismului
       1. Constructori virtuali
       2. Destructori virtuali
    7. Funcții virtuale pure
    8. Type casting
22. Design patterns
    1. Factory Method
    2. Adapter
    3. Chain of Responsability
23. Exceptii in C++

**Bibliografie:**

1. Bjarne Stroustrup - The C++ Programming Language, 4nd edition, 2013
2. Bjarne Stroustrup - Programming: Principles and Practice Using C++ (2nd Edition), 2014
3. **Sisteme distribuite - pentru proba scrisă şi proba orală**

1. Sisteme distribuite de fişiere (SDF)

* 1. Exemple de SDF şi analiza lor
  2. Atomicitate şi ordinea evenimentelor
  3. Consistenţa şi tratarea la nivel hardware şi software
  4. Tehnici şi algoritmi pentru replicarea fişierelor
  5. Sistemul NFS, SNF, Decorum, AFS

2. Echilibrarea încărcării

a. Problematica

b. Echilibrarea statică a încărcării

c. Metode de planificare dinamică

d. Echilibrarea prin predicţia încărcărilor

3. Memoria virtuală distribuită DSM

a. Memoria distribuită simulată

b. Memorie comuna distribuită realizată software

c. Modele de consistenţă

4. Cluster

a. Justificare apariţiei conceptului de cluster şi clasificări

b. Arhitectura unui cluster

c. Tehnici şi suportul pentru disponibilitate

d. Exemple de Clustere

5. GRID

a. Calcul de mare incărcare

b. Descrierea Arhitecturii Grid

c. Clasificări ale Tehnlologiilor Grid

d. Problema programării şi tehnologii utilizate

6. SOA

a. Fundamentele SOA

b. Implementări SOA

c. Tipuri de Nori

7. Peer to Peer P2P

a. Autoorganizarea distribuită

b. Sisteme P2P nestructurate şi structurate, exemple

c. Proiectarea unei aplicaţii P2P

d. Problema clasificării documentelor

8. Sisteme distribuite tolerante la erori

a. Servere, servicii şi relatia lor de dependenţă

b. Abordari hardware

c. Sincronizarea membrilor unui grup tolerant la erori

d. Puncte de salvare a execuţiei şi refacerea

9. Calcul distribuit mobil

a. Arhitecturi specifice

b. Retele sensor

c. Modele de mobilitate şi protocoale de routare

10. Inteligenţa Artificiala Distribuită – DAI

a. Agenţi mobili

b. Limbaje specific pentru comunicare

c. Limbaje pentru conţinut

**Bibliografie:**

1. Muhl, G., Fiege L, Pietzuch P., „Distributed event based systems”, Springer-Verlag Berlin Heidelberg, Germania, 2006
2. Păstravanu O. “Sisteme cu evenimente discrete – Tehnici calitative bazate pe formalismul reţelelor Petri”, Ed. Matrix Rom, Bucureţti, 1997.
3. Pfister G., “In search of clusters”, Prentice Hall, 1995.
4. Raynal, M. , “Distributed Algorithms and Protocols”, John Wiley &Sons Ltd, 1988.
5. Raynal, M., Helary,J.M., “Syncronisation and Control of Distributed Systems and Programs ”, John Wiley &Sons Ltd, 1988.
6. Tanembaumm A. S., "Distributed Systems", Prentice Hall 1993.
7. Mihai Horia Zaharia,  Cătălin Horghidan, Claudiu Mihăilă, Sisteme Distribuite: indrumar de laborator, Politehnium, 2007, ISBN 978-973-621-175-1, 225 format academic
8. Angelo R. Bobak, Distributed and Multi-Database Systems, Boston, US, Artech House, 1995
9. Dmitry Korzun, Andrei Gurtov, Structured Peer-to-Peer Systems - Fundamentals of Hierarchical Organization, Routing, Scaling, and Security, Springer Heidelberg, 2013
10. Wolfgang Effelsberg, Ralf Steinmetz,Thorsten Strufe, Benchmarking Peer-to-Peer Systems Understanding Quality of Service in Large-Scale Distributed Systems, Springer Heidelberg, 2013
11. A.D. Kshemkalyani, M. Singhal, [Distributed Computing: Principles, Algorithms, and Systems,](http://www.cambridge.org/us/catalogue/catalogue.asp?isbn=9780521189842) ISBN: Cambridge University Press, March 2011
12. Nakov S, Dimitrov D, Kirilov R, et al. Fundamentals of computer programming with C#. Sofia: Faber Publishing 2013.
13. Stephens R. Beginning Software Engineering. Indianapolis: John Wiley & Sons, Inc. 2015
14. Millett S. Professional ASP.NET Design Patterns. Indianapollis: Wilwy Publishing Inc. 2010.
15. Coulouris G, Dollimore J, Kindberg T. Distributed Systems: Concepts and Design 5th Edition. New York, PA: Pearson 2011.
16. Freeman A. Expert ASP.NET Web API2 for MVC Developpers.New York, PA: Apress Media LLC 2016.
17. Schumacher M, Fernandez-Buglioni E, Hybertson D, et al, Security Patterns - Integrating Security and Systems Engineering. Chichester: John Wiley & Sons Ltd. 2006:59-83.
18. Dorrans B. Beginning ASP.NET Security. Chichester: John Wiley & Sons Ltd. 2010:255-399.
19. Bellifemine, F., Caire, G., & Greenwood, D. (2007). Developing Multi-Agent Systems with JADE. Chippenham, Wiltshire: John Wiley & Sons Inc.
20. Weiss, G. (2013). Multiagent Systems, (2th ed.). Cambridge: MIT Press.
21. Barreiros, M., & Lundqvist, P. (2011). QOS-Enabled networks. West Sussex: John Wiley & Sons, Ltd.
22. Betz, C. T. (2007). Architecture and Patterns for IT Service Management, Resource Planning, and Governance: Making Shoes for the Cobbler's Children: Making Shoes for the Cobbler's Children. San Francisco: Morgan Kaufmann Publishers Inc.

Decan, Director de departament,

Prof. Vasile-Ion Manta Prof. Petru Caşcaval