



```
DFS(node)
{
    stack ← node
    visited[node] = true
    while stack not empty
        v ← stack
        print v
        for each child c of v
            if not visited[c]
                stack ← c
                visited[c] = true
}
```



Data Structures and Algorithms

Course Program, Evaluation, Exams, Resources

```
DFS(node)
{
    for each child c of node
        DFS(c);
    print the current node;
}
```

Data structures and algorithms

Telerik Software Academy

<http://academy.telerik.com>

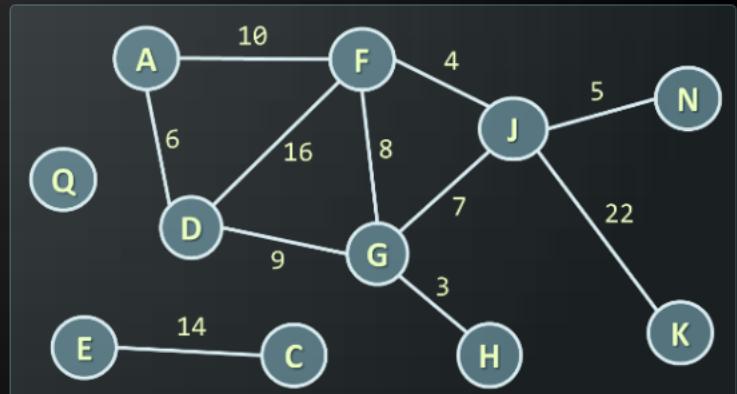
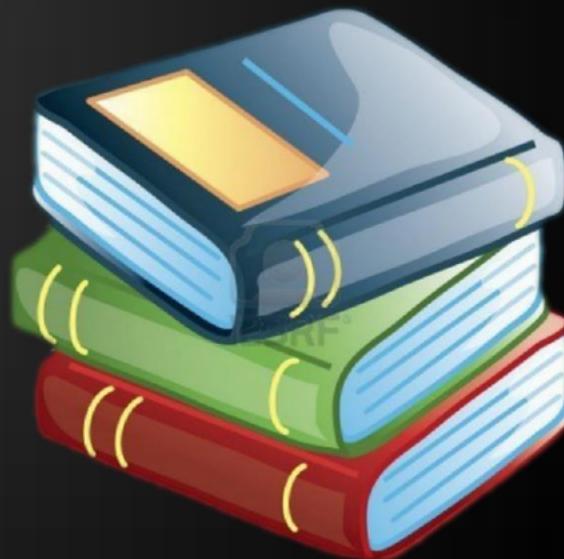


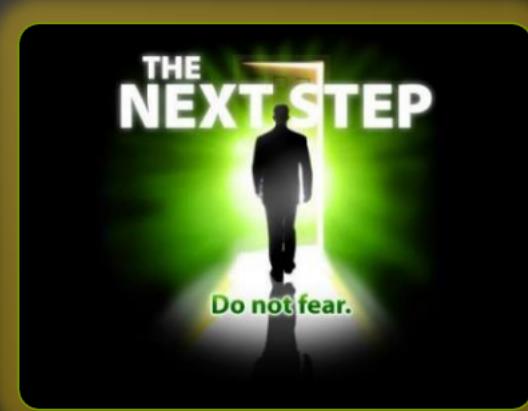
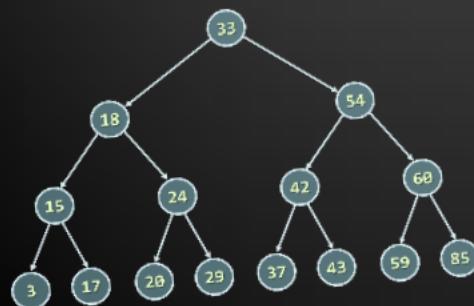
Table of Contents

- ◆ What's Coming Next in the Academy?
- ◆ The Data Structures and Algorithms Program
- ◆ The Trainers Team
- ◆ Exams and Evaluation
 - ◆ The Practical Exam
- ◆ Recommended Books
- ◆ Resources for the Course



Data Structures & Algorithms

The Next Module in the Software Academy (The Programming Track)



```
DFS(node)
{
    for each child c of node
        DFS(c);
    print the current node;
}
```

What's Coming Next?

