Introduction to Programming in Python

Dr. Baldassano

Yu's Elite Education

About Me

- Went to college at Princeton for Electrical Engineering

- PhD at Stanford in Computer Science

- Now a research fellow at the Princeton Neuroscience Institute

Goals of the class

- Learn about the power of programming
- Learn how to think like a programmer
- Starting writing your own programs!

Course website

- Go to www.chrisbaldassano.com, click on Teaching, then “Introduction to Programming”

- Has my email address <chrisb@princeton.edu>, schedule and assignments

Why learn to program?

Why learn to program?

- Computers control the world, programmers control computers!

- Harness the power of computers to:
  - Make new software and apps
  - Automate boring tasks
  - Make art, music, and games
  - Command robots
  - Build artificial intelligence systems
  - Get a job

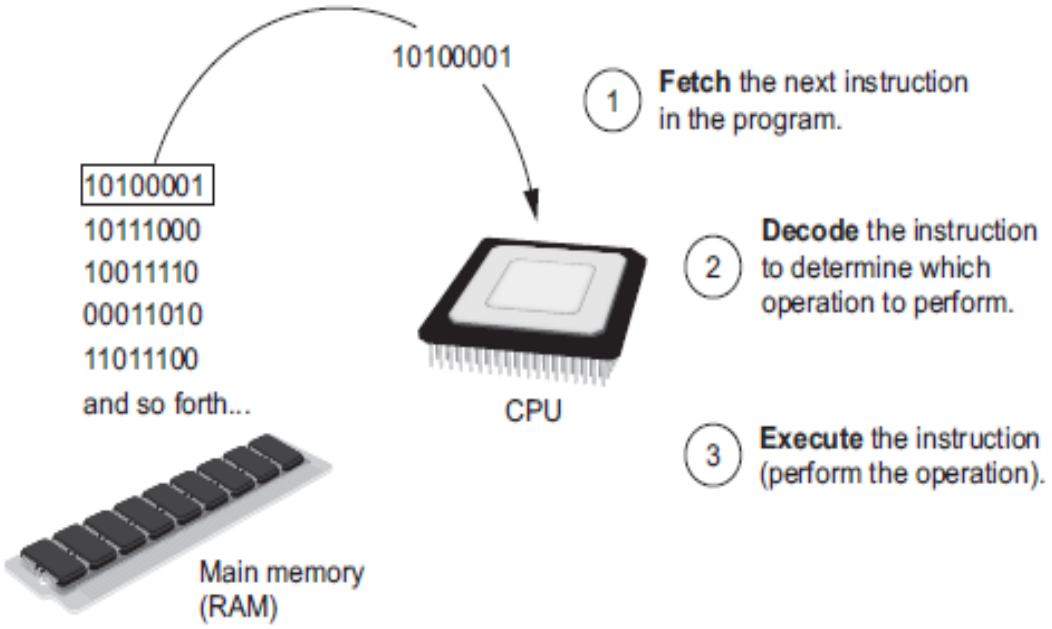
What makes computers so useful?

- Computers can be programmed
  - Designed to do any job that a program tells them to
- Program: set of instructions that a computer follows to perform a task
  - Commonly referred to as *Software*
  - Executed by central processing unit (CPU)
- Programmer: person who can design, create, and test computer programs
  - Also known as software developer

How a Program Works

- Computers perform simple operations on pieces of data
  - Examples: reading data, adding, subtracting, multiplying, and dividing numbers
  - CPU understands instructions written in machine language and included in its instruction set
- To carry out meaningful calculation, CPU must perform many operations
- Executes program in cycle:
  - Fetch: read the next instruction from memory into CPU
  - Decode: CPU decodes fetched instruction to determine which operation to perform
  - Execute: perform the operation

How a Program Works (cont'd.)



