ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

Εισαγωγή στον Προγραμματισμό
Introduction to Programming

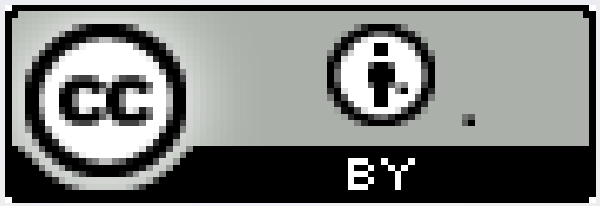
Διάλεξη 1: Εισαγωγή στον Προγραμματισμό και τα Προγράμματα

Γ. Παπαγιαννάκης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται στην άδεια χρήσης **Creative Commons** και ειδικότερα

Αναφορά Δημιουργού 3.0 - Μη εισαγόμενο Ελλάδα (Attribution 3.0– Unported GR)



- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Κρήτης**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

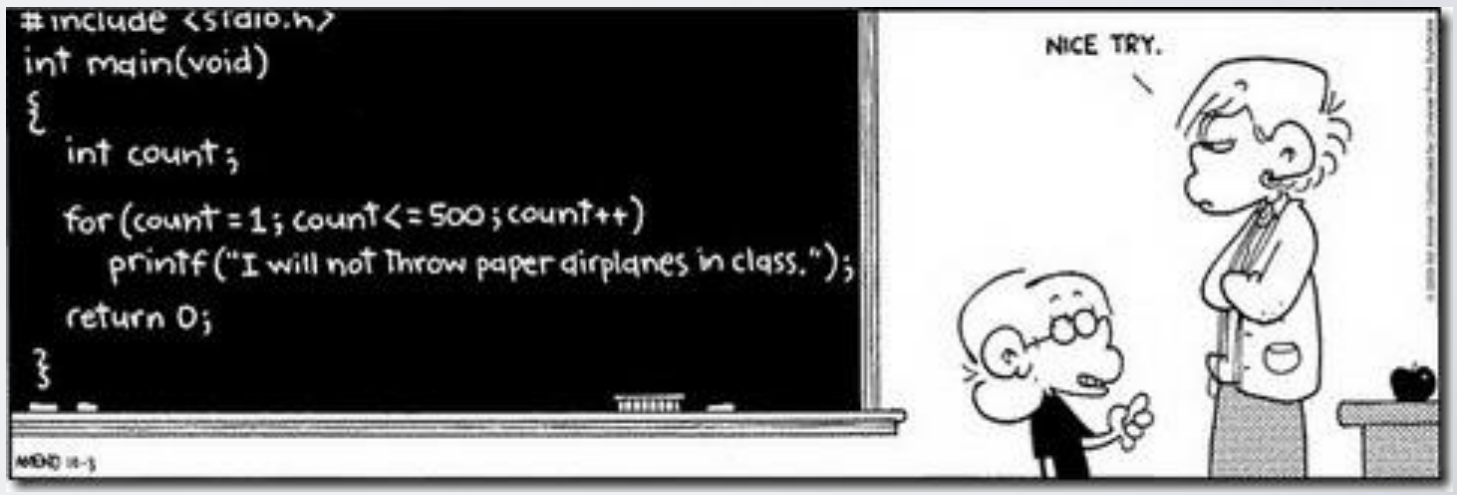
HY-150 Προγραμματισμός
CS-150 Programming

Lecture 1:
Introduction to Programming & Programs

G. Papagiannakis

Overview

- What is this course about?
- What should you already know?
- What will you be expected to do?
- What will you know when you finish
- What is programming
- Applications of programming
- A simple program



Who am I?

