

CSC 221: Introduction to Programming

Fall 2018

History of programming

- developments in hardware
- machine language → assembly language
- high-level languages
- block-based vs. scripting languages
- intro to Python

1

Early programmable devices

the use of algorithms to solve problems can be traced back millennia

- ~2000 BC – Egyptian multiplication
- ~1600 BC – Babylonian mathematical algorithms on clay tablets
- ~300 BC – Euclid's algorithm for GCD

1805 – the first programmable device was Jacquard's loom

- the loom wove tapestries with elaborate, programmable patterns
- a pattern was represented by metal punch-cards, fed into the loom
- using the loom, it became possible to mass-produce tapestries, and even reprogram it to produce different patterns simply by changing the cards



mid 1800's – Babbage designed his "analytical engine"

- its design expanded upon mechanical calculators, but was programmable via punch-cards (similar to Jacquard's loom)
- Babbage's vision described the general layout of modern computers
- he never completed a functional machine – his design was beyond the technology of the day

2

The first "computers"

1930's – several engineers independently built "computers" using electromagnetic relays

- an electromagnetic relay is physical switch, which can be opened/closed via electrical current (widely used in telephone networks at the time)
- Zuse (Nazi Germany), Atanasoff (Iowa State), Stibitz (Bell Labs)
 - Stibitz's MARK I stored 72 numbers, required 1/10 sec to add, 6 sec to multiply still, 100 times faster than previous technology



mid 1940's – vacuum tubes replaced relays

- a vacuum tube is a light bulb containing a partial vacuum to speed electron flow
- vacuum tubes could control the flow of electricity faster than relays since they had no moving parts
- invented by Lee de Forest in 1906



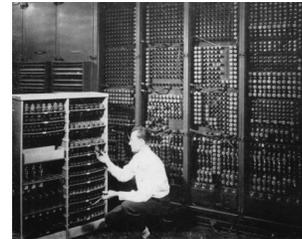
1940's – hybrid computers using vacuum tubes and relays were built

COLOSSUS (1943)

- first "electronic computer", built by the British govt. (based on designs by Alan Turing)
- used to decode Nazi communications during the war

ENIAC (1946)

- first publicly-acknowledged "electronic computer", built by Eckert & Mauchly (UPenn)
- contained 18,000 vacuum tubes and 1,500 relays



3

Stored program computers

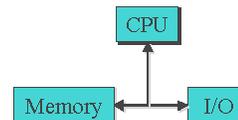
COLOSSUS and ENIAC were not general purpose computers

- could enter input using dials & knobs, paper tape
- but to perform a different computation, needed to reconfigure

von Neumann popularized the idea of a "stored program" computer

- Memory stores both data and programs
- Central Processing Unit (CPU) executes by loading program instructions from memory and executing them in sequence
- Input/Output devices allow for interaction with the user

virtually all modern machines follow this
von Neumann Architecture
(note: same basic design as Babbage)



programming was still difficult and tedious

- each machine had its own machine language, 0's & 1's corresponding to the settings of physical components
- programs developed for one machine could not be run on other types of machines

4

Assembly language example

- still not obvious what it does
- but, clearly easier to work with

what was needed were programming languages closer to the way people think

```
gcc2_compiled.:
    .global _Q_qtod
    .section
    .align 8
.LLC0: .asciz "Hello world!"
    .section
    .align 4
    .global main
    .type main,#function
    .proc 04
main: !#PROLOGUE# 0
    save %sp,-112,%sp
    !#PROLOGUE# 1
    sethi %hi(cout),%o1
    or %o1,%lo(cout),%o0
    sethi %hi(.LLC0),%o2
    or %o2,%lo(.LLC0),%o1
    call __ls__7ostreamPcc,0
    nop
    mov %o0,%i0
    mov %i0,%o0
    sethi %hi(endl__FR7ostream),%o2
    or %o2,%lo(endl__FR7ostream),%o1
    call
__ls__7ostreamPFR7ostream_R7ostream,0
    nop
    mov 0,%i0
    b .LL230
    nop
.LL230: ret
    restore
.LLfel: .size main,.LLfel-main
    .ident "GCC: (GNU) 2.7.2"
```

