

# Programming Methodology

ESM2017-41

객체지향프로그래밍 이론 및 실습

SKKU 시스템경영공학과

조영일

# Programming

컴퓨터 프로그래밍(Computer Programming) 혹은 간단히 프로그래밍(Programming)은

프로그래밍 언어(Programming Language)를 이용해

컴퓨터(Computer) 상에서 동작하는 특정한 문제를 해결하기 위한 프로그램(Program)을 작성하는 행위를 의미한다.

# Computer

컴퓨터(Computer)는 전자회로를 통하여 데이터를 입력 받아 미리 정의된 일련의 과정을 통하여 분석, 처리하여 그 결과를 출력하는 장치이다.

여기서 말하는 미리 정의된 일련의 과정은 컴퓨터를 사용하는 사용자에 의하여 특정 목적을 위해 변경될 수 있으며 이러한 특성을 프로그래밍 가능(Programmable)이라 한다.

즉, 컴퓨터는 단순히 특정 목적 만을 위해 고안된 것이 아니며 다양한 목적을 수행 할 수 있다.

# Program

컴퓨터 프로그램(Computer Program)은 컴퓨터에서 실행될 때 특정 작업을 수행하는 일련의 명령어들의 집합입니다.

컴퓨터 프로그램은 특정 문제를 해결하기 위해 고안된 특정 작업을 수행하기 위한 일련의 명령문의 집합체이며, 컴퓨터 프로그램을 통해 사용자는 컴퓨터를 다양한 문제를 해결하기 위해 다방면에 걸쳐 사용 할 수 있다.

e. g.)

Spreadsheet Program : 표 형식으로 데이터를 입력 받아 그 데이터를 조직하고 분석하기 위한 일련의 명령어들의 집합.

Game Program : 사용자와의 상호작용을 통해 데이터를 입력 받아 그 데이터에 따라 사용자에게 즐거움을 주기 위해 고안된 일련의 명령어들의 집합.

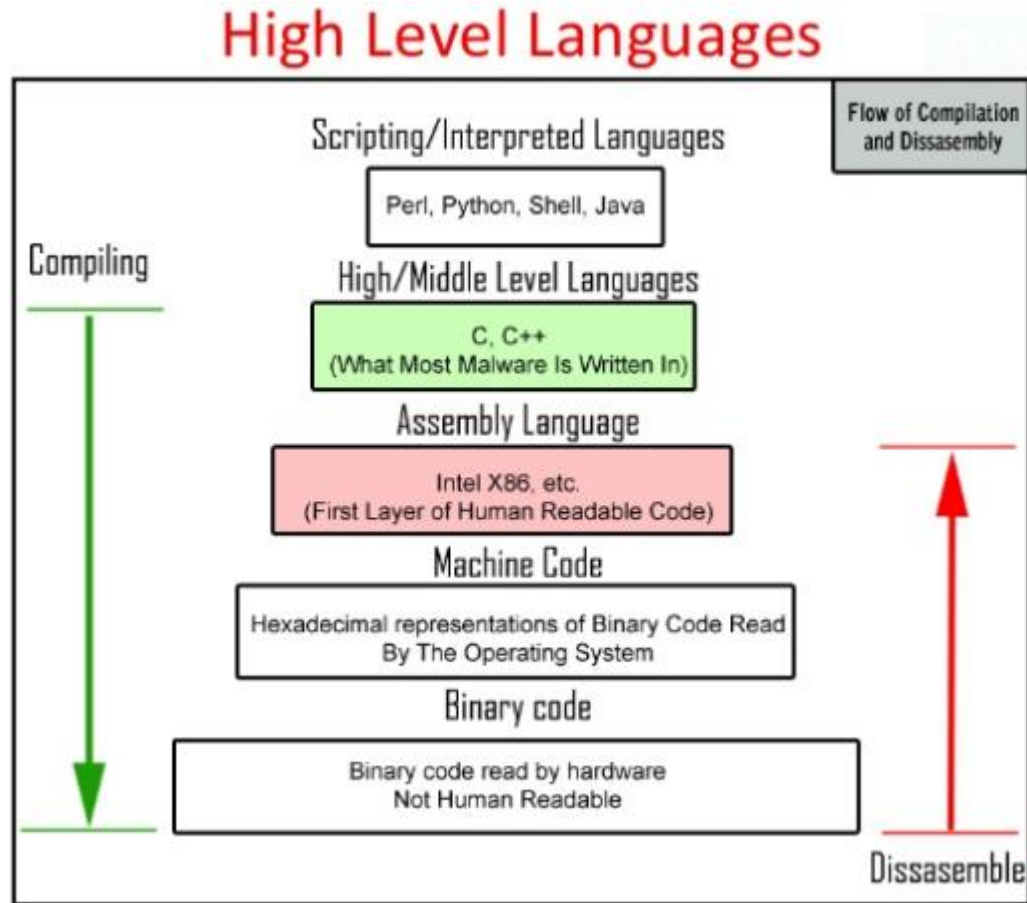
# Programming Language

프로그래밍 언어(Programming Language)는 프로그램을 작성하기 위한 언어이다.

