

General Rules & Regulations

- No Late Arrivals.
- No disturbance in Class
- No use of Mobile Phones(Keep them Silent and in your pockets)
- No Retake of Quizzes/Assignments
- No late submissions
- Student is responsible for his/her short attendance.

Course Plagiarism Policy

- Plagiarism in any kind of assessment including project or sessional/ final exam, assignments quizzes, **will result in F grade in the course.**
- So, what is it?

Plagiarism is **presenting someone else's work or ideas as your own, with or without their consent**, by incorporating it into your work without full acknowledgement. All published and unpublished material, whether in manuscript, printed or electronic form, is covered under this definition.

Books

Textbook:

STARTING OUT WITH PYTHON 4th Edition

by Tony Gaddis

Reference Book:

Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Ipython

by Wes McKinney

Outline

No. of Weeks	Lis of Topics
1	<p>Introduction: What is Artificial Intelligence?</p> <ul style="list-style-type: none">• AI and Data Science hype – and getting past the hype• Why now? – Datafication• Current landscape of perspectives• Skill sets needed <p>AI Computation Challenges.</p>
3	<p>Introduction to Python</p> <ul style="list-style-type: none">• Data Types (conditions + loops)• Print functions• Functions• Built in Data Structures• Regex• Reference/ Trees• OOP• File Handling
2	<p>NumPy</p> <p>Pandas</p>

Outline

No. of Weeks	Lis of Topics
4	<p>Basic Exploratory Data Analysis and the Data Science Process</p> <ul style="list-style-type: none">• Basic tools (plots, graphs and summary statistics) of EDA• Philosophy of EDA - The Data Science Process <p>Plotly</p> <p>Seaborn</p> <p>SciPy</p> <p>SymPy</p> <p>Text Pre-Processing</p> <p> Toolkits for Text processing</p> <p> NLTK</p> <p> SPACY</p> <p>Speech Preprocessing</p> <p> Toolkits for speech processing</p> <p>Image Preprocessing</p> <p> Toolkits for image processing</p>
1.5	Scraping and parsing
1.5	Flask

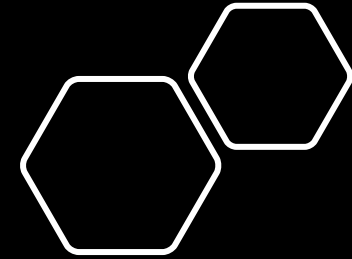
Lecture # 1

Introduction to Computers and Programming



John McCarthy first coined the term
Artificial Intelligence in year 1956

“The science and engineering of making intelligence machines”





Acting humanly

- Proposed by Alan Turing (1950)
- Predicted that by 2000, a machine might have a 30% chance of fooling a lay person for 5 minutes
- Anticipated all major arguments against AI in following 50 years

Systems that Act like Humans

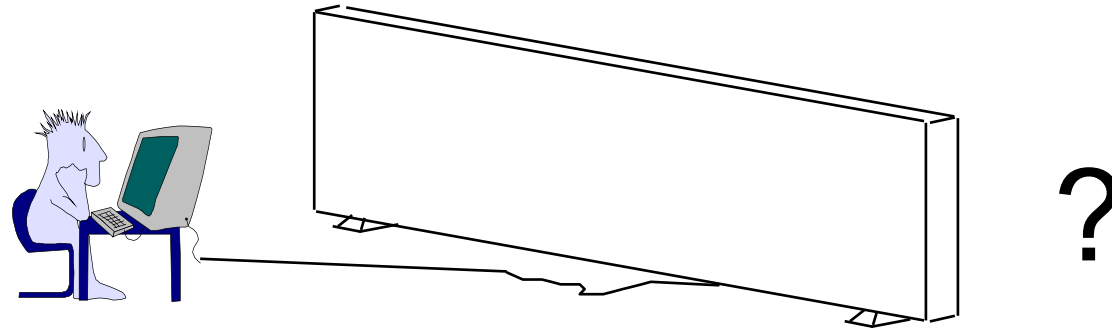
- The Turing Test approach
 - a human questioner cannot tell if
 - there is a computer or a human answering his question, via teletype (remote communication)
 - The computer must behave intelligently
- Intelligent behavior
 - to achieve human-level performance in all cognitive tasks

TURING TEST EXTRA CREDIT:
CONVINCE THE EXAMINER
THAT HE'S A COMPUTER.



