Welcome

Dr. Haitham A. El-Ghareeb

Information Systems Department
Faculty of Computers and Information Sciences
Mansoura University

helghareeb@gmail.com

September 16, 2012















Olympic Flame during the Opening Ceremony of the **2004** Summer Olympics, held in Athens.















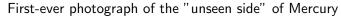


































http://www.helghareeb.net/publications/journals

















































@helghareeb
https://twitter.com/helghareeb













https://plus.google.com/u/0/110986963706567841321

















www.zazzle.com/Artfulinages

OFFICE HOURS

Open most days about 9 o'clock occasionally as early as 8, but some days as late as 10 or 11.

We close about 5:30 or 6:00 occasionally about 4 or 5, but sometimes as late as 11 or 12.

Somedays or afternoons, we aren't here at all, and lately I've been here just about all the time, except when I'm someplace else!











Why Data Structures?





Why Data Structures?

• Design of Effective and Efficient Data Structures and Algorithms





Why Data Structures?

- Design of Effective and Efficient Data Structures and Algorithms
- Advanced Data Structures and Algorithm Design Paradigms



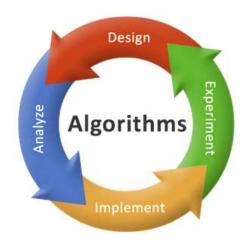


Why Data Structures?

- Design of Effective and Efficient Data Structures and Algorithms
- Advanced Data Structures and Algorithm Design Paradigms
- Algorithm Analysis Techniques Intractability











Course gives the student greater experience in:





Course gives the student greater experience in:

• Program design in a language-independent setting





Course gives the student greater experience in:

- Program design in a language-independent setting
- Discusses pragmatic and mathematical aspects of program efficiency





Course gives the student greater experience in:

- Program design in a language-independent setting
- Discusses pragmatic and mathematical aspects of program efficiency
- Algorithm Design Examination of data structures, abstract data, and algorithm design intended to develop knowledge and programming skills byond an introductory level











Topics in Detail:





• Algorithm and Data Structure Design





- Algorithm and Data Structure Design
- Abstract Data Types





- Algorithm and Data Structure Design
- Abstract Data Types
- Queues





- Algorithm and Data Structure Design
- Abstract Data Types
- Queues
- Stacks





- Algorithm and Data Structure Design
- Abstract Data Types
- Queues
- Stacks
- Lists





- Algorithm and Data Structure Design
- Abstract Data Types
- Queues
- Stacks
- Lists
- Trees and Graphs





- Algorithm and Data Structure Design
- Abstract Data Types
- Queues
- Stacks
- Lists
- Trees and Graphs
- Sorting and Searching





- Algorithm and Data Structure Design
- Abstract Data Types
- Queues
- Stacks
- Lists
- Trees and Graphs
- Sorting and Searching
- Motivation and Introduction of Analysis for Algorithm Complexity





Dr. Haitham A. El-Ghareeb







100 Marks - Grouped as follows:





100 Marks - Grouped as follows:

• Mid-Term Exam: 10 Marks





100 Marks - Grouped as follows:

Mid-Term Exam: 10 Marks

• Labs: 10 Marks





100 Marks - Grouped as follows:

Mid-Term Exam: 10 Marks

Labs: 10 Marks

Attendance: 5 Marks





100 Marks - Grouped as follows:

Mid-Term Exam: 10 Marks

Labs: 10 Marks

• Attendance: 5 Marks

Report: 5 Marks





100 Marks - Grouped as follows:

Mid-Term Exam: 10 Marks

Labs: 10 Marks

• Attendance: 5 Marks

• Report: 5 Marks

• Oral: 10 Marks





100 Marks - Grouped as follows:

Mid-Term Exam: 10 Marks

Labs: 10 Marks

Attendance: 5 Marks

• Report: 5 Marks

Oral: 10 Marks

Project (Bonus): 10 Marks





100 Marks - Grouped as follows:

Mid-Term Exam: 10 Marks

Labs: 10 Marks

Attendance: 5 Marks

• Report: 5 Marks

Oral: 10 Marks

• Project (Bonus): 10 Marks

• Final Exam: 60 Marks





100 Marks - Grouped as follows:

• Mid-Term Exam: 10 Marks

Labs: 10 Marks

Attendance: 5 Marks

Report: 5 Marks

Oral: 10 Marks

Project (Bonus): 10 Marks

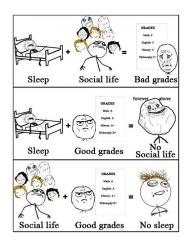
• Final Exam: 60 Marks

Total: 100 Marks





Dr. Haitham A. El-Ghareeb











Write a Report on "Data Structures and Algorithms" Covering:

• Why we need Data Structures





- Why we need Data Structures
- Different Data Structures, Why we need them





- Why we need Data Structures
- Different Data Structures, Why we need them
- \bullet Advantages of "Python / C# / Java" as a Programming Language





