

EECS 368

Programming Language Paradigms

David O. Johnson

Fall 2022

Reminders

- Assignment 1 due: 11:59 PM, Wednesday, September 7
- Assignment 2 due: 11:59 PM, Monday, September 19

Any Questions?

Evolution of Programming Languages

- Machine Language
- Assembly Language
- High-Level Language

Machine Language Coding

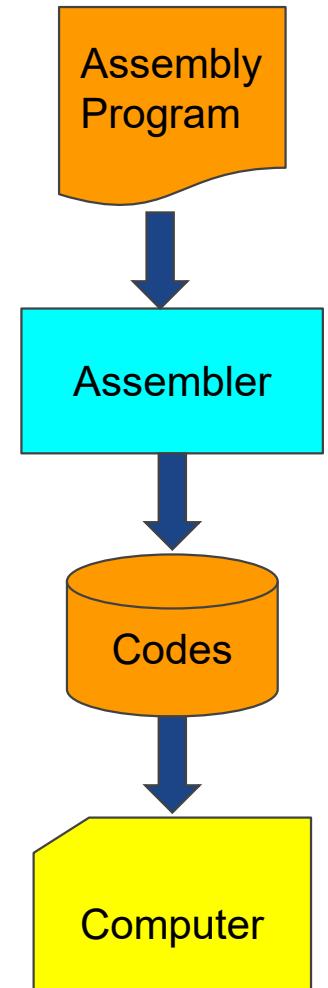
- Machine Code(s)
- Programming computer by literally giving the codes to perform operations.
- Examples are
 - moving data (0x37)
 - adding data (0x17)
 - comparing data (0x28)
 - storing data to tape (...)
- This was interacting with the machine on its terms, 1s and 0s.

Assembly Language Coding

- Programing computer by using mnemonics instead of numerical code.
- Examples are
 - moving data (`ld r1,r2`)
 - adding data (`add r1,r2,r3`)
 - comparing data (`cmp r1,r2`)
 - goto to another set of instructions (`goto label_44`)
- This is slightly better.
 - A transliteration that is easier for humans to understand/remember.
 - Labels eliminated programmer having to remember memory address of where code and data was stored
- Still a one-to-one mapping to machine code.

What an Assembler Does

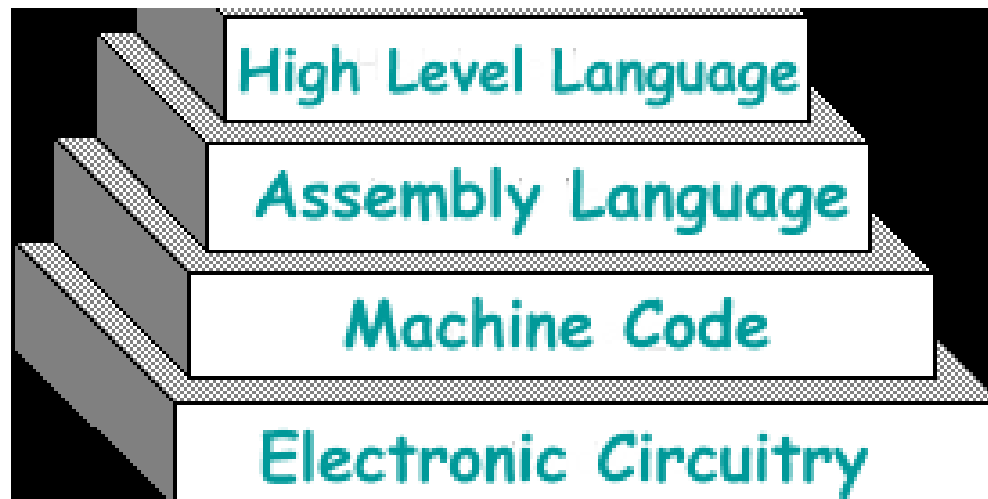
- Runs a program to generate a program
- This is one of the key ideas in Computer Science



Any Questions?

High-Level Language Coding

- A high-level language is a programming language such as C++ or Python that enables a programmer to write programs that are more or less independent of a particular type of computer.
- Such languages are considered high-level because they are closer to human languages and further from machine languages.