컴퓨터는 전자 회로로 구성되어 있으며, 컴퓨터 프로그램은 이 전자회로에서 동작하기 위해 0과 1로 이루어진 이진수로 변환되어 실행된다.

0과 1로 이루어진 컴퓨터 프로그램은 일반적으로 사람이 이해하기 어렵고, 난해하기 때문에 현대적인 프로그래밍에서는 인간이 이해하기 쉬운(인간의 언어에 가까운) 프로그래밍 언어를 통해 프로그램을 작성하고 작성된 프로그램을 변환하여 실행하는 형태로 프로그래밍을 진행한다.

# Programming Language 분류



- 고수준 언어(High Level Language) : 사람의 언어에 가까우며 사람이 이해하기 쉬운 형태의 프로그래밍 언어(Python, Java, C++)

- 저수준 언어(Low Level Language) : 컴퓨터가 이해할 수 있는 언어에 가까우며 일반적으로 사람은 이해하기 어려운 형태(Assembly, Machine Code,/Binary Code)

# Program Development Process

1. 프로그램 설계 : 프로그램이 수행할 기능을 정의하고 논리적인 설계를 수행한다.
2. 프로그램 코드 작성(coding) : 1에서 수행한 논리적인 설계를 프로그래밍 언어로 옮긴다.(Code를 작성한다는 의미에서 Coding)
3. 기계어(Machine language)로 변환(Compile) : 작성된 프로그램을 실제 컴퓨터에서 실행하기 위해 컴퓨터가 해석 할 수 있는 기계어로 변환한다.
4. 프로그램 실행 및 오류 수정(Debugging) : 변환된 기계어 프로그램을 컴퓨터에서 실행하고 발생하는 오류를 1 및 2의 과정으로 돌아가 수정한다.

# Program Development Process

1. 프로그램 설계 : 프로그램이 수행할 기능을 정의하고 논리적인 설계를 수행한다.

- 작성할 프로그램이 가질 기본적인 기능에 대한 정의
- 해당 프로그램을 구현하기 위한 논리적인 구조 및 알고리즘에 대한 구현



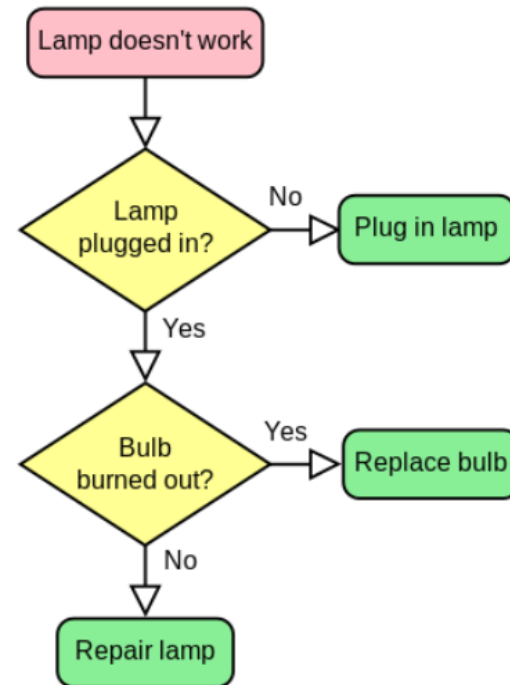
# Program Development Process

1. 프로그램 설계 : 프로그램이 수행할 기능을 정의하고 논리적인 설계를 수행한다.

- 흐름도(flowchart) : 프로그램의 처리 과정을 도식화

기호	명칭	설명
	단말	순서도의 시작과 끝을 나타냄.
	흐름선	순서도 기호 간의 연결 및 작업의 흐름을 표시함.
	준비	작업 단계 시작 전 해야 할 작업을 명시함.
	처리	처리해야 할 작업을 명시함.
	입출력	데이터의 입출력 시 사용함.
	의사 결정	비교 및 판단에 의한 논리적 분기를 나타냄.
	표시	화면으로 결과를 출력함.

<http://tcpschool.com/codingmath/flowchart>



<https://en.wikipedia.org/wiki/Flowchart>

# Program Development Process

1. 프로그램 설계 : 프로그램이 수행할 기능을 정의하고 논리적인 설계를 수행한다.

