



Academia de Studii Economice
Departamentul de Informatică și Cibernetică Economică

Calea Dorobanți, 15-17, Sector 1, București, 010552 (camera 2314)

Tel.: +40 21 319 19 00, ext. 319, 336, Fax: +40 21 311 20 66

www.dice.ase.ro

Contest Topics for Associate Professor
Position 62, year 2023-2024, semester 1

Topics: Software Quality and Testing; Parallel Processing; Object-Oriented Programming; Source Code Programming Security

Software Quality and Testing

1. Principles for writing source code - Clean Code (SOLID, DRY, KISS, YAGNI, WET, naming conventions, defining and implementing functions)
2. Design Patterns: Singleton, Simple Factory, Factory Method, Abstract Factory, Builder, Adapter, Decorator, Facade, Flyweight, Chain of Responsibility, Command, Observer, State, Strategy, Memento
3. Management of source code versions - Git, SVN
4. Unit Testing Concepts
5. Using the JUnit framework (versions 3, 4 and 5)
6. Software quality concepts - metrics, indicators, tools, automatic testing (testing web applications with Selenium platform)

Bibliography

1. I. Ivan, C. Boja, S. Capisizu, M. Popa, *Managementul calității aplicațiilor informatice*, ASE, 2006, România
2. I. Sommerville, *Software Engineering. 9th ed.*, Addison-Wesley, 2011, Statele Unite ale Americii
3. S. Pressman, *Software Engineering: A Practitioner's Approach. 7th ed.*, McGraw-Hill, 2009, Statele Unite ale Americii
4. P. Pocatilu, *Costurile testării software*, ASE, 2004, România
5. S. McConnell, *Code complete, 2nd ed.*, Microsoft Press, 2004, Statele Unite ale Americii
6. Scott Chacon, Ben Straub, *Pro Git, 2nd edition*, Apress, 2014, <http://git-scm.com/book/en/v2>, Statele Unite ale Americii
7. Robert C. Martin, *Clean Code, A Handbook of Agile Software Craftsmanship*, Prentice Hall, 2009, Statele Unite ale Americii
8. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995, https://books.google.ro/books?id=6oHuKQe3TjQC&printsec=frontcover&source=gbs_ge_summary_r&hl=ro#v=onepage&q&f=false, Statele Unite ale Americii



Academia de Studii Economice
Departamentul de Informatică și Cibernetică Economică

Calea Dorobanți, 15-17, Sector 1, București, 010552 (camera 2314)

Tel.: +40 21 319 19 00, ext. 319, 336, Fax: +40 21 311 20 66

www.dice.ase.ro

9. Andy Hunt, Dave Thomas, *Pragmatic Unit Testing in Java with JUnit*, The Pragmatic Programmers, 2004, Statele Unite ale Americii
10. Mădălina Zurini, Alin Zamfiroiu, *Calitate și testare software. Studii de caz*, ASE, București, 2017, România

Parallel Processing

1. Concepts specific to Parallel Processing. Comparison with sequential, asynchronous and concurrent programming.
2. Advantages and disadvantages of parallel processing. Amdahl's law.
3. Parallel processing on multiple threads in Java.
4. Threads in Java. Runnable vs Thread vs Callable
5. Thread orchestration in Java using semaphores and barriers.
6. Problems of parallel processing: race condition, deadlock, livelock, cache coherence, cache false sharing. Examples in Java
7. Atomic, volatile and synchronized in Java.
8. The advantages and disadvantages of threads execution orchestrating using synchronized context.
9. Java collections in the context of multi-threaded parallel processing
10. Thread management in Java through ExecutorService
11. Java Parallel Streams
12. Specific algorithms for parallel processing. Example for sorting and searching substrings.