7

High-level languages

the first high-level languages were developed in the late 1950's

- FORTRAN (1957, Backus at IBM)
- LISP (1959, McCarthy at MIT)
- BASIC (1959, Kemeny at Dartmouth)
- COBOL (1960, Murray-Hopper at DOD)

a high level language can be

translated to machine code by a *compiler* (similar to an assembler) or executed directly by an *interpreter*

```
PROGRAM HELLOWORLD
10 FORMAT (1X,11HELLO WORLD)
WRITE(6,10)
END
```

```
(DEFUN HELLO-WORLD ()
(PRINT (LIST 'HELLO 'WORLD)))
```

```
10 PRINT "HELLO WORLD"
```

```
IDENTIFICATION DIVISION.
PROGRAM-ID. Hello.
ENVIRONMENT DIVISION.
DATA DIVISION.
PROCEDURE DIVISION.
    Display 'Hello, World'.
STOP RUN.
```

mid 1960's - integrated circuits (IC) were produced

- Noyce and Kilby independently developed techniques for packaging transistors and circuitry on a silicon chip
- allowed for mass-producing useful circuitry, eventually entire processors on a single chip (e.g. Intel 4004 in 1971)
- an operating system is a collection of programs that manage peripheral devices and other resources – enabled time-sharing
- as computers became affordable to small businesses, specialized programming languages were developed
 - Pascal (1971, Wirth), C (1972, Ritchie)



8

Personal Computing

late 1970's - Very Large Scale Integration (VLSI)

- manufacturing advances allowed for hundreds of thousands of transistors w/ circuitry on a chip
- resulted in mass-produced microprocessors and other useful IC's
- since computers could be constructed by simply connecting powerful IC's and peripheral devices, they were easier to make and more affordable

1975 - Bill Gates & Paul Allen founded Microsoft

Gates wrote a BASIC interpreter for the first PC (Altair)

1977 - Steve Wozniak & Steve Jobs founded Apple

went from Jobs' garage to \$120 million in sales by 1980

1980 - IBM introduced PC

Microsoft licensed the DOS operating system to IBM

1984 - Apple countered with Macintosh

introduced the modern GUI-based OS (originally developed at Xerox)

1985 - Microsoft countered with Windows



1980's - object-oriented programming began

- represented a new approach to program design which views a program as a collection of interacting software objects that model real-world entities
- Smalltalk (Kay, 1980), C++ (Stroustrup, 1985), Java (Sun, 1995)



Over the last 20+ years

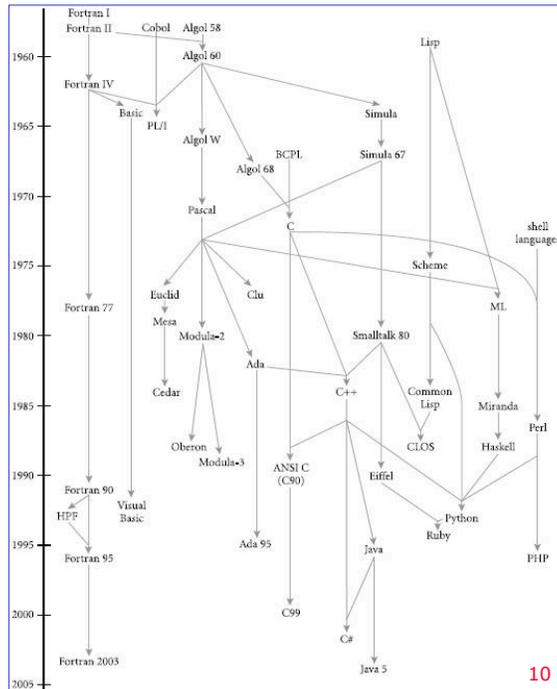
object-oriented programming has dominated large-scale software development

blocks-based languages have proven effective for learning programming basics

- Alice (1994), Scratch (2007)

scripting languages have grown in number and popularity

- typically simpler, interpreted & interactive, often customized for specific niches
- well-suited for solving smaller problems, or gluing together other software
- Python (general purpose, 1989)
- R (data modeling & visualization, 1993)
- PHP (back-end Web development, 1995)
- JavaScript (front-end Web dev., 1995)
- Go (Google development, 2009)
- Swift (iOS development, 2014)