- 의사코드(疑似: 비교할 의, 비슷할 사 | Pseudo: 가짜의- + code) : 프로그램의 설계를 위하여 실제 프로그래밍 언어의 문법을 명확하게 따르지 않는 유사한 형태의 코드나 일상어를 통해 작성한 코드, 문법적인 오류를 신경 쓰지 않고 작성한다.

```
for  $j=1$  to  $NP_2$  (for all butterflies in Subpopulation 2) do  
    Calculate the walk step  $dx$  by Eq. (7);  
    Calculate the weighting factor by Eq. (8);  
    for  $k=1$  to  $D$  (all the elements in  $j$ th butterfly) do  
        Generate  $x_{j1,k}^{t+1}$  by Eq. (4) and Eq. (5).  
    end for  $k$   
    Generate  $x_{j2}^{t+1}$  by performing SAR operator as Eq. (10).  
    Generate  $x_{j,new}^{t+1}$  according to greedy strategy as Eq. (12).  
end for  $j$ 
```

# Program Development Process

2. 프로그램 코드 작성(coding) : 1에서 수행한 논리적인 설계를 프로그래밍 언어로 옮긴다.(Code를 작성한다는 의미에서 Coding)
  - 프로그램에 대한 논리적 설계가 완료되었다면, 프로그램의 목적/실행환경 등에 따라 적합한 프로그래밍 언어를 선택 한 이후 논리적 설계를 프로그래밍 언어의 문법(Syntax)에 맞추어 작성하는 과정을 진행한다.
  - 프로그램의 논리적 설계가 올바르게 이루어졌다면, 그것을 문법대로 코드화 하는 과정은 상대적으로 난이도가 쉽다.
  - 코드는 메모장/vi 등 일반적인 텍스트 에디터를 통해서 작성 할 수 있지만, 현대적인 프로그래밍 환경에서는 프로그램 작성의 효율성을 위하여 고안된 전용 코드 작성 프로그램인 통합개발환경(IDE)을 통해 코드를 작성하는 경우가 대부분임.

# Program Development Process

2. 프로그램 코드 작성(coding) : 1에서 수행한 논리적인 설계를 프로그래밍 언어로 옮긴다.(Code를 작성한다는 의미에서 Coding)
  - 통합개발환경(IDE, Integrated Development Environment)는 코드 편집, 디버깅, 컴파일, 배포 등 프로그램 개발과 관련된 모든 작업을 하나의 프로그램 안에서 처리 할 수 있도록 고안된 소프트웨어를 의미함
    - 코드 편집기(Code Editor) : 프로그래밍 코드를 작성하고 수정하기 위한 도구, 프로그래밍 언어에 따라서 작성된 코드의 문법 오류를 검사하고 코드의 가독성을 높이기 위한 강조 표현(Syntax Highlighting) 이 지원됨
    - 컴파일러(Compiler) : 작성된 프로그래밍 코드를 컴퓨터에서 실행 가능한 형태로 변환하는 도구
    - 디버거(Debugger) : 작성된 프로그램의 오류(Bug)를 수정하는 작업(Debugging)을 도와주는 도구
    - 특정 프로그래밍 언어에 특화된 IDE가 있으며, 요즘에는 하나의 IDE가 여러가지 프로그래밍 언어를 지원하는 경우가 많음
    - 유명 IDE 예시 : Visual Studio, Eclipse, IntelliJ IDEA

# Program Development Process

- 3. 기계어(Machine language)로 변환(Compile) : 작성된 프로그램을 실제 컴퓨터에서 실행하기 위해 컴퓨터가 해석 할 수 있는 기계어로 변환한다.
- 기계어(Machine Code) : 저수준 언어이며 컴퓨터가 해석 할 수 있는 유일한 언어, 0과 1로만 표기되기 때문에 사람은 이해 하기 어렵다.
- Compile 과정은 사람이 해석 할 수 있는 고수준 프로그래밍 언어로 작성된 프로그램을 기계어로 변환하는 과정을 의미하며, 어떠한 프로그래밍 언어로 작성되었더라도 실제 실행을 위해선 Compile 과정이 수반되어야 한다.
- 프로그래밍 언어에 따라 Compiler 혹은 Interpreter 에 의해 수행된다.
  - Compiler : 작성된 프로그램 코드 전체를 한번에 Machine Code로 변환하는 방식(C, C++ 등)
  - Interpreter : 작성된 프로그램 코드를 한번에 한줄씩 실행하며 Machine Code로 변환하는 방식(Python, JavaScript, Ruby 등)

# Program Development Process

4. 프로그램 실행 및 오류 수정(Debugging) : 변환된 기계어 프로그램을 컴퓨터에서 실행하고 발생하는 오류를 1 및 2의 과정으로 돌아가 수정한다.
- 실제 프로그램을 실행하고 해당 프로그램의 동작이 최초의 설계와 일치하는지를 검증한다.
  - 또한, 다양한 테스트를 통해 최초의 설계 상 결함(논리적인 오류) 혹은 프로그램 작성과정의 실수 인해 프로그램이 오류가 생기지 않는지를 검증한다.
  - 검증 결과에 따라 1, 2의 과정으로 돌아가 프로그램의 설계 혹은 작성된 코드를 수정 후 다시 검증과정을 시작한다.