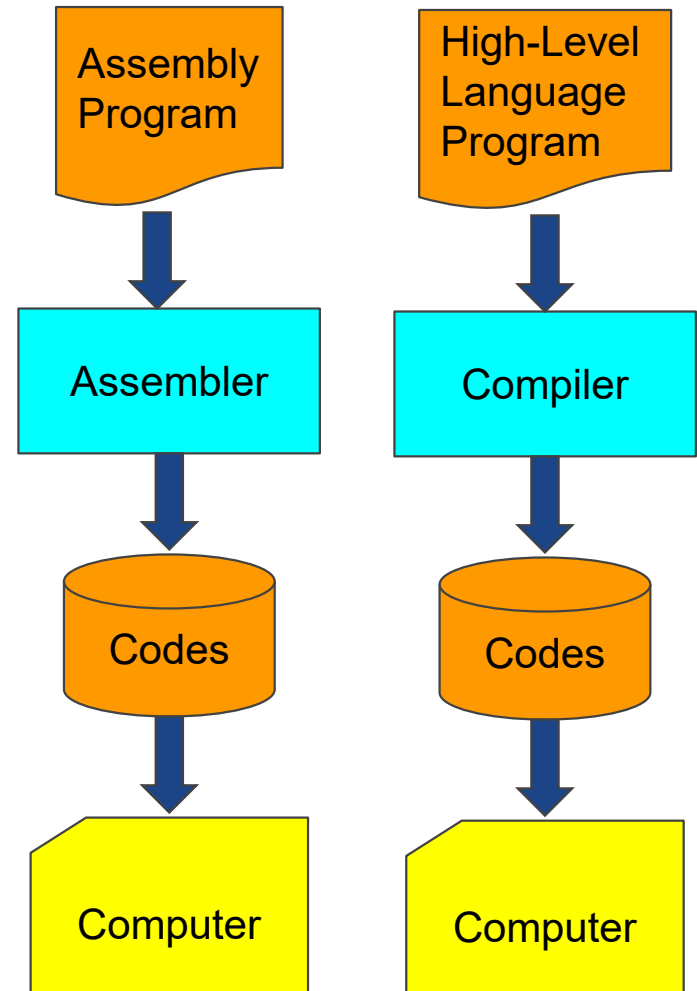


Domain Specific Languages (DSL)

- A Domain-Specific Language (DSL) is a high-level language specialized to a particular application domain.
 - More on this later.
- This is in contrast to a general-purpose language (GPL), which is broadly applicable across domains.
 - C++
 - Python

