CSC 241 Introduction to Computer Science

Marcus Schaefer

based on presentations by Ljubomir Perkovic and Amber Settle

Computer Science

The branch of knowledge concerned with the construction, programming, operation, and use of computers. (OED)

Students in the program will develop a broad set of skills and expertise

- Programming and software development skills, the technical tools of the IT trade
- An understanding of modern Computer Systems, which you will use to develop computer applications
- Skills in application areas such as security and cryptography, robotics and computer vision, data mining and databases, distributed and mobile systems, intelligent systems and gaming, computational biology, etc.

http://www.cdm.depaul.edu/academics/Pages/BSinComputerScience.aspx

Computational Problem Solving

- Model your problem
  - Graphs, equations, stochastic process …
- Solve the problem
  - Step-by-step algorithm
  - resource efficient (running time, memory)
- Implementation (programming)
  - Programming
  - Software Engineering (large-scale systems

→ Computational Thinking

Areas

Theoretical Computer Science

- Programming languages
- Algorithms and data structures
- Information theory
- Distributed/parallel systems

Applied Computer Science

- Artificial Intelligence
- Computer Architecture
- Operating Systems
- Databases
- Computer Graphics
- Cryptography
- Robotics
- Scientific Computing
- Software Engineering

Application Areas

- Biology (computational biology)
- Finance (computational finance)
- Physics (scientific computing)
- Humanities (computational linguistics, computational philosophy)
- …

Some problems

- search for document on the web
- search for an image
- efficient audio/video streams through the Internet
- encrypt/decrypt communication
- authenticate yourself to a system (tokens)
- mine web-data to make predictions
- customize your news page

Algorithms

- We describe the step-by-step instructions needed to solve a problem as an algorithm
  - An algorithm is an ordered sequence of unambiguous instructions that when executed on a specific input produces the desired output
  - Of course the algorithm needs to be correct, and the more theoretical areas of computer science concern themselves with how to do this
- To specify an algorithm you need to make the problem statement as precise as possible
  - A first step: Clearly specify the input and output

Don’t Do this at Home

EXAMPLE: CREATING A SPAM LIST

Spam list algorithm

Input: URL

Output: A list of e-mail addresses appearing on web pages belonging to the web site in the URL domain

Algorithm (assuming email_list is initially empty):

Spam_List_Algorithm(URL)

1. Download the page at URL
2. Search the URL file for e-mail addresses and add them to the e-mail list

Spam list algorithm

Input: URL

Output: A list of e-mail addresses appearing on the web-page URL and the pages it links to

Algorithm (assuming email_list and link_list are initially empty):

Spam_List_Algorithm(URL)

1. Add URL to link_list
2. As long as link_list is not empty, pick one of the URLs, call it U
    1. Download the file at U
    2. Search the file for e-mail addresses and add them to email_list
    3. Search the file for hyperlinks and add them to link_list
    4. Remove U from link_list

From algorithms to programs

- A person could execute the spam list algorithm
  - Algorithms are best automated
  - Computing and algorithms have existed for thousands of years but the creation of computers has accelerated growth in the field
  - To automate an algorithm we need to write a program