- ◆ Data Structures and Algorithms (DS&A)
 - ◆ Continuation of High-Quality Code course
 - ◆ Data structures (lists, trees, hash-tables, graphs), complexity, algorithms (recursion, combinatorics, dynamic programming, graphs)
- ◆ Lectures 3 times a week
- ◆ The course exam?
 - ◆ Practical exam (@ BGCoder)



What's Coming Next?

2013

Oct - Nov

Nov - Jan

Jan - Mar

Mar

Mar - Apr

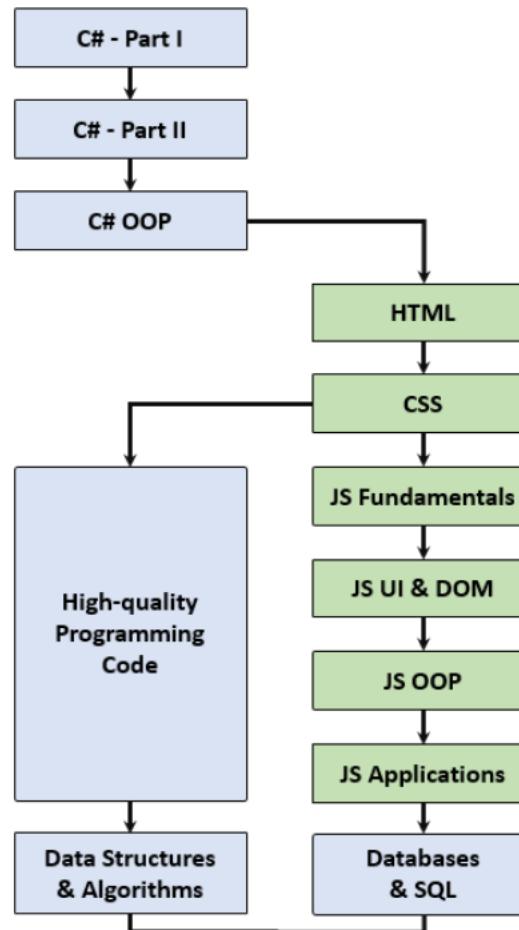
Apr - May

May - Jun

Jun - Jul

Jul

Aug



2014

Sept - Nov

Mobile Development

Web Development

QA Academy

Mobile

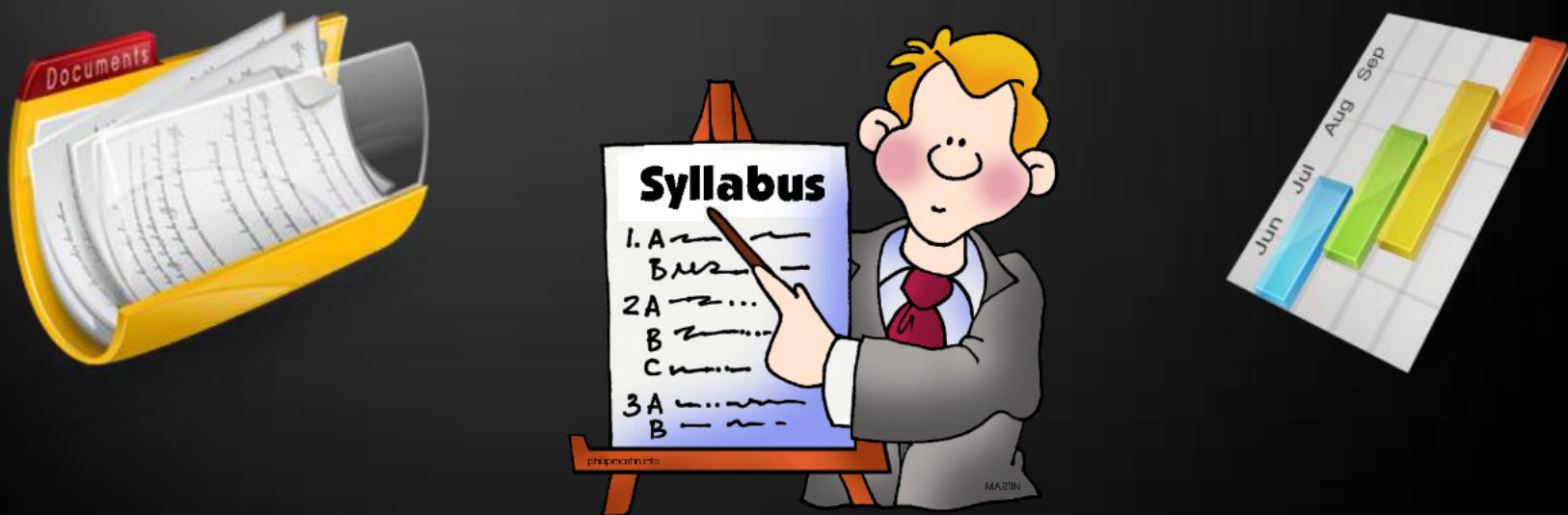
Starts	Exam at	Course
9-Sep	22-Sep	Web Services & cloud
23-Sep	30-Sep	Hydrib Apps
1-Oct	14-Oct	Android
15-Oct	28-Oct	iOS
29-Oct	11-Nov	Windows Universal

Web

Starts	Exam at	Course
9-Sep	22-Sep	Web Services & cloud
23-Sep	29-Sep	SPA Apps
30-Sep	13-Oct	End-to-end apps
14-Oct	27-Oct	ASP.NET Web Forms
28-Oct	10-Nov	ASP.NET MVC

Data Structures and Algorithms Course Program

What Will We Cover in the Course?



The Course Program

1. **Course Overview.**
Overview of Data Structures, ADT and Algorithms.
Algorithm Complexity.
2. **Linear Data Structures: List, Queue, Stack**
Trees, Tree-Like Structures, Balanced Search Trees.
Tree Traversals: BFS and DFS
3. **Dictionaries, Hash Tables and Sets**
4. **Advanced Data Structures. Wintellect Power**
Collections. Bag, Multi-Dictionary, Priority Queue.
Data Structure Efficiency.
5. **Sorting and Searching Algorithms**

The Course Program (2)

6. Recursion
7. Combinatorial Algorithms. Generating Variations, Permutations, Combinations