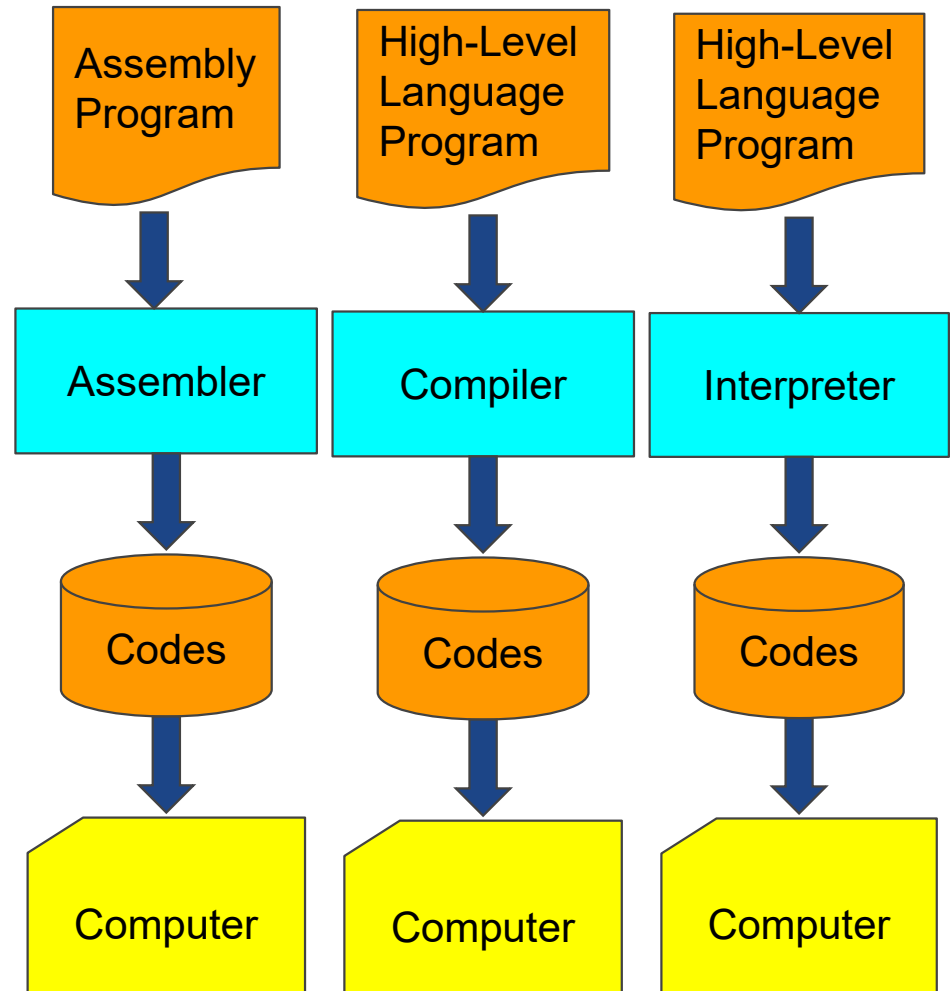
What a Compiler Does

- Runs a program to generate a program
- The same as an Assembler
- C++ is a compiled language



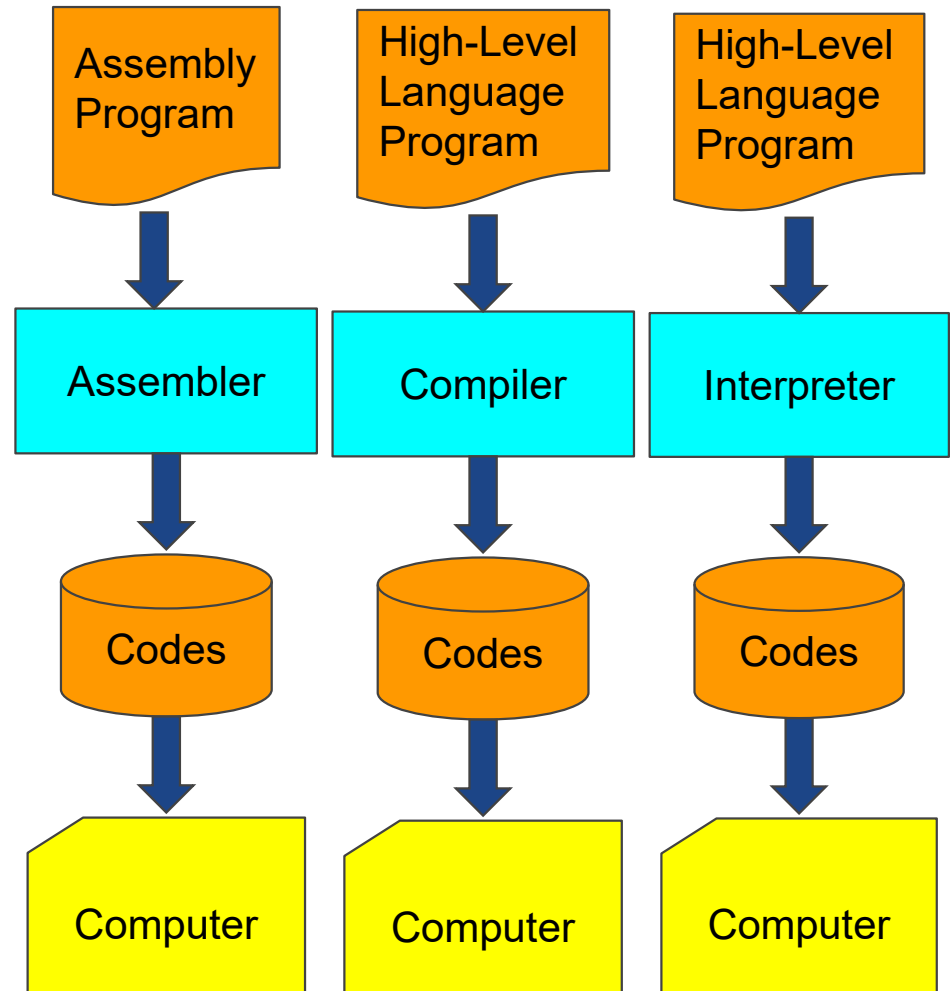
What an Interpreter Does

- Runs a program to generate a program
- The same as a Compiler
- Python is an interpreted language