- A program is an algorithm written in a form that the computer can understand

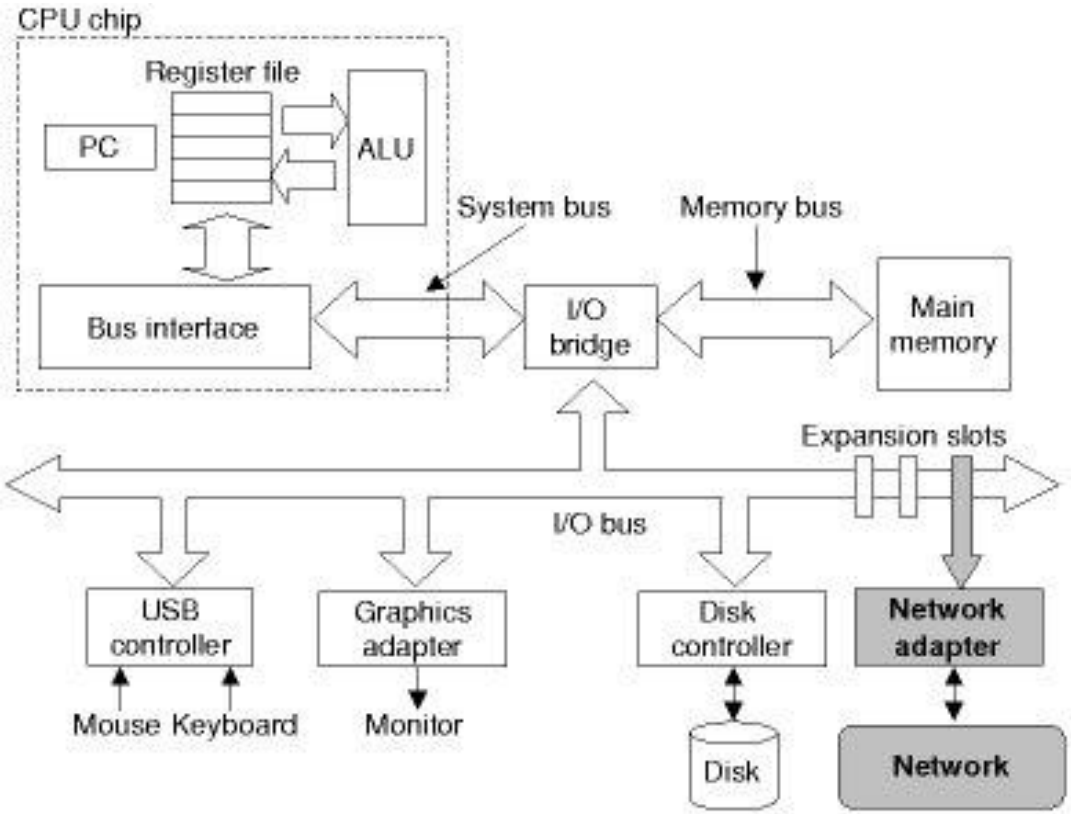
- An algorithm versus a program:
  - Both are precise step-by-step descriptions for solving a problem
  - An algorithm is written for humans and is computer system independent
  - A program uses a language humans can understand but can be executed on a computer system, so that it is dependent on the system, including the hardware, OS, programming language, etc.

COMPUTER SCIENCE
A SHORT OVERVIEW

Computer systems

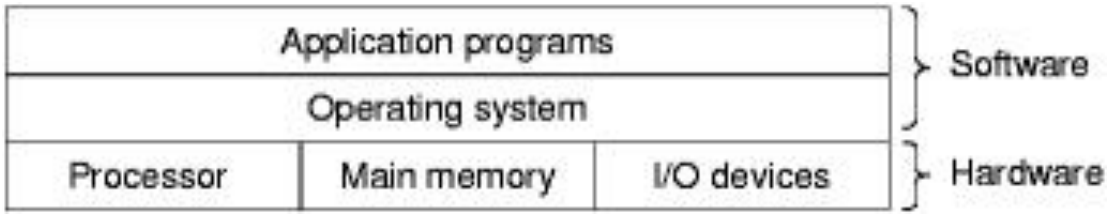
- In order to implement an algorithm as an actual program, a programmer needs an understanding of the systems that will execute the algorithm
- A computer system consists of some or all of the following components:
  - Computer hardware
  - Operating system
  - Network and network protocols
  - Programming languages
  - Application programming interface (API)

Computer hardware



Operating systems

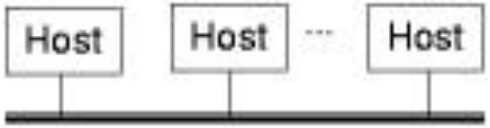
- The operating system is the layer between the hardware and the applications programs
  - Applications do not directly access the keyboard, the disk, the main memory, the network (and Internet), or the display