8. Dynamic Programming. Divide-and-Conquer. Classical Dynamic Programming Problems
9. Graphs, Representation and Basic Graph Algorithms (Shortest Paths, Minimal Spanning Tree). Other Algorithms (Greedy, Geometry, Randomized)
10. Problem Solving Methodology.
11. Exam Preparation and Exam



The Trainers Team



◆ Nikolay Kostov

- ◆ Team Lead, Senior Developer and Trainer @ Telerik Corp.
- ◆ Student at Sofia University
 - ◆ Computer Science
- ◆ IT and Informatics competitions contestant
- ◆ Graduate from the second season of Telerik Software Academy
- ◆ Email: [nikolay.kostov \[at\] telerik.com](mailto:nikolay.kostov[at]telerik.com)
- ◆ Blog: <http://nikolay.it>



◆ Doncho Minkov

- ◆ Senior Technical Trainer
@ Telerik Software Academy
- ◆ Student in Sofia University
 - ◆ Software Engineering
- ◆ Contestant in the Informatics competitions
- ◆ Graduate from the first season of
Telerik Software Academy
- ◆ Email: **doncho.minkov [at] telerik.com**
- ◆ Blog: **<http://minkov.it>**



◆ Ivaylo Kenov

- ◆ Technical Trainer @ Telerik Software Academy
- ◆ Graduate from the fourth season of Telerik Software Academy
- ◆ Mathematical competitions contestant
- ◆ E-mail: ivaylo.kenov [at] telerik.com
- ◆ Champion in OOP and DSA
- ◆ <http://ivaylo.bgcoder.com>



Trainers Team (4)

◆ Evlogi Hristov

- Technical Trainer
@ Telerik Software Academy
- Graduate from the fourth season
of Telerik Software Academy
- E-mail: evlogi.hristov [at] telerik.com



Evaluation

Thank God There Are Bonuses!