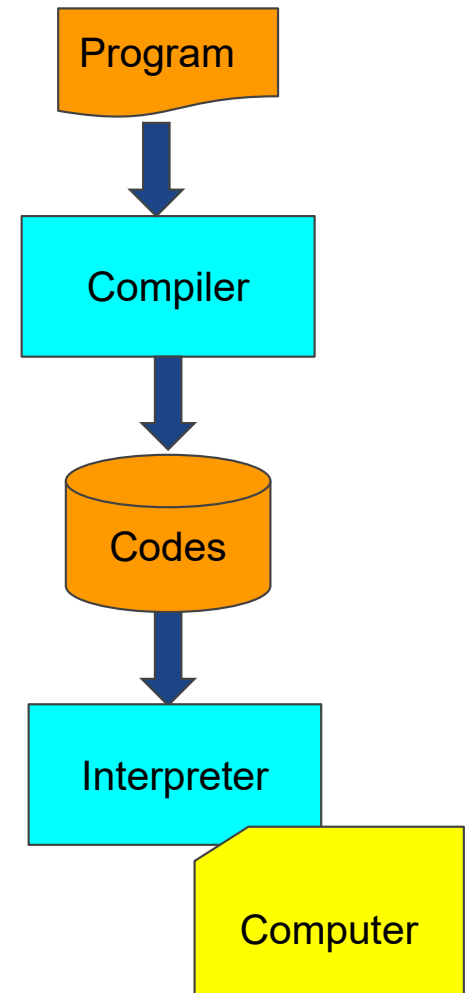
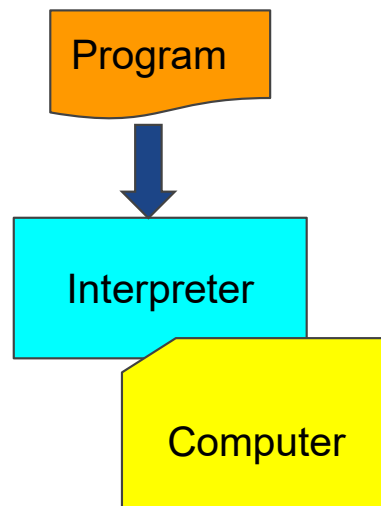
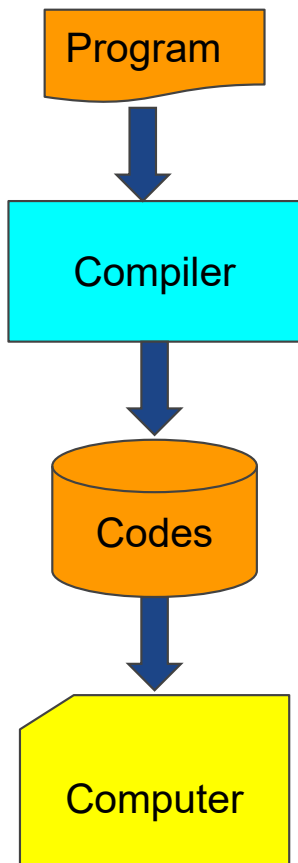


Compiler vs. Interpreter

- So, what is the difference between a compiler and an interpreter?
 - When the program is translated to machine code
- Compiler:
 - Write the entire program
 - Compile it
 - Test it
- Interpreter:
 - Write each line of code and test it



Not That Simple



Other EECS Courses

- In EECS 662 - Programming Languages, you will write an Interpreter.
- In EECS 665 - Compiler Construction, you will obviously write a Compiler.

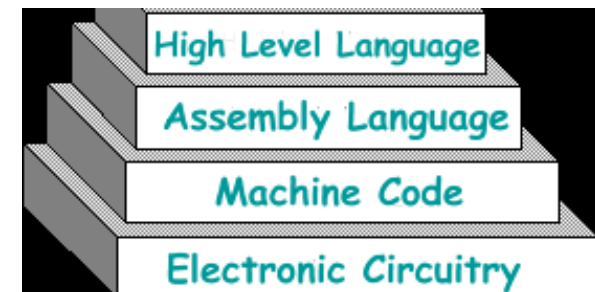
Any Questions?