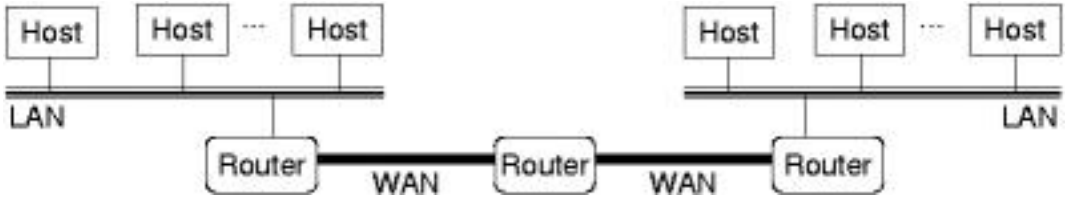
- The operating system has two functions:
  - To protect the hardware from misuse
  - To provide application programs with an interface through which they can manipulate hardware devices

Networks and network protocols

- A network allows communication between computer systems
- Individual computers (hosts) are connected to form a local area network (LAN)

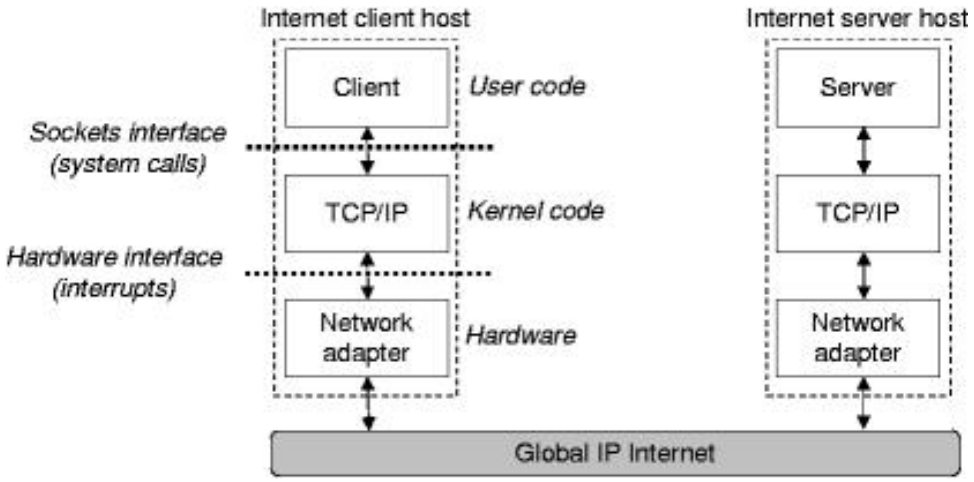


- An internetwork is obtained when several LANs are interconnected
  - The Internet is the most well-known



Internet applications

- The following is a diagram of the hardware and software organization of an Internet application:



- The World Wide Web (consisting of browser clients and web servers) is an example of an application running on the Internet

Programming languages

- Computer applications such as the WWW consist of one or more programs written in some programming language for some architecture/OS/ network system
- A programming language is an artificial language that can be used to control the behavior of a machine
  - Its purpose is to provide instructions to a computer
  - It must be more precise than other forms of human expression
    - Humans understand (mostly) when you speak in an incorrect or ambiguous way
    - Computers are unable to figure out what the programmer intended to write (and don't try)

Programming languages

Ada, Algol, APL, Basic, C, Clojure, C++, C#, Cobol, Eiffel, Forth, Fortran, Haskell, Java, JavaScript, Lisp, Logo, ML, Pascal, Perl, PL/I, Prolog, Python, Ruby, Scala, Scheme, Smalltalk, Visual Basic, VBScript

Hello World

```
PRINT "Hello world!"
```

```
#include <iostream>
using namespace std;
int main() {
        cout << "Hello World!" << endl;
        return 0;
}
```

```
print("Hello World")
```

```
          public class HelloWorld {
             public static void main(String[] args) {
                 System.out.println("Hello world!");
             }
          }
```

```
(print "Hello world!")
```

```
SELECT 'Hello world!' FROM DUAL
```

Programming languages

- There is no single programming language that is best for all tasks
  - If there were, the others wouldn't exist …
  - Each language has its strengths and weaknesses
- The advantage of Python
  - It has simple syntax
    - It is easier to learn than other languages
    - It allows very fast development
  - For example, Google encourages its software developers to build prototypes for new applications in Python
- An application programming interface (API) is a source code interface that a computer application, operating system, or library provides to support requests for services to be made of it by a computer program
  - Example: The Python API includes services to download web pages, search for patterns in files, etc.