Data Structures & Algorithms – Evaluation

◆ Evaluation components

- ◆ Practical exam – 75%
- ◆ Homework – 10%
 - ◆ Homework evaluation – 5%
- ◆ Attendance in class – 10%



◆ Bonuses

- ◆ Forums activity – bonus up to 5%



Pass / Excellence / Fail Criteria

- ◆ Criteria for pass / pass with excellence / fail

- ◆ Pass with excellence

- ◆ Very high exam results

- or

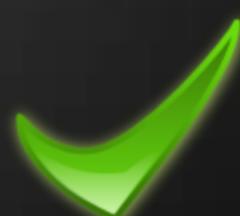
- ◆ High total results

- ◆ Pass

- ◆ Average total results

- ◆ Fail

- ◆ Low total results or low exam results



Homework Peer Reviews

- ◆ Everyone will get feedback for their homework
- ◆ Everyone will give feedback for few random homework submissions
 - Students submit homework anonymously
 - Please exclude your name from the submissions!
 - For each homework submitted
 - Students evaluate 3 random homeworks
 - From the same topic, after the deadline
 - Give written feedback, at least 200 characters
 - Low-quality feedback → report for punishment

- ◆ 5 practical problems for 8 hours
- ◆ Real-time feedback @ BGCoder



Screenshot of the BGCoder.com website showing practice results for "Telerik Academy Exam 1 @ 6 Dec 2011 Morning".

The page displays a table of user scores across five problems:

Nr	User	Answer	Total points	Problem 1 - Ship Damage	Problem 2 - Tribonacci	Problem 3 - Fir Tree	Problem 4 - We All Love Bits!	Problem 5 - Pillars
1	Aleksander Kolev	500	100	100	100	100	100	100
2	Lachezar Tsonev	500	100	100	100	100	100	100
3	Pavel Asenov	490	100	100	100	100	100	90
4	Deyan Bolkdev	470	70	100	100	100	100	100
5	Venelin Banov	460	80	100	100	100	100	80
6	Fatme Havaluova	390	0	100	100	100	100	90
7	Blagovesta Kostova	350	50	100	100	100	100	0
8	Nikolay Demirev	330	10	90	100	70	60	
9	Andrey Simeonov	300	0	100	100	0	100	
10	Boris Gutsev	300	100	100	100	0	0	
11	Krasimir Gitsov	300	100	100	100	0	0	
12	Krasin Georgiev	300	0	100	0	100	100	
13	Nikolay Avramov	300	100	100	100	0	0	
14	Stefan Chonov	290	20	90	100	0	80	
15	Eyup Yusein	220	100	90	0	0	30	
16	Milen Palavrov	220	20	100	100	0	0	
17	Vera Stojnova	210	20	90	100	0	0	



Use the Same Email!

- ◆ Please use the same email address in:
 - <http://telerikacademy.com>
 - <http://bgcoder.com>
 - <http://forums.academy.telerik.com>
- ◆ Otherwise your score could be incorrectly calculated!