Systems that Act like Humans

- You enter a room which has a computer terminal. You have a fixed period of time to type what you want into the terminal, and study the replies. At the other end of the line is either a human being or a computer system.



- If it is a computer system, and at the end of the period you cannot reliably determine whether it is a system or a human, then the system is deemed to be intelligent.

Introduction to Artificial Intelligence

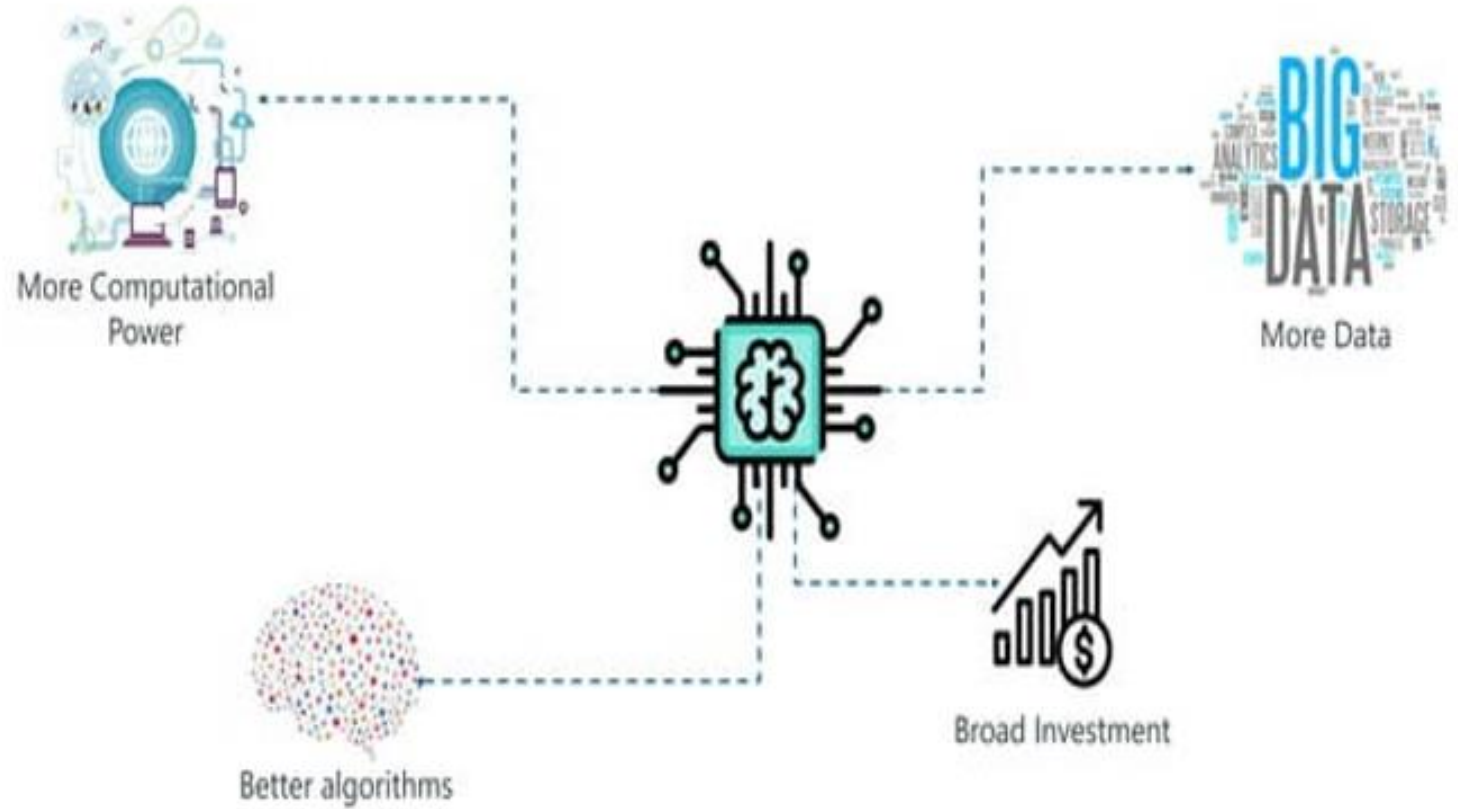
AI systems that improve their performance through experience and data analysis.