Imperative vs. Declarative Programming Languages

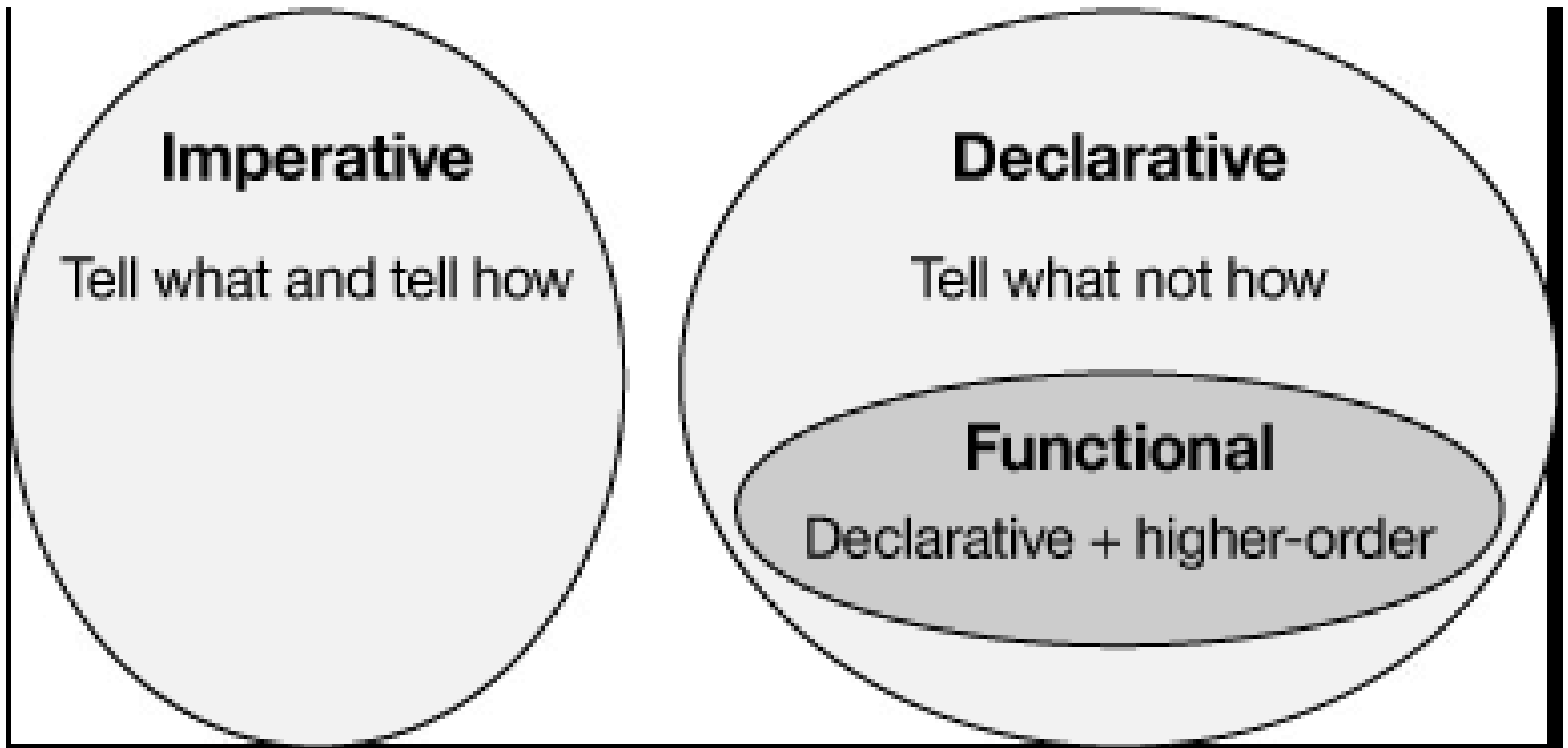
	Imperative	Declarative
Programming Style	Perform step-by-step tasks and manage changes in state	Define what the problem is and what data transformations are needed to achieve the solution
Task	Tell the computer what to do	Describe what you want as <u>an end result</u>
Programmer Focus	how to execute	what to execute
Primary flow control	Iterations, loops, conditionals, and function/method calls	Function calls (including recursion)
Primary manipulation unit	Instances of class or structures	Function as first-class object and data collections
Notable language (for contrasting the style)	Assembly language, C++, Python, JavaScript	SQL, Haskell, Erlang
Real life example	<i>Go two blocks, make a right turn, proceed for three blocks, arrive at airport</i>	<i>Go to nearest airport, please</i>

