EL2310 – Scientific Programming

Lecture 17: Conclusion

Ramviyas Parasuraman (ramviyas@kth.se)
Yasemin Bekiroglu (yaseminb@kth.se)

Royal Institute of Technology – KTH

Overview

Lecture 17: Conclusions

- Reminders
- File I/O in C++
- Conclusion of the Lectures

The help sessions

- Scheduled: C++ help session: Fri 24.10.2015, 15:00-17:00, Room ”22:an”, Teknikringen 14
- Change? Doodle poll: http://doodle.com/poll/ze6yeazdictg3aht

File Input and Output

- Use `fstream` library
- use `oftream` for output and `ifstream` for input to or from files.

EL2310

- Covered basics of programming,
- Started with MATLAB, continued with C and finished with C++.

MATLAB - What you should have learned:

- Be comfortable working with MATLAB
- Writing scripts and functions using basic elements of programming (loops, branching, ...)
- Taking advantage of in-built functions (load data, plot data), especially the visualization capabilities
- Translating a mathematical problem into a MATLAB code.

- Understand a MATLAB code by seeing it.
- Know when (and how) to use MATLAB in another course.

MATLAB - What you should have learned:

- Be comfortable working with MATLAB
- Writing scripts and functions using basic elements of programming (loops, branching, ...)
- Taking advantage of in-built functions (load data, plot data), especially the visualization capabilities
- Translating a mathematical problem into a MATLAB code.
- Understand a MATLAB code by seeing it.
- Know when (and how) to use MATLAB in another course.

MATLAB - What you should have learned:

- Be comfortable working with MATLAB
- Writing scripts and functions using basic elements of programming (loops, branching, ...)
- Taking advantage of in-built functions (load data, plot data), especially the visualization capabilities
- Translating a mathematical problem into a MATLAB code.

- Understand a MATLAB code by seeing it.
- Know when (and how) to use MATLAB in another course.

MATLAB - What you should have learned:

- Be comfortable working with MATLAB
- Writing scripts and functions using basic elements of programming (loops, branching, ...)
- Taking advantage of in-built functions (load data, plot data), especially the visualization capabilities
- Translating a mathematical problem into a MATLAB code.

- Understand a MATLAB code by seeing it.
- Know when (and how) to use MATLAB in another course.

C - What you should have learned:

- Working with C: how to write, compile, link, execute.
- Declaring and initializing variables, basic data types, pointers, memory allocation...
- Writing basic programs (loops, branching, ...)
- Using standard libraries (e.g. for printing data)
- Understand others C code.
- Know when (and how) to use C in another course.

C - What you should have learned:

- Working with C: how to write, compile, link, execute.
- Declaring and initializing variables, basic data types, pointers, memory allocation...
- Writing basic programs (loops, branching, ...)
- Using standard libraries (e.g. for printing data)
- Understand others C code.
- Know when (and how) to use C in another course.

C - What you should have learned:

- Working with C: how to write, compile, link, execute.
- Declaring and initializing variables, basic data types, pointers, memory allocation...
- Writing basic programs (loops, branching, ...)
- Using standard libraries (e.g. for printing data)
- Understand others C code.
- Know when (and how) to use C in another course.

C - What you should have learned:

- Working with C: how to write, compile, link, execute.
- Declaring and initializing variables, basic data types, pointers, memory allocation...
- Writing basic programs (loops, branching, ...)
- Using standard libraries (e.g. for printing data)
- Understand others C code.
- Know when (and how) to use C in another course.

C++ - What you should have learned:

- What of C you can use in C++ and what C++ has to offer more (or in a different way) ...
- Object Oriented Programming Paradigm: Classes, Polymorphism, Inheritance, Overloading,...
- Declaring classes and creating objects, accessing members, ...
- Understanding of 'conceptual programming', i.e. hiding of functions, declaring of static, const, virtual ...
- Understand others C++ code on seeing it.
- Know when (and how) to use C++ in another lecture.

C++ - What you should have learned:

- What of C you can use in C++ and what C++ has to offer more (or in a different way) ...
- Object Oriented Programming Paradigm: Classes, Polymorphism, Inheritance, Overloading,...
- Declaring classes and creating objects, accessing members, ...
- Understanding of 'conceptual programming', i.e. hiding of functions, declaring of static, const, virtual ...
- Understand others C++ code on seeing it.
- Know when (and how) to use C++ in another lecture.

C++ - What you should have learned:

- What of C you can use in C++ and what C++ has to offer more (or in a different way) ...
- Object Oriented Programming Paradigm: Classes, Polymorphism, Inheritance, Overloading,...
- Declaring classes and creating objects, accessing members, ...
- Understanding of 'conceptual programming', i.e. hiding of functions, declaring of static, const, virtual ...
- Understand others C++ code on seeing it.
- Know when (and how) to use C++ in another lecture.