Recommended Books

The C# Textbook



The official textbook for the course

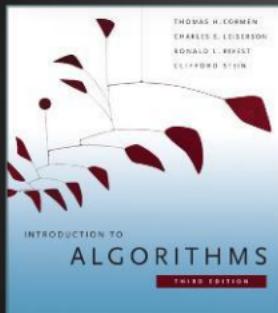
- “Introduction to Programming with C#”, Nakov S. and his team, 2010
- Freely downloadable from:
www.introprogramming.info

- The C# programming tracks follows the book
- Data Structures → chapters 16-19
- Algorithms → chapters 23-26 (partially)

Recommended Books



Telerik Algo Academy
algoacademy.telerik.com



Introduction to Algorithms
Cormen, Leiserson, Rivest, a
ISBN 9780262033848, goo.gl/cf3b5



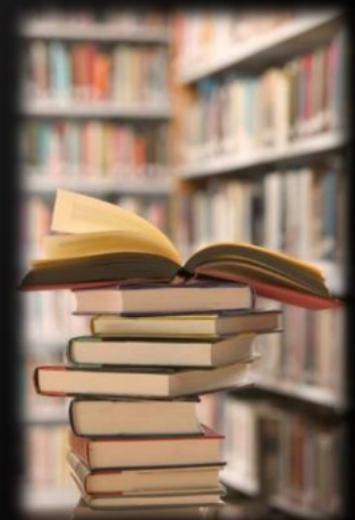
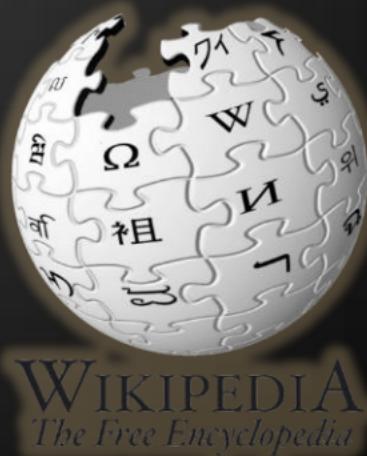
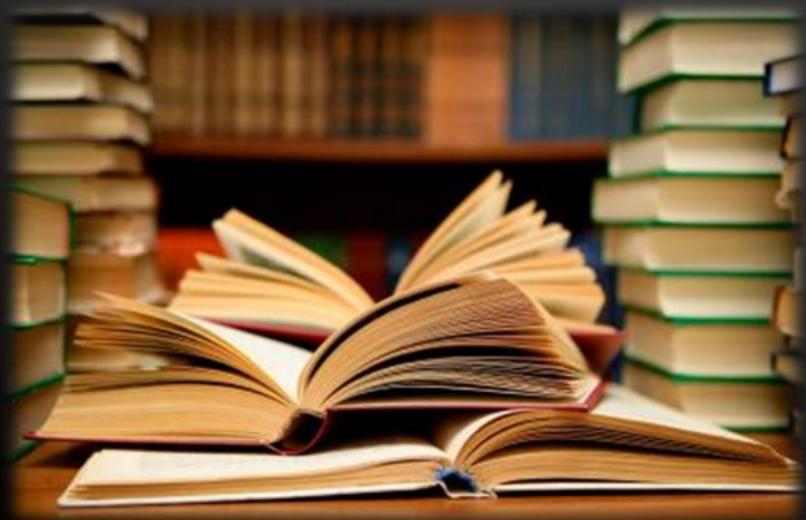
Programming = ++Algorithms;
Preslav Nakov, Panayot Dobrikov,
ISBN 954-8905-06-X
www.programirane.org

[Animated structures and algorithms](#)



Resources

What We Need in Addition to this Course Content?



Course Web Site & Forums

- ◆ Register for the "Telerik Academy Forums":

forums.academy.telerik.com/csharp-programming/data-structures-algorithms

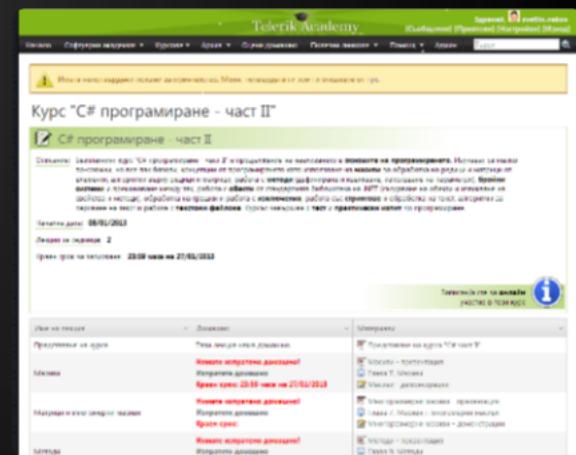
- ◆ Discuss the course exercises with your colleagues
 - ◆ Find solutions for the exercises
 - ◆ Share source code / discuss ideas
- ◆ The DS&A official web site:



<http://academy.telerik.com/student-courses/programming/data-structures-algorithms/>

Telerik Integrated Learning System (TILS)

- ◆ The Telerik Integrated Learning System (TILS)
 - ◆ www.telerikacademy.com
 - ◆ Important resource for all students
 - ◆ Homework submissions
 - ◆ Homework peer reviews
 - ◆ Presence cards with barcode
 - ◆ Reports about your results

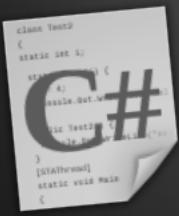
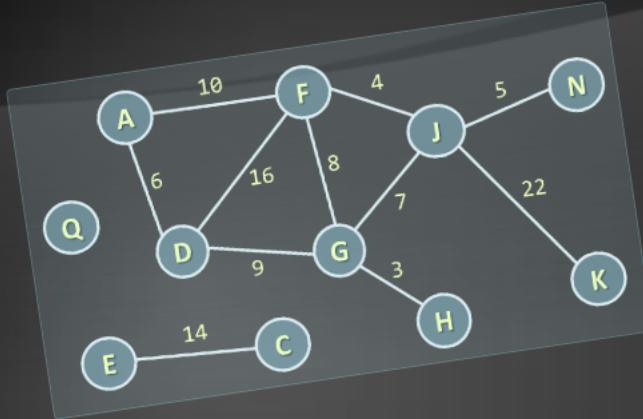


telerikacademy.com/Courses/Courses/Details/186

Required Software

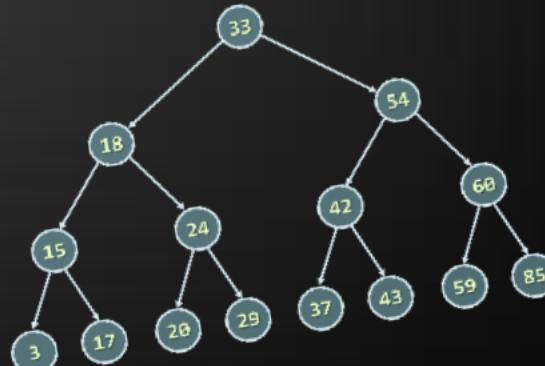
- ◆ Software needed for this course:
 - ◆ Microsoft Windows (XP / Win7 / Win8)
 - ◆ Microsoft Visual Studio 2013 or Visual Studio Express 2013 (free version of VS 2013)
 - ◆ .NET Framework 4.5 (included in Visual Studio)
 - ◆ Visual Studio 2012 is also OK





Questions?

```
DFS(node)
{
    for each child c of node
        DFS(c);
    print the current node;
}
```



```
DFS(node)
{
    stack ← node
    visited[node] = true
    while stack not empty
        v ← stack
        print v
        for each child c of v
            if not visited[c]
                stack ← c
                visited[c] = true
}
```

Free Trainings @ Telerik Academy

- ◆ C# Programming @ Telerik Academy

- ◆ csharpfundamentals.telerik.com



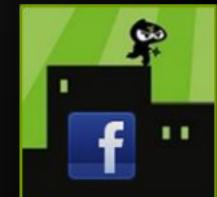
- ◆ Telerik Software Academy

- ◆ academy.telerik.com

Telerik Academy

- ◆ Telerik Academy @ Facebook

- ◆ facebook.com/TelerikAcademy



- ◆ Telerik Software Academy Forums

- ◆ forums.academy.telerik.com