- http://www.csd.uoc.gr/~papagian
- Assistant Professor of Computer Graphics, University of Crete
- Research Fellow, Computer Vision and Robotics Laboratory, ICS-FORTH
- Senior researcher and Visiting Lecturer on Computer Graphics (2006-2009), MIRALab, University of Geneva
- PhD in Computer Science, “An illumination registration model for dynamic virtual humans in mixed reality” (2002-2006)
- Research assistant (1999-2006), MIRALab, University of Geneva

Εισαγωγή

- Περιεχόμενο :
  - μέθοδοι προγραμματισμού
  - προγραμματιστικές αρχές
    - δομημένος προγραμματισμός,
    - αφαιρετικότητα, (abstraction)
    - υλοποίηση,
    - έλεγχος, και αποσφαλμάτωση
    - καλές πρακτικές
  - γλώσσα προγραμματισμού: C++

Στόχοι

- Αλγοριθμική σκέψη & Προγραμματισμός
  - Βάση για την Πληροφορική και Υπολογιστικά Μαθηματικά
  - Μέθοδοι επίλυσης προβλημάτων - αλγόριθμοι
  - Κωδικοποίηση αλγορίθμων
  - Διόρθωση/Κατανόηση/Αλλαγή προγράμματος
    - ενημέρωση
    - επαναχρησιμοποίηση

Γλώσσα προγραμματισμού C++

- Μια από τις πιο σημαντικές γλώσσες :
  - Έχει στοιχεία υψηλού & χαμηλού επιπέδου
  - Γενική – πολλαπλές εφαρμογές
  - Αποδοτικά προγράμματα
  - Συμβατότητα / πρότυπο (ISO)

- Σημαντικό βήμα για την εκμάθηση άλλων γλωσσών

Είναι σημαντικός ο προγραμματισμός για επαγγελματική αποκατάσταση;

The Doers - A TEDxAthens Documentary: http://www.youtube.com/watch?v=zxCMXT4RaNo&list=PLBE7D87788B2430F1

Διδασκαλία

- Διαλέξεις
- Φροντιστήρια
- Ασκήσεις
- Multiple-choice quiz
- Τελικό Διαγώνισμα

Πρόγραμμα μαθήματος

- Πρόγραμμα Μαθήματος
  - Τρίτη 5-7 (Αμφ. Α) και Πέμπτη 11-1 (Αμφ. Α)
  - Παρασκευή 5-7 (Αμφ. Α) για φροντιστήριο και αναπληρώσεις

Περιεχόμενο μαθήματος

- Εισαγωγή στον Προγραμματισμό και στη γλώσσα προγραμματισμού C++
- Τύποι Δεδομένων , Τελεστές και Αριθμητικές Εκφράσεις
- Είσοδος & Έξοδος Δεδομένων
- Εντολές επιλογής & Συνθήκες επιλογής
- Εντολές επανάληψης
- Συναρτήσεις, Εμβέλεια Μεταβλητών
- GUI programming, κλάσεις και αντικείμενα
- Πίνακες
- Συναρτήσεις
- Δομές δεδομένων
- Δυναμικές δομές δεδομένων
- Αναδρομή
- Αποσφαλμάτωση προγραμμάτων
- Τεχνικές προγραμματισμού μεγάλων προγραμμάτων, εισαγωγή στον αντικειμενοστρεφή προγραμματισμό

Εμβάθυνση σε αυτό το μάθημα

- Πρακτικότητα
- Πληρότητα
- Εφαρμογή σε πραγματικά προβλήματα
- Επεκτασιμότητα

Εισαγωγή: Εργαλεία

- Σε LINUX/MacOSX
  - g++, μεταφραστής/compiler της C++
  - gdb, debugger της C++ (εύρεση λαθών)
  - Editors : Emacs, vi, vim, gvim, pico,, XCODE για την συγγραφή προγραμμάτων
- Σε Windows
  - Ολοκληρωμένα Περιβάλλοντα προγραμματισμού της C++, όπως ο δωρεάν Microsoft Visual Studio 2010 C++ express: http://www.microsoft.com/visualstudio/en-us/products/2010-editions/visual-cpp-express/
  - cygwin ή MinGW (http://www.mingw.org/), για προγραμματισμό
- Πρόταση:
  - Στήστε ένα βολικό περιβάλλον προγραμματισμού και εξοικειωθείτε μαζί του

Βοήθεια!

