Overview of Programming Languages
In the hardware module, you saw the basic execution of a program

“Programs” are written in specific languages

Each language has its strengths and weaknesses, and own “grammar” and syntax

We will take a high-level look at several languages
Assembly Language

- Existed almost since the beginning of computers
- Low-level language
  - Virtually machine language
  - Associate a symbolic name to the machine language instructions

```
.model small
.stack
.data
message   db "Hello world!!!", "$"
.code
main proc
  mov ax,seg message
  mov ds,ax
  mov ah,09
  lea dx,message
  int 21h

  mov ax,4c00h
  int 21h
main  endp
end main
```
FORTRAN (1954-1958)

- **FORmula TRANslator**
- Developed by John Backus as an alternative to assembly language
- Dedicated to mathematical calculations
- Still in use today

```plaintext
PROGRAM HELLO
PRINT *, 'Hello, World!'
END
```
LISP (1958-1960)

- **LIS**t Processing

- Invented by John McCarthy at MIT

- Commonly used in Artificial Intelligence

(write-line "Hello World!")
Algol (1960)

- **ALGO**rithmic Language
- First machine independent language
- Developed by an international committee and had a standard
- Introduced block structured programming

```plaintext
BEGIN
  print("Hello World!")
  print(newline)
END
```
COBOL (1960)

- **COmmon Business Oriented Language**
- Used in business, finance, and administrative systems
- Very wordy – tried to be natural
- Lots of legacy programs exist in COBOL

```cobol
IDENTIFICATION DIVISION.
PROGRAM-ID. HELLO-WORLD.

ENVIRONMENT DIVISION.

DATA DIVISION.

PROCEDURE DIVISION.
DISPLAY "Hello, World!".
STOP RUN.
```
BASIC (1964)

- Beginner’s All-purpose Symbolic Instruction Code
  - Was designed to be easy to learn.
  - This is the language Gates and Allen implemented for the Altair

```plaintext
10 PRINT "Hello, World!"
20 END
```
Why the name C?
Because it came after B!!

- Designed as a language to write system software
- Was used to write the first Unix implementations

```c
#include <stdio.h>

/* Hello */
int main(void)
{
    printf("Hello, World!"_ordered);
    return 0;
}
```
SQL (1970s)

- **Standard Query Language**

- Query language for relational databases

- I would not call this a general purpose programming language

```sql
CREATE TABLE message (text char(15));

INSERT INTO message (text) VALUES ('Hello, World!');

SELECT text FROM message;

DROP TABLE message;
```
C++ (1981-1986)

- One step better than C!!
- Object-oriented version of C.
- Commonly used in industry

```cpp
#include <iostream>

int main(){
  std::cout <<
     "Hello, World!\n";
}
```
**HTML (1989)**

- **HyperText Markup Language**

- Again not really a programming language

- Describes how a page should be displayed

```html
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```
Python (1991)

- Not named for the snake. Can you guess?
- Popular scripting language

```python
print "Hello, World!"
```
Java (1994)

- Originally Oak marketing wanted Java
- Portable language – write once run anywhere

```java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!";
    }
}
```
Javascript (1995)

- Scripting language used to embed code into web pages
- Shares the syntax of C or Java but is not Java
- Interpreted by the web browser

```html
<script language="JavaScript">
    document.write('Hello, World!');
</script>
```
- In music a sharp indicates a high-step higher pitch

- Developed by Microsoft

- Intended to be a simple, modern, general-purpose, object-oriented programming language.

```csharp
using System;

class HelloWorld{
    static void Main() {
        System.Console.WriteLine("Hello, World!");
    }
}
```
# TIOBE Index (August 2010)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>==</td>
<td>Java</td>
<td>17.915%</td>
<td>-1.47%</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>==</td>
<td>C</td>
<td>17.147%</td>
<td>+0.29%</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>↑</td>
<td>C++</td>
<td>9.812%</td>
<td>-0.18%</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>↓</td>
<td>PHP</td>
<td>8.370%</td>
<td>-1.79%</td>
<td>A</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>==</td>
<td>(Visual) Basic</td>
<td>5.797%</td>
<td>-3.40%</td>
<td>A</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>↑</td>
<td>C#</td>
<td>5.016%</td>
<td>+0.83%</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>↑</td>
<td>Python</td>
<td>4.583%</td>
<td>+0.65%</td>
<td>A</td>
</tr>
<tr>
<td>8</td>
<td>18</td>
<td>↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑</td>
<td>Objective-C</td>
<td>3.368%</td>
<td>+2.78%</td>
<td>A</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>↓↓↓↓↓</td>
<td>Perl</td>
<td>2.447%</td>
<td>-2.08%</td>
<td>A</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>==</td>
<td>Ruby</td>
<td>1.907%</td>
<td>-0.47%</td>
<td>A</td>
</tr>
<tr>
<td>Program/Application</td>
<td>Lines of Code</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Open Solaris</td>
<td>9.7 million</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linux 2.6.32</td>
<td>12.6 million</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows 2000</td>
<td>35-60 million</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mac OS X 10.4</td>
<td>86 million</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boeing 777</td>
<td>8-10 million</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Space Shuttle</td>
<td>420K (shuttle) 1.4 million ground</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Space Station</td>
<td>40 million</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mozilla</td>
<td>2 million</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Why So Many Languages?

- Evolution
  - Computer Science is still young
  - When I went to school...

- Special Purpose
  - Many new problem domains
  - Animation

- Personal Preference
Categories of Languages

- Declarative
  - Describe a problem rather than defining a solution

- Imperative
  - Describes how to solve a problem

- Scripting
  - A list of commands that can be executed without user interaction. Typically interpreted.

- Object Oriented
  - Programs consist of interacting objects

- Functional
  - Define programs as mathematical functions
The only language that a computer understands is machine language.

A program, written in a programming language other than machine language, must be translated into machine language in order for it to run on a computer.

Programs are typically either compiled or interpreted.
Compilation

Source Program → Compiler → Target Program

Input → Target Program → Output
Traditional Compilation

Source Program → Compiler → Assembly Language → Assembler → Machine Language
Interpretation

Source Program

Interpreter

Input

Output
Subject/predicate: All sentences are about something or someone. The something or someone that the sentence is about is called the *subject* of the sentence. In the following sentences the subjects are shown in red. Note how the subject is often, but not always, the first thing in the sentence.

- **John** often *comes late to class*.
- **My friend and I** both have a dog named **Spot**.
- **On Saturdays** I never get up before **9 o'clock**.
- **Before giving a test the teacher should make sure that the students are well-prepared**.
- **Lying on the sofa watching old films** is **my favorite hobby**.
Do Any of these make sense?

- John often is my favorite hobby.
- On Saturdays I never get up before 9 o'clock.
- Before giving a test the teacher have a dog named Spot.
- Lying on the sofa watching old films is often late to class.
When using a programming language you must understand both the syntax and the semantics of the language.

- Syntax refers to the rules you must follow to form valid statements in the language.
  - What a program should look like
- Semantics describe the meaning of a statement in a language.
  - What a program means