Bibliography

1. Oracle, JDK 11 Documentation, <https://docs.oracle.com/en/java/javase/11/>
2. Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, Doug Lea, *Java Concurrency in Practice*, Addison-Wesley Professional; 1st edition (May 9, 2006), ISBN, 978-0321349606
3. Jeff Friesen, *Concurrency in Java*, Apress; 1st ed. edition (November 23, 2015), ISBN 978-1484216996
4. Javier Fernández González, *Mastering Concurrency Programming with Java 9*, Packt Publishing; 2nd Revised edition (July 17, 2017), ISBN 978-1785887949
5. David E. Culler, Anoop Gupta, Jaswinder Pal Singh, *Parallel Computer Architecture: A Hardware/Software Approach*, Morgan Kaufmann Publishers, 1997, Statele Unite ale Americii



Academia de Studii Economice
Departamentul de Informatică și Cibernetică Economică

Calea Dorobanți, 15-17, Sector 1, București, 010552 (camera 2314)

Tel.: +40 21 319 19 00, ext. 319, 336, Fax: +40 21 311 20 66

www.dice.ase.ro

6. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, *Introduction to Parallel Computing, Second Edition*, Addison Wesley, 2003, <https://www-users.cs.umn.edu/~karypis/parbook/>, Statele Unite ale Americii
7. Blaise Barney, *Introduction to Parallel Computing*, 2018, https://computing.llnl.gov/tutorials/parallel_comp/, Statele Unite ale Americii
8. Gheorghe Dodescu, Bogdan Oancea, Mădălina Răceanu, *Procesare paralelă*, Ed. Economica, 2002, România
9. Cosmina Ivan, *Tehnologii și Aplicații în Calculul Paralel și distribuit*, UTPRESS, 2019, <https://biblioteca.utcluj.ro/files/carti-online-cu-coperta/390-5.pdf>, România

Object-Oriented Programming

1. Procedural programming elements: functions, transfer parameters, pointers to data and functions, memory classes.
2. Object Oriented concepts: class, object, constructor, destructor, access methods, this pointer. Declaration and implementation of methods in a class and outside it.
3. Copy constructor, operator = overload, objects in HEAP and namespaces.
4. Conversions between different types of objects (cast operator, operator = and copy constructor), vector objects, *const* modifier, static members (static), constant objects, constant pointers to objects and pointers to constant objects.
5. Try-catch mechanism in C ++.
6. Operator overloading.
7. Inheritance. Polymorphism.
8. Virtual functions, overriding functions, multiple inheritance.
9. RTTI type mechanisms, multiple inheritance and dynamic cast.
10. Template functions and classes. Methods of encapsulation template functions and classes in static libraries / dynamic.
11. Input/Output operations using streams (console and files). The concepts of serialization / deserialization of objects.
12. Standard Template Library (STL), containers, iterators and algorithms. Classes for string, map, list, vector, etc.

Bibliography

1. Ion Smeureanu, Marian Dârdală, *Programarea orientată obiect în limbajul C++*, Editura CISON, 2002, România;
2. Ion Smeureanu, Marian Dârdală, *Programarea în limbajul C/C++*, Editura CISON, 2001, România
3. Bjarne Stroustrup, *The C++ Programming Language, 3rd Edition*, Addison-Wesley, <http://www.research.att.com/~bs/3rd.html>, Statele Unite ale Americii
4. C++ Standard, ISO, <https://isocpp.org/std/the-standard>, Statele Unite ale Americii



Academia de Studii Economice
Departamentul de Informatică și Cibernetică Economică

Calea Dorobanți, 15-17, Sector 1, București, 010552 (camera 2314)

Tel.: +40 21 319 19 00, ext. 319, 336, Fax: +40 21 311 20 66

www.dice.ase.ro

Source Code Programming Security

1. Low Level Security;
2. String Vulnerabilities;
3. Race Conditions;
4. Reverse Engineering Vulnerabilities;
5. OOP Related Vulnerabilities;
6. Web Related Vulnerabilities;
7. Virtual Machine-Based Languages Vulnerabilities.

Bibliography

1. David LeBlanc, Michael Howard, *Writing Secure Code*, Pearson Education, 2002
2. John Viega, Gary R, McGraw, *Building Secure Software: How to Avoid Security Problems the Right Way*, Pearson Education, 2001
3. University of Maryland, *Software Security Course*,
<https://www.coursera.org/learn/software-security>
4. University of Virginia, *Defense Against the Dark Arts*,
<http://www.cs.virginia.edu/~ww6r/CS4630/syllabus.html>