- Απορίες στο μάθημα
- Κατά τις ώρες των φροντιστηρίων
- Στις ομάδες συζητήσεων στο web-site του μαθήματος
- Στις ώρες γραφείου

Εισαγωγή: Περί Αντιγραφής

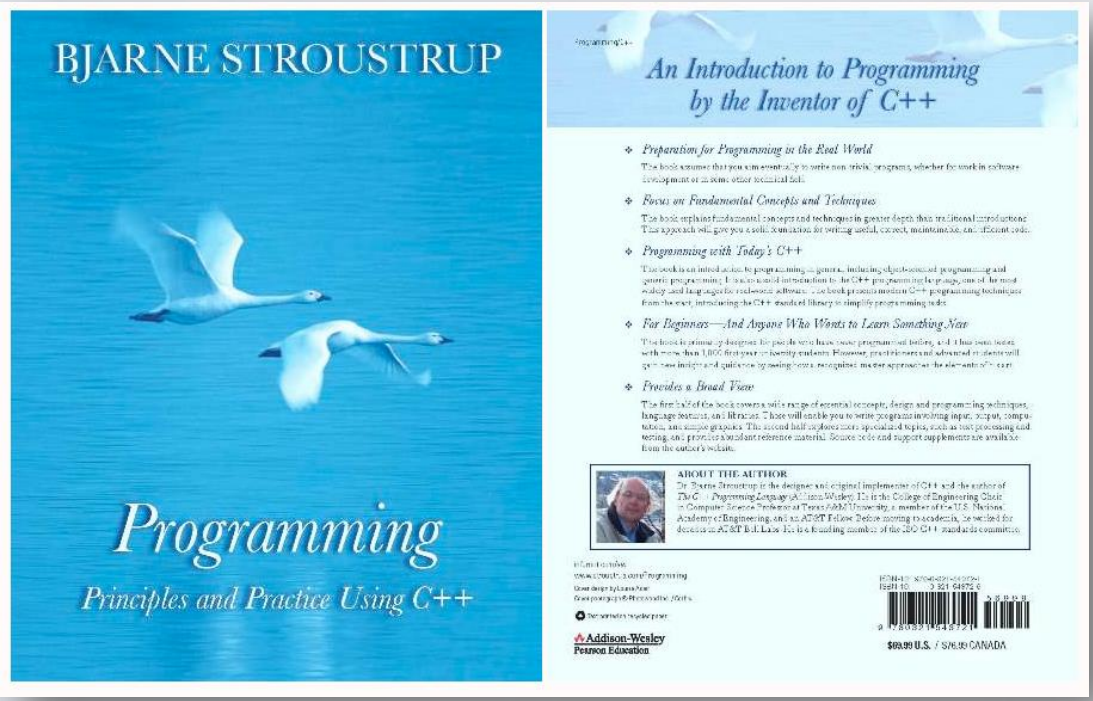
- Τι είναι:
  - Αντιγραφή κώδικα ή μέρους αυτού από άλλη πηγή (συμφοιτητή, διαδίκτυο, κάποιον τρίτο, κτλ)
  - Αντιγραφή της ιδέας ενός αλγορίθμου (χωρίς αναφορά στις πηγές)
  - Αντιγραφή κειμένου ή μέρους αυτού κατά την διάρκεια εξέτασης
  - Αυτόματη ανίχνευση
- Συνέπειες: το λιγότερο μηδενισμός της άσκησης, διαγωνίσματος κτλ. σε όλα τα μέρη που εμπλέκονται στην αντιγραφή
- Όποιος αντιγράφει στον προγραμματισμό θα αντιγράφει συνέχεια: το μάθημα είναι η βάση για τα περισσότερα μαθήματα του τμήματος
- Μην αφήνετε τους άλλους να αντιγράψουν από εσάς. Κλειδώστε τις περιοχές σας:
  - chmod 700 mydir
  - chmod 600 myfile

Προτάσεις

- Ρωτήστε, ενημερωθείτε, διαβάστε, ζητήστε βοήθεια
- Βοήθεια ≠ δεν προσπαθώ
- Εκμεταλλευτείτε τα εργαστήρια-φροντιστήρια (6)
- Ο προγραμματισμός μαθαίνεται μόνο με προγραμματισμό ~ ποδήλατο - κολύμβηση
  - Υπομονή, επιμονή, προσπάθεια
  - Η εξάσκηση (διάβασμα και προγραμματισμός) θα σας κάνει ικανούς για τη λύση ασκήσεων
  - Καλός προγραμματιστής ~ γραμμές κώδικα που έχει γράψει