From Machine Language to Assembly Language

- Impractical for people to write in machine language
- Assembly language: uses short words (mnemonics) for instructions instead of binary numbers
  - Easier for programmers to work with
- Assembler: translates assembly language to machine language for execution by CPU

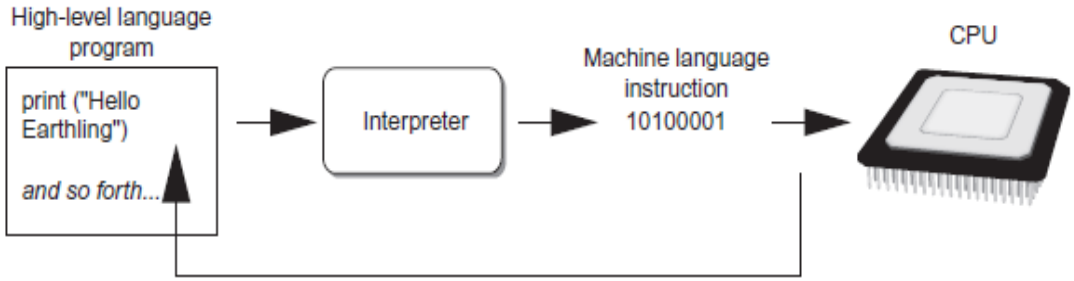
High-Level Languages

- Low-level language: close in nature to machine language
  - Example: assembly language
- High-Level language: allows simple creation of powerful and complex programs
  - No need to know how CPU works or write large number of instructions
  - More intuitive to understand
  - Example: Python!

Why Python?

- Very popular high-level language
- Easy to get started, but also used in complicated real-world systems
- Is an *interpreted* language
  - This means we can enter commands one at a time to try things out, rather than only be able to run complete programs
- Note: We'll be using Python 3 (many online materials still use Python 2)

Interpreters



The interpreter translates each high-level instruction to its equivalent machine language instructions and immediately executes them.

This process is repeated for each high-level instruction.

Let’s try Python!

Quiz

- Which of these statements has the correct syntax?

A. print(Hello World)
B. Print(‘Hello World’)
C. print(‘Hello World)
D. print(‘Hello World’)

Making scripts

- Entering one command at a time is great for testing, but we want to be able to save and run many commands at once

- We can put multiple commands into a script (.py) file

Quiz

- Which program will print X and then Y?

A.

```
# print('X')
# print('Y')
```

B.

```
# adfasdlkasi
print('X')
print('Y')
```

C.

```
print(X)
print('Y')
```

D.

```
# Print some letters
print('X') print('Y')
```

Using variables

- Almost every program needs to keep track of information
- We want to be able to apply the same operation to different pieces of information

- A *variable* is a name we give to a piece of data
- Assign a variable using the assignment operator

  x = ‘World’

Variables (cont'd.)

- In assignment statement, variable receiving value must be on left side
- You can only use a variable if a value is assigned to it
- Rules for naming variables in Python:
  - Variable name cannot be a Python key word
  - Variable name cannot contain spaces
  - First character must not be a letter or an underscore
  - After first character may use letters, digits, or underscores
  - Variable names are case sensitive
- Variable name should reflect its use

Quiz

- Which program will display Hello?

A.

```
print(Hello)
```

B.

```
Hello = 'Hi there'
print(Hello)
```

C.

```
Banana = 'Hello'
print(Banana)
```

D.

```
'Hello' = greeting
print(greeting)
```

Assignment

- Install Python 3.4.3 from https://www.python.org/downloads/
- Write a simple .py script that uses variables and print statements (and run it to make sure it works!)
- Email it to me at chrisb@princeton.edu
- Due before next class
- [FYI: Assignments are also posted at chrisbaldassano.com]

Re-scheduling October 20th class

- I will be at a conference in Chicago on October 20th
- We can reschedule the class for October 21st (Wed), 23rd (Fri) or 26th (Mon)

- Part of your assignment: Email me which of these days you would be available