AI systems that can analyze and interpret visual information from the world.

AI that can understand, interpret, and generate human language.

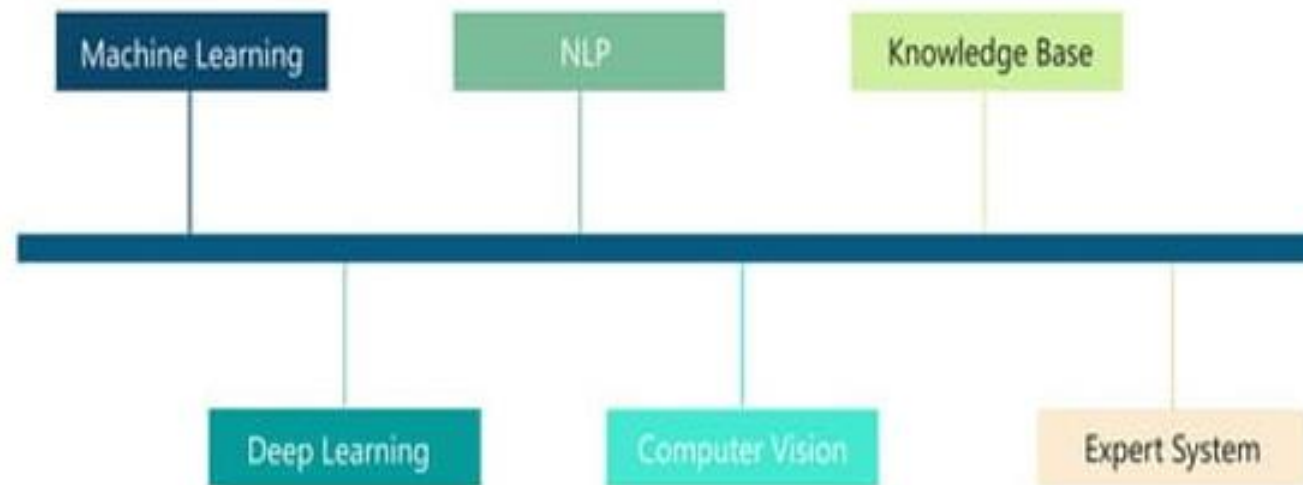
AI-powered machines that can perform physical tasks and interact with the environment.

Demand of AI



What is AI

The Theory and development of computer systems able to perform task normally requiring human intelligence, such as visual perceptions, speech recognitions, decision-making, and translation between the languages.



How Can We Build an AI ?

- **Think About This:**
- What does it take to create an AI system that can learn and adapt?
- Imagine the steps from an idea to a functioning AI - where does programming fit in?
- **Programming for AI:**
- How does a program work?
- Which programming languages are best suited for developing an AI?

How a Program Works

Concept:

A computer's CPU can only understand instructions that are written in machine language. Because people find it very difficult to write entire programs in machine language, other programming languages have been invented.

How a Program Works

- CPU is the most important component in a computer
- CPU is not a brain
- CPU is not smart
- CPU is an electronic device that is designed to do specific things.

How a Program Works

CPU is designed to perform the following operations:

- Read a piece of data from main memory
- Adding two numbers
- Subtracting one number from another number
- Multiplying two numbers
- Dividing one number by another number
- Moving a piece of data from one memory location to another
- Determining whether one value is equal to another value

How a Program Works

- CPU only understands instructions written in **machine language**
- Machine language instructions are written in 1s and 0s
- The entire set of instructions that a CPU can execute is known as the CPU's **instruction set**
- Each brand of microprocessors (Intel, AMD, and Motorola) has a unique instruction set