- Why we need Data Structures
- Different Data Structures, Why we need them
- ullet Advantages of "Python / C# / Java" as a Programming Language
- How we can use "Python" "C#" "Java" in implementing "Data Structures and Algorithms"





- Why we need Data Structures
- Different Data Structures, Why we need them
- \bullet Advantages of "Python / C# / Java" as a Programming Language
- How we can use "Python" "C#" "Java" in implementing "Data Structures and Algorithms"
- Limits of Computer Resources





- Why we need Data Structures
- Different Data Structures, Why we need them
- \bullet Advantages of "Python / C# / Java" as a Programming Language
- How we can use "Python" "C#" "Java" in implementing "Data Structures and Algorithms"
- Limits of Computer Resources
- Why Computer Algorithms are Important





- Why we need Data Structures
- Different Data Structures, Why we need them
- \bullet Advantages of "Python / C# / Java" as a Programming Language
- How we can use "Python" "C#" "Java" in implementing "Data Structures and Algorithms"
- Limits of Computer Resources
- Why Computer Algorithms are Important
- Different Programming problems that can be solved by Algorithms





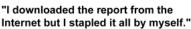
- Why we need Data Structures
- Different Data Structures, Why we need them
- ullet Advantages of "Python / C# / Java" as a Programming Language
- How we can use "Python" "C#" "Java" in implementing "Data Structures and Algorithms"
- Limits of Computer Resources
- Why Computer Algorithms are Important
- Different Programming problems that can be solved by Algorithms
- \bullet Examples of "Python / C# / Java based Solutions" for different "Computer Algorithms" challenges





Dr. Haitham A. El-Ghareeb









Dr. Haitham A. El-Ghareeb







Project

Bonus

Working Project must be delivered at the end of the subject. Project entitles how we used Data Structures and Algorithms to solve one or more of the challenging problems "8 Queen Problem, Sorting, etc."









How the Project Leader understood it



How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed



What Programming Language Shall I Learn?

common times, Livin St., Sar Spr., Colors State, Proceedings of the Section of the Colors of the Col PLUS TOPROS CAN THEN, MACROS COMMAN INSTRUMENTAL PROPERTY STATE OF THE PROPERTY STATES OF T States one of the Spines Of Spines to the Spines of Spines of Spines on Spines of Spin One Patrick on the Patrick of the Anglish State of Second NAME OF THE PARTY Special Color of the Color of t Programming Language of the pr Stanger State Column State Sta ther moved, all the Stability of the PRODUCE SEASON SELECTION PRODUCES PROTOCOLS Tomas (Alleren and Annual Person Person and Annual Person Service Was a secretary and David Stone Spreaduring Laurent and the Proposite State toronic tree has been stated with the state of the state NAME AND ADDRESS OF OWNER, WHEN PARTY AND PROPERTY PARTY AND PERSONS ASSESSED. BY STATE WAS DEED TO SEE THE PARTY AND PARTY AN their SOL TOM from Parent State Spirit State Season State TEX (10 seasons provided up all addressed from it programmy between the formers (but former). Marriary House Provinces And Statement Collections Deposited Street, Street, Street Wolfstern may that there is a more hope to write. STOPRA Program (Constitute DATA Speciments, MEATEN, MATERIAL MC 11425-Dis 1990 C. projekter som Starter School, J.





I don't like Programming!

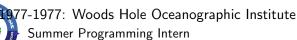
- Network Engineer
- Web Designer 'Front End Engineer'
- Graphic Designer '2D, 3D'
- System Administrator
- Software Testing
- Quality Assurance Team
- etc...





Peter Norvig

- 1996-1998: Junglee Corp.
 - Chief Scientist
- 1994-1996: Harlequin, Inc.
 - Chief Designer
- 1991-1994: Sun Microsystems Labs
 - Senior Scientist
- 1986-1991: University of California, Berkeley
 - Research Faculty Member
- 1985-1986: University of Southern California
 - Assistant Professor
- 1978-1980: Higher Order Software, Inc.
 - Member of Technical Staff





Teach Yourself Programming in Ten Years

Peter Norvig

- 2001-now: Google
 - Director of Research (2006-now);
 - formerly Director of Search Quality (2002-2006)
 - ▶ and Machine Learning (2001)
- 1998-2001: NASA Ames Research Center
 - Division Chief, Computational Sciences









• Get interested in programming, and do some because it is fun.





- Get interested in programming, and do some because it is fun.
- Program. The best kind of learning is learning by doing.





- Get interested in programming, and do some because it is fun.
- Program. The best kind of learning is learning by doing.
- Talk with other programmers; read other programs.





- Get interested in programming, and do some because it is fun.
- Program. The best kind of learning is learning by doing.
- Talk with other programmers; read other programs.
- Work on projects with other programmers. Be the best programmer on some projects; be the worst on some others.