Declarative Programming

- You probably understand what Imperative Programming is from your classes in C++ and Python.
- But what is Declarative Programming?
- Declarative programming is a method to abstract away the control logic (e.g., if-then-else, looping) required for software to perform an action.
- Declarative programming is a non-imperative style of programming in which programs describe their desired results without explicitly listing commands or steps that must be performed.
- Instead, it involves stating what the task or desired outcome is.
- Declarative programming is a high-level programming concept, which is the opposite of imperative programming.
- Declarative programming languages:
 - Sometimes referred to as 4th generation languages.
 - Sometimes described as above high-level languages

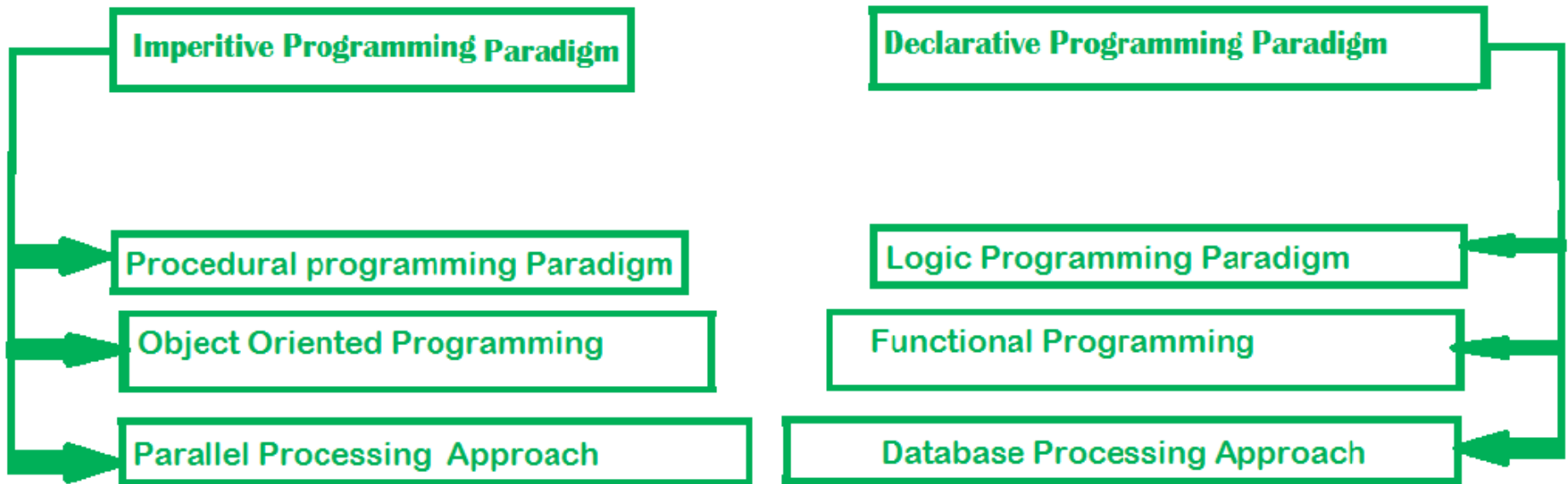


Imperative vs. Declarative Programming Languages

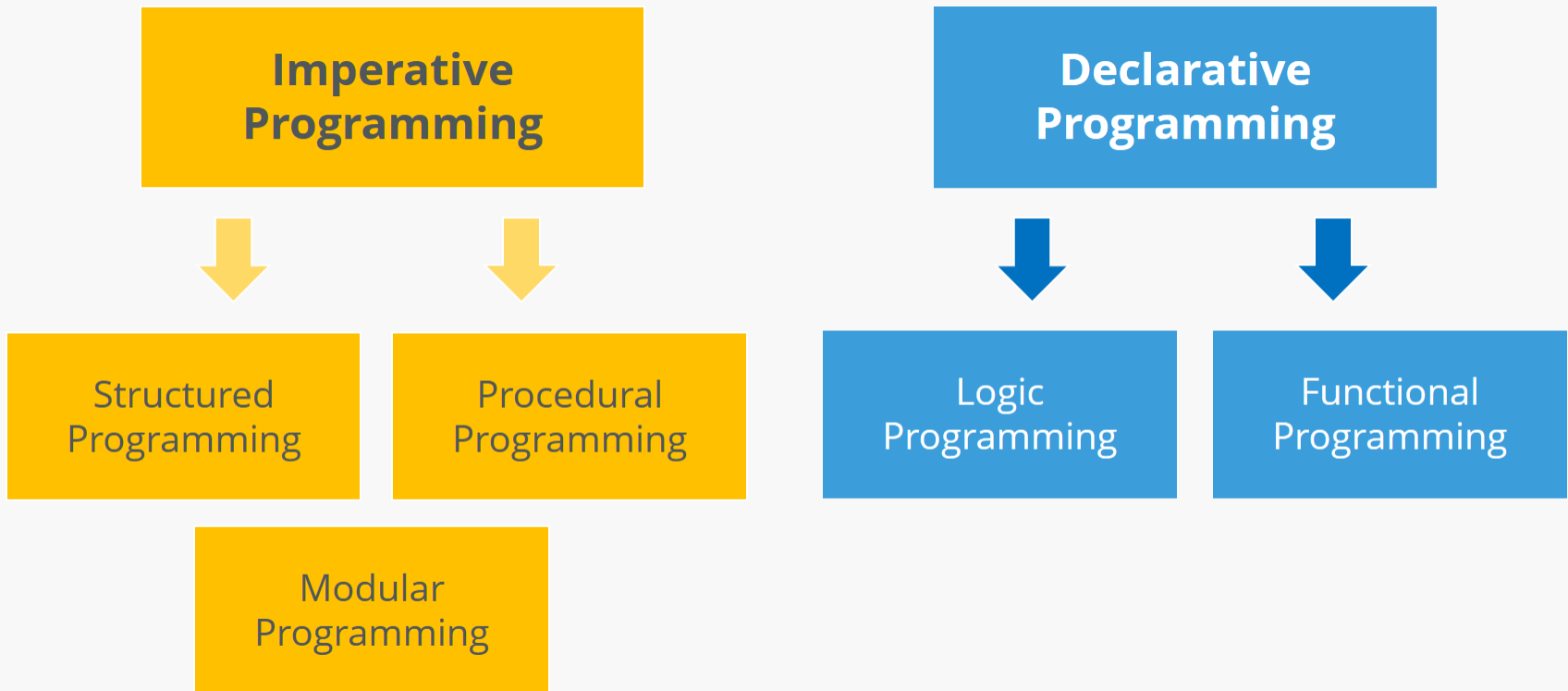


Other Programming Paradigms

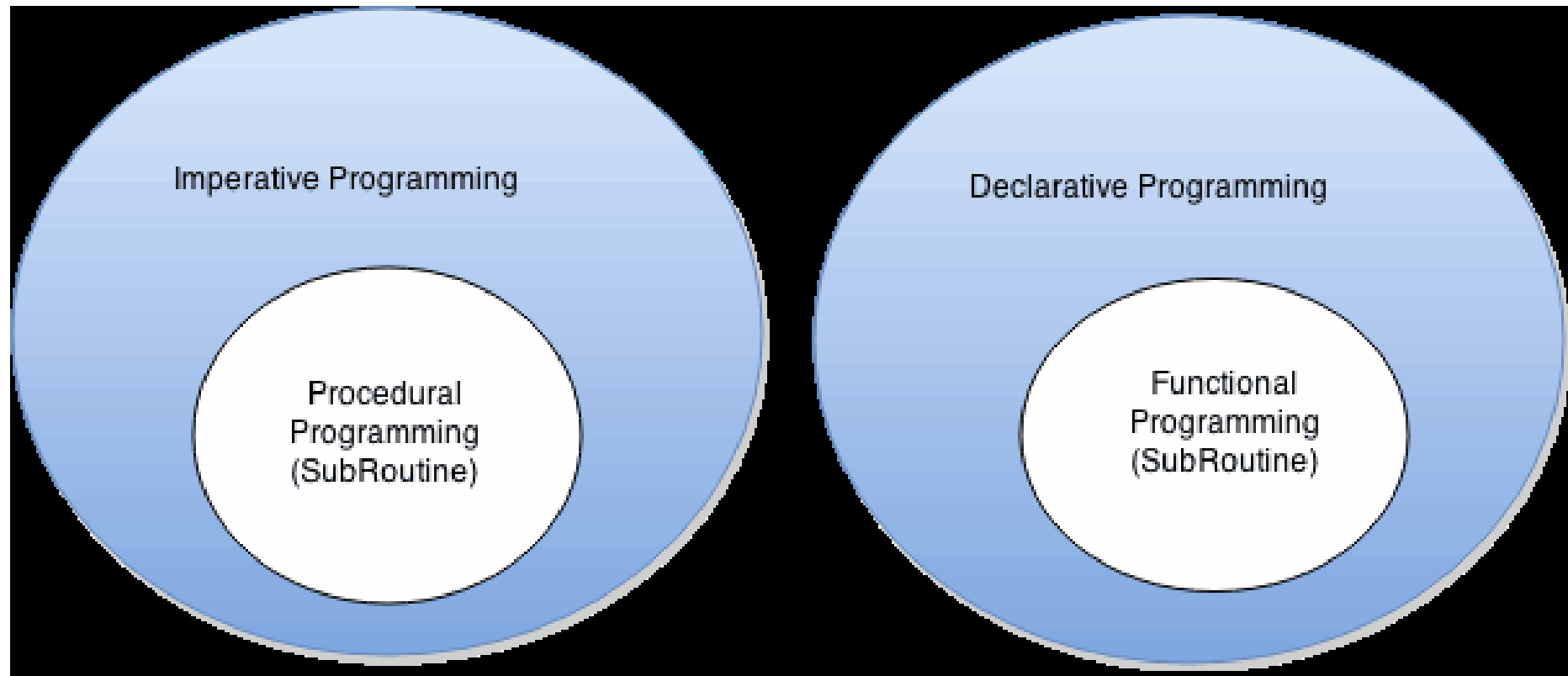
Programming Paradigms