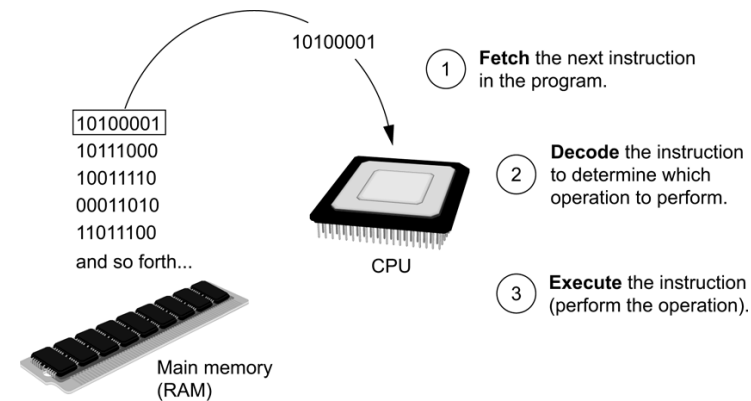
How a Program Works

Fetch-decode-execute cycle is the term used when the CPU executes the instructions in a program.

The cycle consist of three steps:

- Fetch
- Decode
- Execute

Figure 1-17 The fetch-decode-execute cycle



How a Program Works

From Machine Language to Assembly Language

Computers only understand machine language

Machine language is difficult to write

Assembly language uses short words that are known as mnemonics

Assembler is used to translate an assembly language program to machine language

1-20

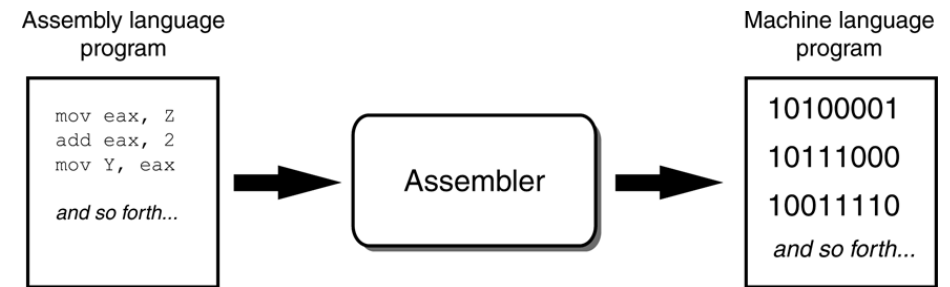


Figure 1 An assembler translates an assembly language program to a machine language program

How a Program Works

High-Level Languages

- Assembly language is referred to as a low-level language

- High-level languages allow you to create powerful and complex programs without knowing how the CPU works, using words that are easy to understand.

For example:

Ada, BASIC, Python, C++, Ruby, Visual Basic

Do you know of any other high-level computer programming languages?