- Get interested in programming, and do some because it is fun.
- Program. The best kind of learning is learning by doing.
- Talk with other programmers; read other programs.
- Work on projects with other programmers. Be the best programmer on some projects; be the worst on some others.
- Learn at least a half dozen programming languages.





- Get interested in programming, and do some because it is fun.
- Program. The best kind of learning is learning by doing.
- Talk with other programmers; read other programs.
- Work on projects with other programmers. Be the best programmer on some projects; be the worst on some others.
- Learn at least a half dozen programming languages.
 - ▶ Include one language that supports class abstractions (like Java or C++),





- Get interested in programming, and do some because it is fun.
- Program. The best kind of learning is learning by doing.
- Talk with other programmers; read other programs.
- Work on projects with other programmers. Be the best programmer on some projects; be the worst on some others.
- Learn at least a half dozen programming languages.
 - Include one language that supports class abstractions (like Java or C++),
 - one that supports functional abstraction (like Lisp or ML),





- Get interested in programming, and do some because it is fun.
- Program. The best kind of learning is learning by doing.
- Talk with other programmers; read other programs.
- Work on projects with other programmers. Be the best programmer on some projects; be the worst on some others.
- Learn at least a half dozen programming languages.
 - Include one language that supports class abstractions (like Java or C++),
 - one that supports functional abstraction (like Lisp or ML),
 - one that supports syntactic abstraction (like Lisp),





- Get interested in programming, and do some because it is fun.
- Program. The best kind of learning is learning by doing.
- Talk with other programmers; read other programs.
- Work on projects with other programmers. Be the best programmer on some projects; be the worst on some others.
- Learn at least a half dozen programming languages.
 - ▶ Include one language that supports class abstractions (like Java or C++),
 - one that supports functional abstraction (like Lisp or ML),
 - one that supports syntactic abstraction (like Lisp),
 - one that supports declarative specifications (like Prolog or C++ templates),



- Get interested in programming, and do some because it is fun.
- Program. The best kind of learning is learning by doing.
- Talk with other programmers; read other programs.
- Work on projects with other programmers. Be the best programmer on some projects; be the worst on some others.
- Learn at least a half dozen programming languages.
 - ▶ Include one language that supports class abstractions (like Java or C++),
 - one that supports functional abstraction (like Lisp or ML),
 - one that supports syntactic abstraction (like Lisp),
 - one that supports declarative specifications (like Prolog or C++ templates),
 - one that supports coroutines (like Icon or Scheme),



- Get interested in programming, and do some because it is fun.
- Program. The best kind of learning is learning by doing.
- Talk with other programmers; read other programs.
- Work on projects with other programmers. Be the best programmer on some projects; be the worst on some others.
- Learn at least a half dozen programming languages.
 - Include one language that supports class abstractions (like Java or C++),
 - one that supports functional abstraction (like Lisp or ML),
 - one that supports syntactic abstraction (like Lisp),
 - one that supports declarative specifications (like Prolog or C++ templates),
- The second secon

one that supports coroutines (like Icon or Scheme), and one that supports parallelism (like Sisal).





• Remember that there is a "computer" in "computer science".





- Remember that there is a "computer" in "computer science".
 - ▶ Know how long it takes your computer to execute an instruction,





- Remember that there is a "computer" in "computer science".
 - Know how long it takes your computer to execute an instruction,
 - fetch a word from memory (with and without a cache miss),





- Remember that there is a "computer" in "computer science".
 - Know how long it takes your computer to execute an instruction,
 - fetch a word from memory (with and without a cache miss),
 - ▶ read consecutive words from disk,





- Remember that there is a "computer" in "computer science".
 - Know how long it takes your computer to execute an instruction,
 - fetch a word from memory (with and without a cache miss),
 - read consecutive words from disk,
 - and seek to a new location on disk.





- Remember that there is a "computer" in "computer science".
 - ▶ Know how long it takes your computer to execute an instruction,
 - fetch a word from memory (with and without a cache miss),
 - read consecutive words from disk,
 - and seek to a new location on disk.
- Get involved in a language standardization effort. It could be the ANSI C++ committee, or it could be deciding if your local coding style will have 2 or 4 space indentation levels.





Course Resources

- Course Page: http://courses.helghareeb.net/2012-2013/dsa
 - ▶ You will find: Updated Slides, Important Announcements, etc.
- Course G+ Page:
 - https://plus.google.com/u/1/b/108828963463022379521/108828963463022379521/posts
 - ▶ You can post questions, start discussions, share material, etc.





Text Book

- Lecture Notes will be available soon inchallah
- Related Links and Further Readings are posted on Web site
- Fresh items are posted on G+





Teaching Assistants

- Osama Abu Al-Nasr
- Mohamed Al-Hossini
- Eslam Reda
- Shahenda Al-Kholi
- Mohamed Abd-Raboh





Labs

- Visual Studio .Net
- .Net Framework
- C#
- Iron Python





Finally







Finally