C++ - What you should have learned:

- What of C you can use in C++ and what C++ has to offer more (or in a different way) ...
- Object Oriented Programming Paradigm: Classes, Polymorphism, Inheritance, Overloading,...
- Declaring classes and creating objects, accessing members, ...
- Understanding of 'conceptual programming', i.e. hiding of functions, declaring of static, const, virtual ...
- Understand others C++ code on seeing it.
- Know when (and how) to use C++ in another lecture.

C++ - What you should have learned:

- What of C you can use in C++ and what C++ has to offer more (or in a different way) ...
- Object Oriented Programming Paradigm: Classes, Polymorphism, Inheritance, Overloading,...
- Declaring classes and creating objects, accessing members, ...
- Understanding of 'conceptual programming', i.e. hiding of functions, declaring of static, const, virtual ...
- Understand others C++ code on seeing it.
- Know when (and how) to use C++ in another lecture.

In general:

- Understanding of basic concepts in programming.
- Be skilled enough using MATLAB, so it does not pose a problem in other courses.
- Solve problems and implement algorithms in C and C++.
- Be able to read and understand existing code written in C or C++.
- Know the importance of writing code which others can understand, change, correct and build upon.

In general:

- Understanding of basic concepts in programming.
- Be skilled enough using MATLAB, so it does not pose a problem in other courses.
- Solve problems and implement algorithms in C and C++.
- Be able to read and understand existing code written in C or C++.
- Know the importance of writing code which others can understand, change, correct and build upon.

In general:

- Understanding of basic concepts in programming.
- Be skilled enough using MATLAB, so it does not pose a problem in other courses.
- Solve problems and implement algorithms in C and C++.
- Be able to read and understand existing code written in C or C++.
- Know the importance of writing code which others can understand, change, correct and build upon.

In general:

- Understanding of basic concepts in programming.
- Be skilled enough using MATLAB, so it does not pose a problem in other courses.
- Solve problems and implement algorithms in C and C++.
- Be able to read and understand existing code written in C or C++.
- Know the importance of writing code which others can understand, change, correct and build upon.

In general:

- Understanding of basic concepts in programming.
- Be skilled enough using MATLAB, so it does not pose a problem in other courses.
- Solve problems and implement algorithms in C and C++.
- Be able to read and understand existing code written in C or C++.
- Know the importance of writing code which others can understand, change, correct and build upon.

Summary

We have learned tools to program but we have not done much Computer Science yet

- Algorithms: *Sorting*, *Mapping*, ...
- Data structures: *Trees*, *Graphs*, ...
- Complexity
- Discrete Math
- ...

How to continue?

- The aim of this course was to get you started
- Hundreds of References and Books - to learn more and have a quick lookup for more specific things you need.
- Some more concentrated programming courses at KTH:
  - **DD2387** Programsystemkonstruktion med C++ 6,0 hp
  - **DD2456** Avancerade objektorienterade system 7,5 hp
- Experience - your own project.

Still to do:

- Our Evaluation
  - You should pass all the projects (Matlab, C, C++)
  - The course is only pass or fail
  - Will be available through BILDA after the C++ project
- Your Evaluation
  - For collecting feedback and opinions about the course.

Still to do:

- Our Evaluation
  - You should pass all the projects (Matlab, C, C++)
  - The course is only pass or fail
  - Will be available through BILDA after the C++ project
- Your Evaluation
  - For collecting feedback and opinions about the course.

Getting involved

- CVAP (Computer Vision and Active Perception) Lab `http://www.nada.kth.se/cvap/` does research in,
  - Computer Vision
  - Robotics and Autonomous Systems
  - **Machine Learning** and **AI**
- If you are interested,
  - Research interaction
  - **2D5348** Individual course in Computer Science
  - Msc. Thesis work
    `https://www.kth.se/en/csc/2.3721/cas/opening`

Getting involved - CVAP

Like to test-drive a robot? - Volunteers needed!

- Volunteer for a user study experiment regarding Human-Robot interface?

Like to test-drive a robot? - Volunteers needed!

- What? Drive the Scooby robot on a maze for a few minutes and answer some questions
- How long? 20 - 30 minutes (max!)
- Where? CVAP (Teknikringen 14, 7th floor)
- When? From week 44-46 (Oct 25 - Nov 14)
- Will I get rewarded? Maybe!
- Other benefits: Get to see cool robots!
- Will I get the robot afterward? Nope.