1-21

How a Program Works

Key Words, Operators, and Syntax: an Overview

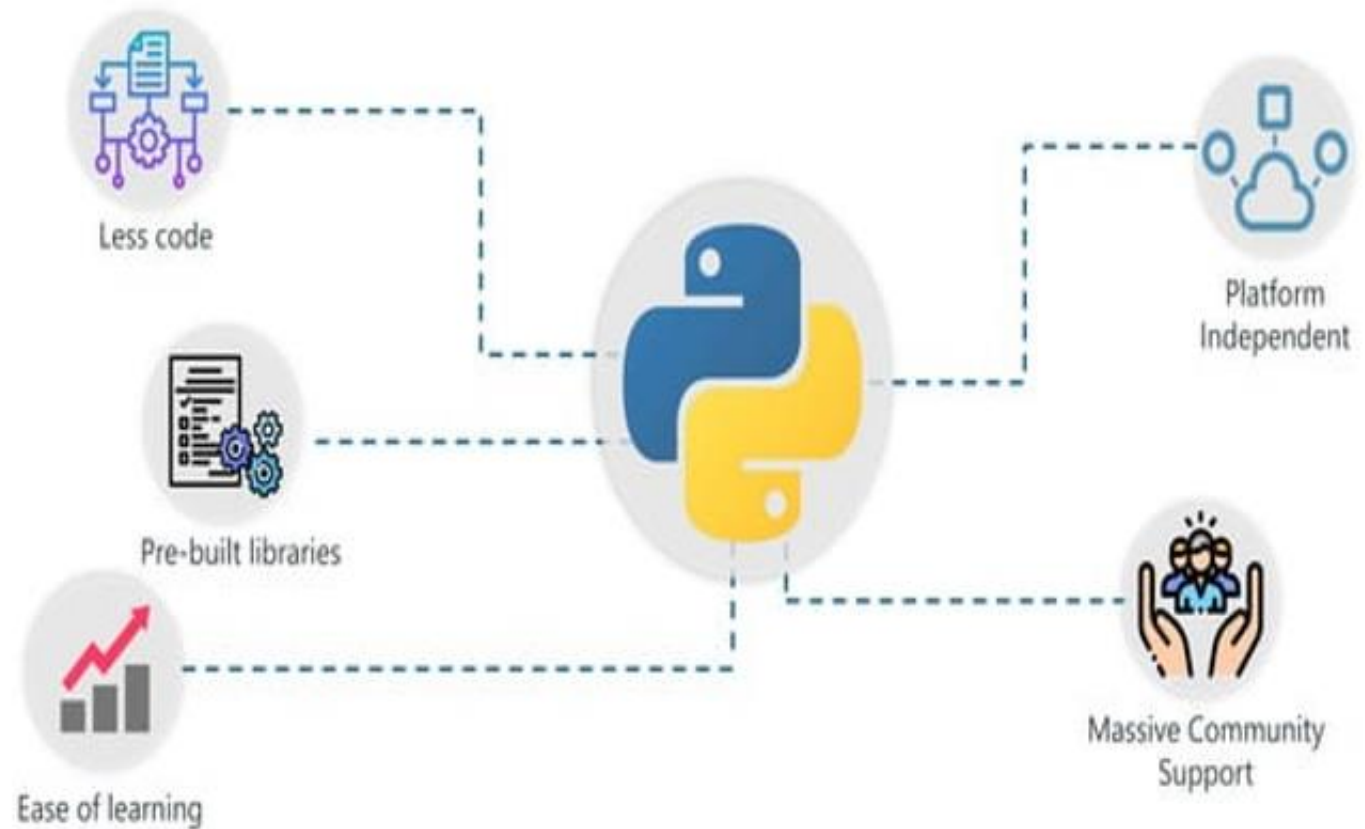
- **Key words** or **reserved words** have specific meaning and purpose in the programming language
- **Operators** perform various operations on data
- **Syntax** is a set of rules that must be strictly followed when writing a program
- **Statements** are individual instructions written in a programming language

How a Program Works

Compilers and Interpreters

- The statements written in a high-level language are called **source code** or simply **code**
- Source code is translated to machine language using a compiler or an interpreter
- **Syntax error** is a mistake such as a:
 - Misspelled word
 - Missing punctuation character
 - Incorrect use of an operator

Why Python for AI



Using Python

Concept:

The Python interpreter can run Python programs that are saved in files, or can interactively execute Python statements that are typed at the keyboard. Python comes with a program named IDLE that simplifies the process of writing, executing, and testing programs.

Using Python

The Python Interpreter

- A program that can read Python programming statements and execute them is the **Python interpreter**
- Python interpreter has two modes:
 - **Interactive** mode waits for a statement from the keyboard and executes it
 - **Script** mode reads the contents of a file (**Python program** or **Python script**) and interprets each statement in the file

Using Python

Interpreter Mode

Invoke Python interpreter through Windows or command line

>>> is the prompt that indicates the interpreter is waiting for a Python statement

```
>>> print 'Python programming is fun!'
```

```
Python programming is fun!
```

```
>>>
```



Statements typed in interactive mode are not saved as a program

Using Python

Writing Python Programs and Running Them in Script Mode

Use a text editor to create a file containing the Python statements

Save the file with a **.py** extension

To run the program:

```
>>> python test.py 
```

Using Python

The IDLE Programming Environment

Integrated DeveLopment Environment (**IDLE**)

- Automatically installed when Python language is installed
- It has a built-in text editor
- IDLE editor colorizes code

Figure 1-21 IDLE