Other Programming Paradigms



Other Programming Paradigms



Any Questions?

EECS 368

- We are going to look at two languages in detail
 - JavaScript (Imperative)
 - Haskell (Declarative – Functional)
- We will also look at Cloud Computing from a language point of view.
- We will study several DSLs:
 - JavaScript Support Vector Graphics (SVG)
 - JavaScript Canvas
 - Haskell QuickCheck

Tools for Understanding Languages

- **Syntax** - precisely describing what a language is
- **Semantics** - precisely describing what a language does
 - Static Semantics
what happens at compile time
 - Dynamic Semantics
what happens at runtime

Syntax vs. Semantics

Syntax

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello World";
    return 0;
}
```

Semantics

- The C++ language defines several headers, which contain information that is either necessary or useful to your program. For this program, the header `<iostream>` is needed.
- The line `using namespace std;` tells the compiler to use the `std` namespace. Namespaces are a relatively recent addition to C++.
- The line `int main()` is the main function where program execution begins.
- The next line `cout << "Hello World";` causes the message "Hello World" to be displayed on the screen.
- The next line `return 0;` terminates `main()` function and causes it to return the value 0 to the calling process.

Any Questions?

In-Class Problem

1. In your own words, describe what an assembler does.
2. In your own words, describe what a compiler does.
3. In your own words, describe what an interpreter does.
4. In your own words, describe what an imperative programming language is.
5. In your own words, describe what a declarative programming language is.
6. In your own words, describe what a domain specific language is.
7. Which of the following is an imperative language? Which is a declarative language?
 - a) C++
 - b) Python
 - c) JavaScript
 - d) Haskell

- You may work together.
- You may ask me or one of the SIs for help.
- Submit your solution as a PDF to Canvas before 11:59 PM today.
- You may leave when you are done.