Course Bibliography

- B. Stroustrup, “Programming: Principles and Practise Using C++,” Addison-Wesley, pp. 1–1268, Jul. 2009.
- B. Stroustrup, Προγραμματισμός με C++, Εκδόσεις Παπασωτηρίου, 2009



Abstract

- Today, we'll outline the aims for this course and present a rough course plan.
- We'll introduce the basic notion of programming and give examples of areas in which software is critical to our civilization.
- Finally, we'll present the simplest possible C++ program and outline how it can be made into running code.

Overview

- Course aims and outline
- Programming
- “Hello, world!”
- Compilation

This is a course

- In Programming
- For beginners
  - who want to become professionals
    - i.e., people who can produce systems that others will use
  - who are assumed to be bright
    - Though not (necessarily) geniuses
  - who are willing to work hard
    - Though do need sleep occasionally, and take a normal course load
- Using the C++ programming language

Not!

- A course in
  - The C++ programming language
- For students
  - who want to become language lawyers
    - We try not to get bogged down in technical obscurities
  - who are assumed to be a bit dim and fairly lazy
    - We try not to spoon feed
- Using
  - Some untested software development methodologies and a lot of unnecessarily long words

The Aims

- Teach/learn
  - Fundamental programming concepts
  - Key useful techniques
  - Basic Standard C++ facilities
- After the course, you'll be able to
  - Write small colloquial C++ programs
  - Read much larger programs
  - Learn the basics of many other languages by yourself
  - Proceed with an "advanced" C++ programming course
- After the course, you will not (yet) be
  - An expert programmer
  - A C++ language expert
  - An expert user of advanced libraries

The Means

- Lectures
  - Attend every one
- Notes
  - Read notes ahead (about one per lecture)
    - Will be posted online
  - Read the notes again after each lecture
    - Will be updated online
  - Feedback is welcome (typos, suggestions, etc.)

Cooperate on Learning

- Except for the work you hand in as individual contributions, we strongly encourage you to collaborate and help each other
- If in doubt if a collaboration is legitimate: ask!
  - Don’t claim to have written code that you copied from others
  - Don’t give anyone else your code (to hand in for a grade)
  - When you rely on the work of others, explicitly list all of your sources – i.e. give credit to those who did the work
- Don’t study alone when you don’t have to
  - Form study groups
  - Do help each other (without plagiarizing)
- Go to your TA's labs
  - Go prepared with questions
  - The only stupid questions are the ones you wanted to ask but didn’t

Why C++ ?

- You can't learn to program without a programming language
- The purpose of a programming language is to allow you to express your ideas in code
- C++ is the language that most directly allows you to express ideas from the largest number of application areas
- C++ is the most widely used language in engineering areas
  - http://www.research.att.com/~bs/applications.html

Why C++ ?

- C++ is precisely and comprehensively defined by an ISO standard
  - And that standard is almost universally accepted
- C++ is available on almost all kinds of computers
- Programming concepts that you learn using C++ can be used fairly directly in other languages
  - Including C, Java, C#, and (less directly) Fortran
  - Objective-C (iOS development) is also very similar to C++

Course outline in Parts

- Part I: The basics
  - Types, variables, strings, console I/O, computations, errors, vectors functions, source files, classes
- Part II: Input and Output
  - File I/O, I/O streams
  - Graphical output
  - Graphical User Interface
- Part III: Data structures and algorithms
  - Free store, pointers, and arrays
  - Lists, maps, sorting and searching, vectors, templates
  - The STL
- Part IV: Broadening the view
  - Software ideals and history
  - Text processing, numerics, embedded systems programming, testing, C, etc.

Rough course outline (Cont.)