Programming is programming

all programming languages build on basic concepts/commands

- variables
- operations
- conditionals
- loops
- data types
- data structures

Scratch vs. Python →



```
count = 0
count += 1
new_num = count*2 + 1
if new_num > 5:
    print("greater")
if new_num > 5:
    print("greater")
else:
    print("not greater")
for _ in range(10):
    print(random.randint(1, 6))
while count >= 0:
    print(count)
    count -= 1
```

11

Python

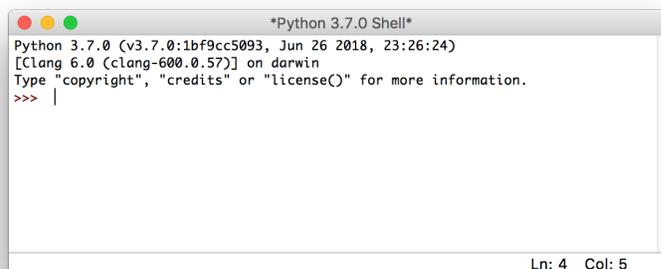
Python is an industry-strength, modern scripting language

- invented by Guido van Rossum in 1989
- has evolved over the years to integrate modern features
v 2.0 (2000), v 3.0 (2008)
- simple, flexible, free, widely-used, widely-supported, upwardly-mobile



Python can be freely downloaded from www.python.org

- the download includes an Integrated DeveLopment Environment (IDLE) for creating, editing, and interpreting Python programs

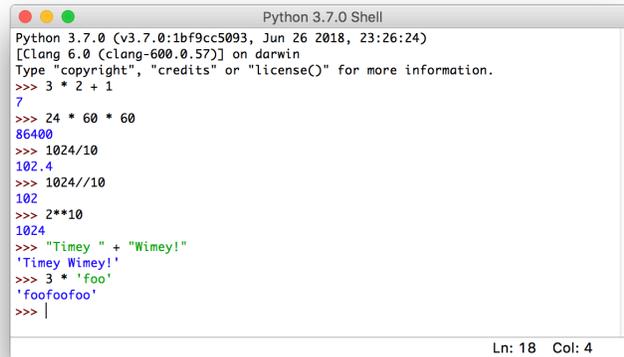


12

Python interpreter

the `>>>` prompt signifies that the interpreter is waiting for input

- you can enter an expression to be evaluated or a statement to be executed



```
Python 3.7.0 Shell
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 26 2018, 23:26:24)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> 3 * 2 + 1
7
>>> 24 * 60 * 60
86400
>>> 1024/10
102.4
>>> 1024//10
102
>>> 2**10
1024
>>> "Timey " + "Wimey!"
'Timey Wimey!'
>>> 3 * 'foo'
'foofoofoo'
>>> |
```

note that IDLE colors text for clarity:

- prompts are reddish
- user input is black (but *strings*, text in quotes, are green)
- answers/results are blue

13

If you want to know more...

check out the following (purely optional) links

[Inventors: The History of Computers](#)

[Computer Museum History Center](#)

[Transistorized! from PBS.org](#)

[Apple Computer Reading List](#)

[The History of Microsoft](#)

[Internet Pioneers: Tim Berners-Lee](#)

[Internet Pioneers: Marc Andreessen](#)

[Wikipedia entry on Programming Languages](#)

[Webopedia entry on Programming Languages](#)

[Python Official Website](#)

14