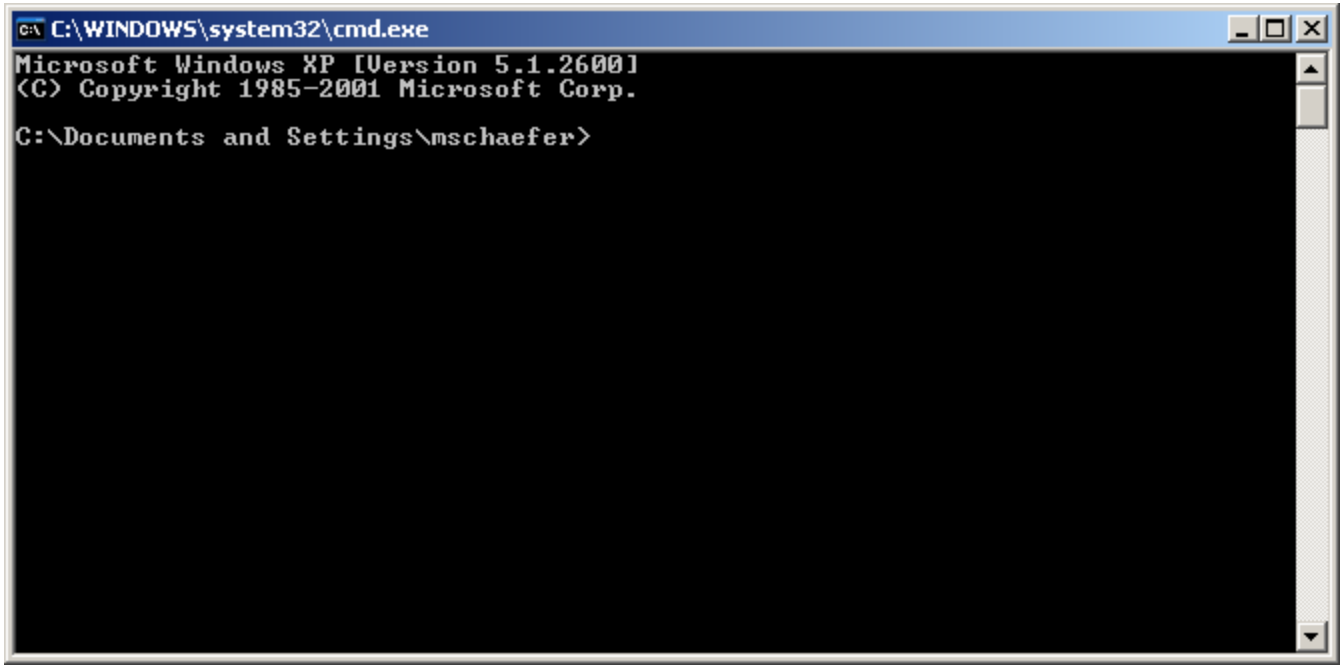
The focus of our class

- We will learn the following things in this class:
  - The Python programming language
  - The Python API
  - How to use Python to solve basic computer science problems (and with it)
  - An overview of the problems that interest computer scientists
- Next we need to understand how to get started writing Python programs