- Throughout
  - Program design and development techniques
  - C++ language features
  - Background and related fields, topics, and languages
- Note: Appendices
  - C++ language summary
  - C++ standard library summary
  - Index (extensive)
  - Glossary (short)

Promises

- Detail: We will try to explain every construct used in this course in sufficient detail for real understanding
  - There is no “magic”
- Utility: We will try to explain only useful concepts, constructs, and techniques
  - We will not try to explain every obscure detail
- Completeness: The concepts, constructs, and techniques can be used in combination to construct useful programs
  - There are, of course, many useful concepts, constructs, and techniques beyond what is taught here

More Promises

- Realism: the concepts, constructs, and techniques can be used to build “industrial strength” programs
  - i.e., they have been used to …
- Simplicity: The examples used are among the simplest realistic ones that illustrate the concepts, constructs, and techniques
  - Your exercises and projects will provide more complex examples
- Scalability: The concepts, constructs, and techniques can be used to construct large, reliable, and efficient programs
  - i.e., they have been used to …

Feedback request

- Please post questions and constructive comments to
- the web-site discussion forums (not mailing list)
- Your feedback will be most appreciated
  - On style, contents, detail, examples, clarity, conceptual problems, exercises, missing information, depth, etc.
- Local course support website forums:
  - http://www.csd.uoc.gr/~hy150b (https://elearn.uoc.gr/course/view.php?id=160)
    - Ανακοινώσεις, νέα, ειδήσεις
    - Γενικά (θεωρία, εργαλεία, compiliers, IDE)
    - Άσκηση 1
    - Άσκηση 2
    - Άσκηση 3
    - Άσκηση 4

Why programming?

- Our civilization runs on software
  - Most engineering activities involve software

- Note: most programs do not run on things that look like a PC
  - a screen, a keyboard, a box under the table

Ships

- Design
- Construction
- Management

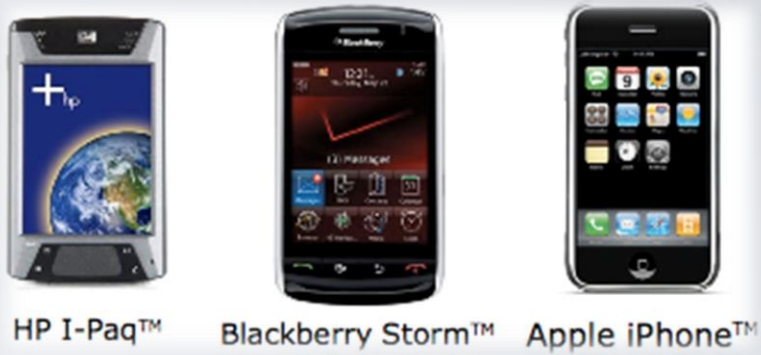
- Monitoring
- Engine
- Hull design
- Pumps

