

수업이 지향하는 6가지 목표는 다음과 같다:

- 1) 자료구조를 활용한 지식의 표현
- 2) 반복과 재귀에 대한 이해
- 3) 데이터 타입과 과정의 추상화
- 4) 클래스와 메소드를 활용한 시스템의 구성과 모듈화
- 5) 검색이나 분류같은 다양한 형태의 알고리즘
- 6) 알고리즘의 복잡도 분석 (프로그램 성능 비교)

컴퓨터는 보통 계산을 수행하고 결과를 저장한다.

이러한 계산은 사칙연산같은 간단한 연산일수도, 프로그래머가 임의로 정의한 계산일 수도 있다.

컴퓨터는 사용자가 알려주는 것만을 수행한다.

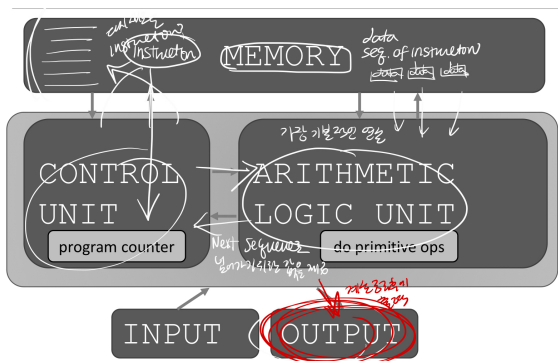
지식에선 선언적 지식과 절차적 지식이 있다.

선언적 지식은 참과 거짓을 판명하는 하나의 지식이고, 절차적 지식은 이 선언적 지식을 활용하는 방법이다.

절차는 어떠한 흐름의 각 방법들을 포함한다.

이에 더해 “언제 멈춰야 하는지” 를 꼭 포함해야 한다.

기본적인 컴퓨터의 연산장치는 크게 1) 메모리, 2) 컨트롤 유닛, 3) 알고리즘 유닛으로 구성된다.



알고리즘 로직 유닛의 Primitive Ops는

- 1) arithmetic and logic
- 2) simple tests
- 3) moving data의 3단계를 거쳐 나간다.

프로그래밍의 6가지 기본 동작:

- 1) Moving left
- 2) Moving right
- 3) Read
- 4) Write
- 5) Scan
- 6) Doing nothing

+ 여기에 프로그래머 임의 primitive 제작!

이론적으로 한 언어로 짜인 프로그램은 다른 프로그램으로 실행가능하도록 제작하는 것이 가능

Error의 3가지 유형 :

- 1) Syntactic Errors
- 2) static semantic errors
- 3) different meaning errors

프로그램은 정의된 뜻과 실행될 명령의 집합체이다.

명령은 인터프리터로 하여금 행동을 지시한다.

프로그램은 데이터 객체를 갖고 modify하며,

이 데이터 객체는 각자의 “타입” 을 갖고 있다.

이 타입은 연산과 행동에 중요한 영향을 미친다.

데이터 객체는 Scalar와 Non-Scalar로 나눌 수 있다.

Scalar는 다음과 5가지 형태를 포함한다:

- 1) int - 정수를 표현 (ex. 5)
- 2) float - 실수를 표현 (ex. 3, 27)
- 3) bool - 참/거짓을 표현 (True / False)
- 4) NoneType - 아무것도 아님을 표현 (None)

Scalar 사이에는 데이터 형태의 변환이 가능하다.

Expression는 Scalar 사이의 상호작용을 담당한다.

Variable = value를 통해서 변수를 정의한다.

만약 계산 과정에서 변수를 재정의하게 된다면,

기존의 미사용 value는 여전히 메모리에 남는다.

단, 기존에 연결된 변수가 끊어진 것 뿐이다.