CSC 241
Introduction to Computer Science I

Marcus Schaefer
based on work by Ljubomir Perkovic and Amber Settle
Computer Science

The branch of knowledge concerned with the construction, programming, operation, and use of computers. (OED)
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
• Test/Debug
  • Software Engineering (large-scale systems)

→ Computational Thinking
Modeling
Revolving Century
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
- recommend movies
Algorithms

Step-by-step sequence of definite instructions turning an input into an output

Theoretical Computer Science
- finding algorithms (if they exist)
- evaluating efficiency
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

Hello World

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

public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("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: https://www.python.org/downloads/release/python-343/

2. Create a csc241 directory to store your programs (and subdirectories as needed)

3. Open IDLE, the Python Integrated Development Environment
Idle

```python
Python 3.3.0 (v3.3.0:bd8afbb90ebf2, Sep 29 2012, 10:55:48) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> 3+4
7
>>> |
```
## Class Outline

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to computer science and programming</td>
</tr>
<tr>
<td>2</td>
<td>Data in Python: Objects and Types</td>
</tr>
<tr>
<td>3</td>
<td>Input and Output</td>
</tr>
<tr>
<td>4-5</td>
<td>Control Flow</td>
</tr>
<tr>
<td>6</td>
<td>Midterm</td>
</tr>
<tr>
<td>7</td>
<td>More on Types</td>
</tr>
<tr>
<td>8</td>
<td>Modules and Libraries</td>
</tr>
<tr>
<td>9-10</td>
<td>Problem Solving and Advanced Topics</td>
</tr>
</tbody>
</table>