Aircraft

- Communication
- Control
- Display

- Signal processing
- “Gadget” control
- Monitoring

Phones



- Voice quality
- User interfaces
- Billing
- Mobility

- Switching
- Reliability
- Provisioning
- Images

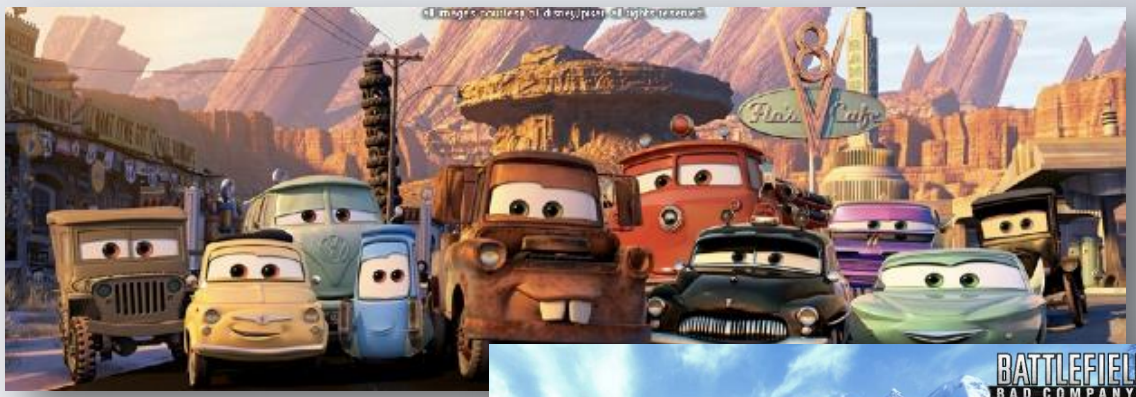
Energy

- Control
- Monitoring
- Analysis
- Design

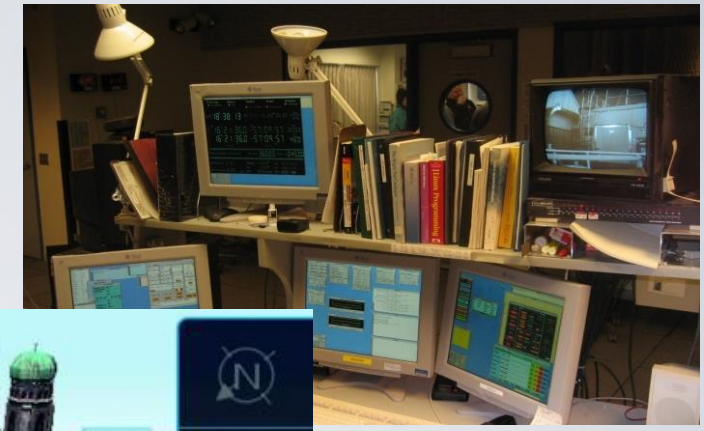
- **Communications**
- **Visualization**
- **Manufacturing**

Computer Graphics

- Movie Industry
- 3D Computer games
- Scientific visualization
- Computer Aided Design (CAD)
- Virtual and Augmented Reality
- Digital photography and video
- Human Computer Interaction
- Art



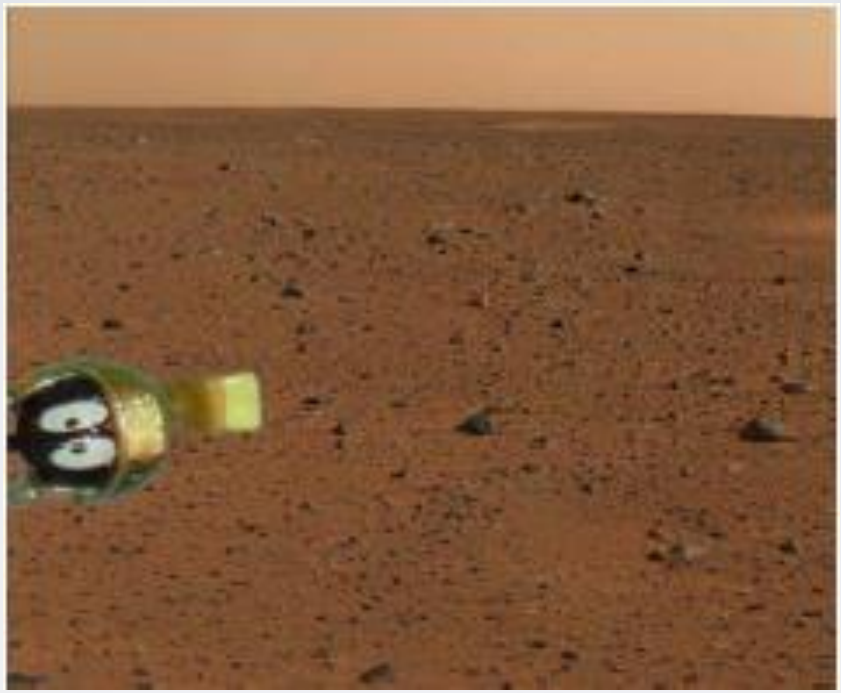
PC/workstation/mobile



- There's a lot more to computing than games, word processing, browsing, and spreadsheets!

Where is C++ Used?

- Just about everywhere



Android, iOS, Mars rovers, animation, graphics, Photoshop, GUI, OS, compilers, slides, chip design, chip manufacturing, semiconductor tools, etc.