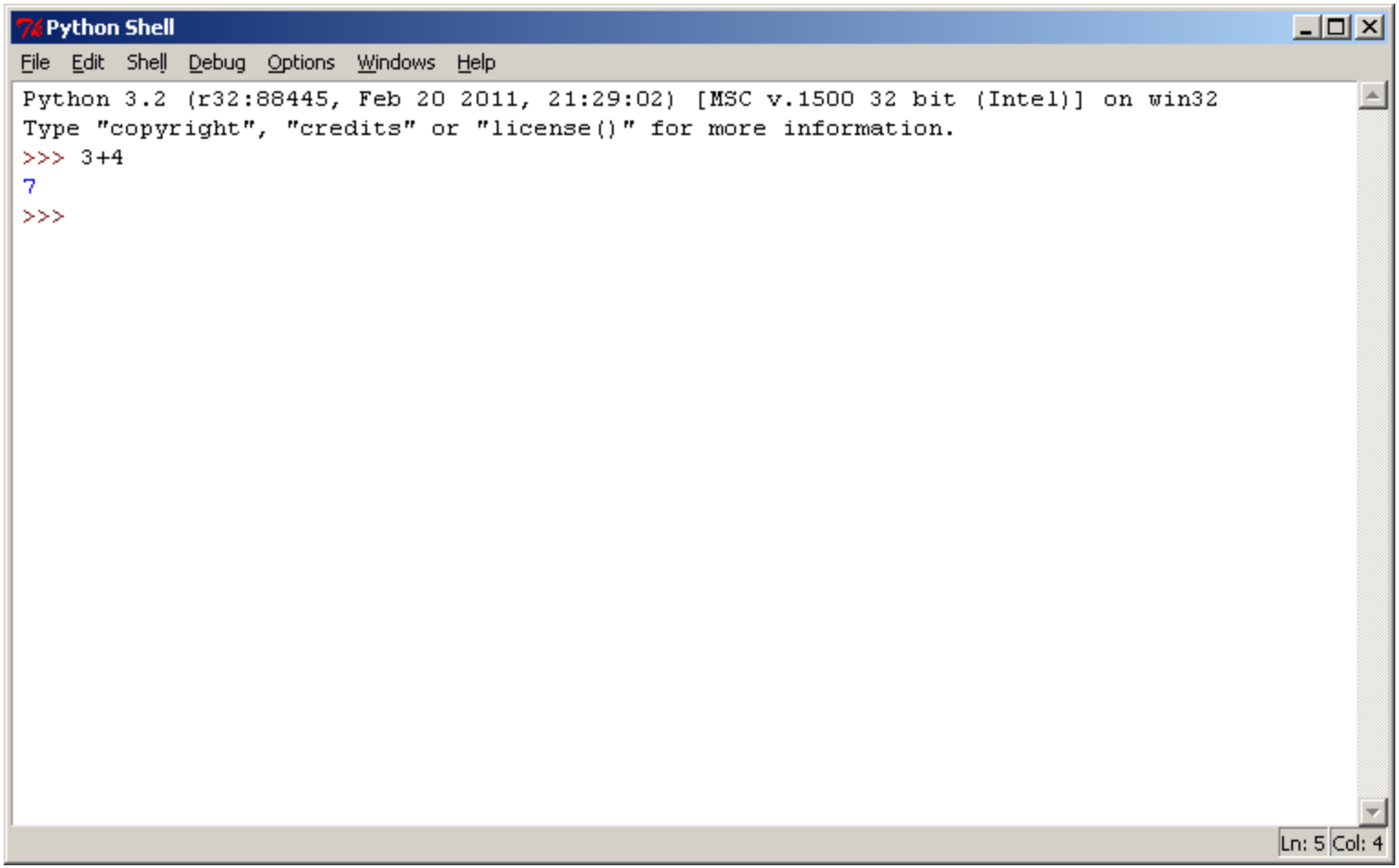
Basic system setup

1. Download the Python software from: http://python.org/download/releases/3.2.2/
2. Create a csc241 directory to store your programs (and subdirectories as needed)
3. Open a command-line window
   - On Windows 7: Start → All Programs → Accessories → Command Prompt
   - Ask if you don't know how to find it on your system
   - We will sometimes run Python programs from the command-line window
4. Open IDLE, i.e. the Python Integrated DeveLopment Environment

Command-Line Window



Idle


Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2 (r32:88445, Feb 20 2011, 21:29:02) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 3+4
7
>>>
Ln: 5 Col: 4

Class Outline

Week	
1	Introduction to computer science and programming
2	Data in Python: Objects and Types
3	Input and Output
4-5	Control Flow
6	Midterm
7	More on Types
8	Modules and Libraries
9-10	Problem Solving and Advanced Topics