See www.research.att/~bs/applications.html

A first program – just the guts…

```
// …

int main()                          // main() is where a C++ program starts
{
        cout << "Hello, world!\n";        // output the 13 characters  Hello, world!
                                         // followed by a new line
        return 0;                        // return a value indicating success
}

// quotes delimit a string literal
// NOTE: “smart” quotes “  ”  will cause compiler problems.
//                  so make sure your quotes are of the style  " "
// \n is a notation for a new line
```

A first program – complete

```
// a first program:

#include "std_lib_facilities.h"      // get the library facilities needed for now

int main()                           // main() is where a C++ program starts
{
        cout << "Hello, world!\n"; // output the 13 characters  Hello, world!
                                                    // followed by a new line
        return 0;                    // return a value indicating success
}

        // note the semicolons; they terminate statements
        // curly brackets { … } group statements into a block
        // main( ) is a function that takes no arguments ( )
        //      and returns an int (integer value) to indicate success or failure
```

A second program

```
// modified for Windows console mode:

#include "std_lib_facilities.h" // get the facilities for this course from the site

int main()                          // main() is where a C++ program starts
{
        cout << "Hello, world\n";        // output the 13 characters  hello, world!
                                         // followed by a new line
        keep_window_open();             // wait for a keystroke
        return 0;                       // return a value indicating success
}

// without keep_window_open() the output window will be closed immediately
// before you have a chance to read the output (on Visual C++ 2005)
```

Hello, world!

- "Hello world" is a very important program
  - Its purpose is to help you get used to your tools
    - Compiler
    - Program development environment
    - Program execution environment
  - Type in the program carefully
    - After you get it to work, please make a few mistakes to see how the tools respond; for example
      - Forget the header
      - Forget to terminate the string
      - Misspell return (e.g. retrun)
      - Forget a semicolon ;
      - Forget { or }
      - …
  - Typical execution:
    - g++ -o helloWorld HelloWorld.cpp
    - ./helloWorld

```
GPs-MacBook:Chapter02 Giwrgakis$  g++ -o helloWorld HelloWorld.cpp
GPs-MacBook:Chapter02 Giwrgakis$ ./helloWorld
Hello, world!
 GPs-MacBook:Chapter02 Giwrgakis$
```

…few usual errors

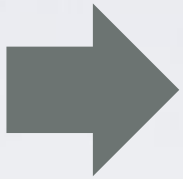
```
// no #include here
int main()
{
        cout << "Hello, World!\n";
        return 0;
}
```

```
#include "std_lib_facilities.h"
int main()
{
        cout << "Hello, World!\n;
        return 0;
}
```

```
#include "std_lib_facilities.h"
integer main()
{
        cout << "Hello, World!\n";
        return 0;
}
```

```
#include "std_lib_facilities.h"
int main()
{
        cout < "Hello, World!\n";
        return 0;
}
```

```
#include "std_lib_facilities.h"
int main()
{
        cout << "Hello, World!\n"
        return 0;
}
```

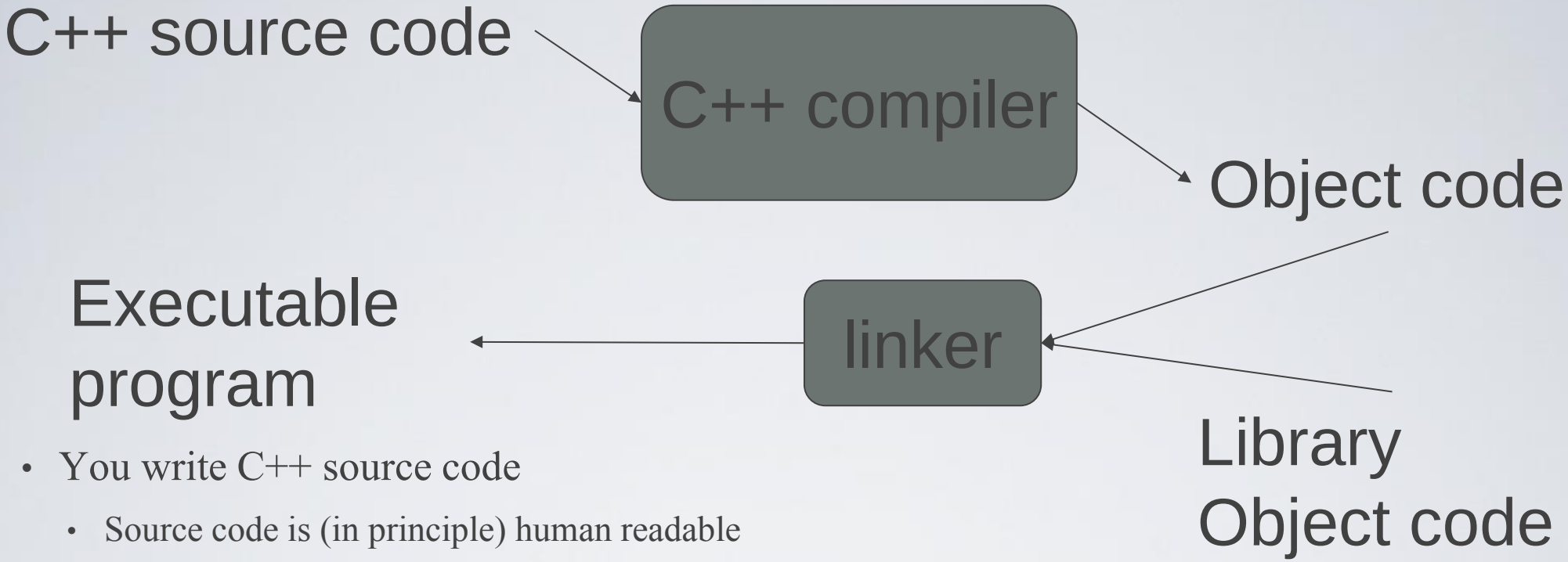


```
#include "std_facilities.h"
int main()
{
        cout << "Hello, World!\n";
        return 0;
}
```

Hello world

- Only cout << "Hello, world!\n" directly does nothing
- That's normal
  - Most of our code, and most of the systems we use simply exist to make some other code elegant and/or efficient
  - "real world" non-software analogies abound
- "Boiler plate," that is, notation, libraries, and other support is what makes our code simple, comprehensible, trustworthy, and efficient.
  - Would you rather write 1,000,000 lines of machine code?
- This implies that we should not just "get things done"; we should take great care that things are done elegantly, correctly, and in ways that ease the creation of more/other software:
  - Style Matters!

Compilation and linking



- You write C++ source code
  - Source code is (in principle) human readable
- The compiler translates what you wrote into object code (sometimes called machine code)
  - Object code is simple enough for a computer to “understand”
- The linker links your code to system code needed to execute
  - E.g. input/output libraries, operating system code, and windowing code
- The result is an executable program
  - E.g. a .exe file on windows or an a.out file on Unix

So what is programming?

- Conventional definitions
  - Telling a very fast moron exactly what to do
  - A plan for solving a problem on a computer
  - Specifying the order of a program execution
    - But modern programs often involve millions of lines of code
    - And manipulation of data is central
- Definition from another domain (academia)
  - A … program is an organized and directed accumulation of resources to accomplish specific … objectives …
    - Good, but no mention of actually doing anything
- The definition we'll use
  - Specifying the structure and behavior of a program, and testing that the program performs its task correctly and with acceptable performance
    - Never forget to check that "it" works
- Software == one or more programs

Programming

- Programming is fundamentally simple
  - Just state what the machine is to do
- So why is programming hard?
  - We want "the machine" to do complex things
    - And computers are nitpicking, unforgiving, dumb beasts
  - The world is more complex than we'd like to believe
    - So we don't always know the implications of what we want
  - "Programming is understanding"
    - When you can program a task, you understand it
    - When you program, you spend significant time trying to understand the task you want to automate
  - Programming is part practical, part theory
    - If you are just practical, you produce non-scalable unmaintainable hacks
    - If you are just theoretical, you produce toys

Things to remember

- What is programming?
- Set-up a development environment
  - Familiarize yourself with the tools
  - Make sure you all compile/run the “HelloWorld.cpp” program by next lecture
- How the compiler and linker work
- For next time: Read Chapters 1 and 2 from Stroustrup

Acknowledgements

Bjarne Stroustrup

Programming -- Principles and Practice Using C++

http://www.stroustrup.com/Programming/

Thank you!

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης