

# 프로그래밍 언어론

동아대학교 소프트웨어대학 조장우



# 담당교수

---

- 조장우, 동아대학교 소프트웨어대학 교수
- 연구실: S06 619호
- e-mail: [jwjo@dau.ac.kr](mailto:jwjo@dau.ac.kr)
- phone: 200-7780
- 강의자료: 가상대학



# 성적평가

---

- 중간시험: 35%
- 기말시험: 35%
- 과제: 20%
- 출석: 10%



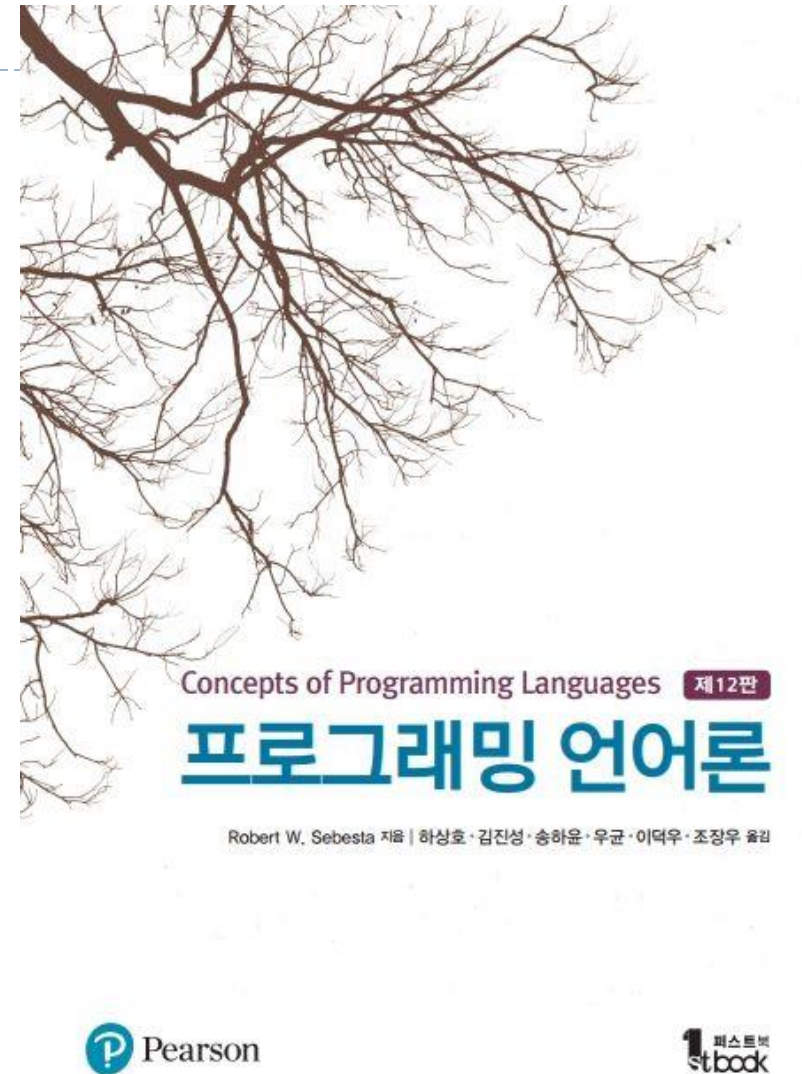
# 교재

- 교재

- 프로그래밍언어론 12판, 하상호, 김진성, 송하윤, 우균, 이덕우, 조장우 번역, 퍼스트북, 2024년

- 참고문헌

- Kenneth C. Louden, Programming languages : Principles and Practice, 2nd Edition, Thomson
- 프로그래밍 언어론: 원리와 실제, 인피니티북스, 창병모, 2021.



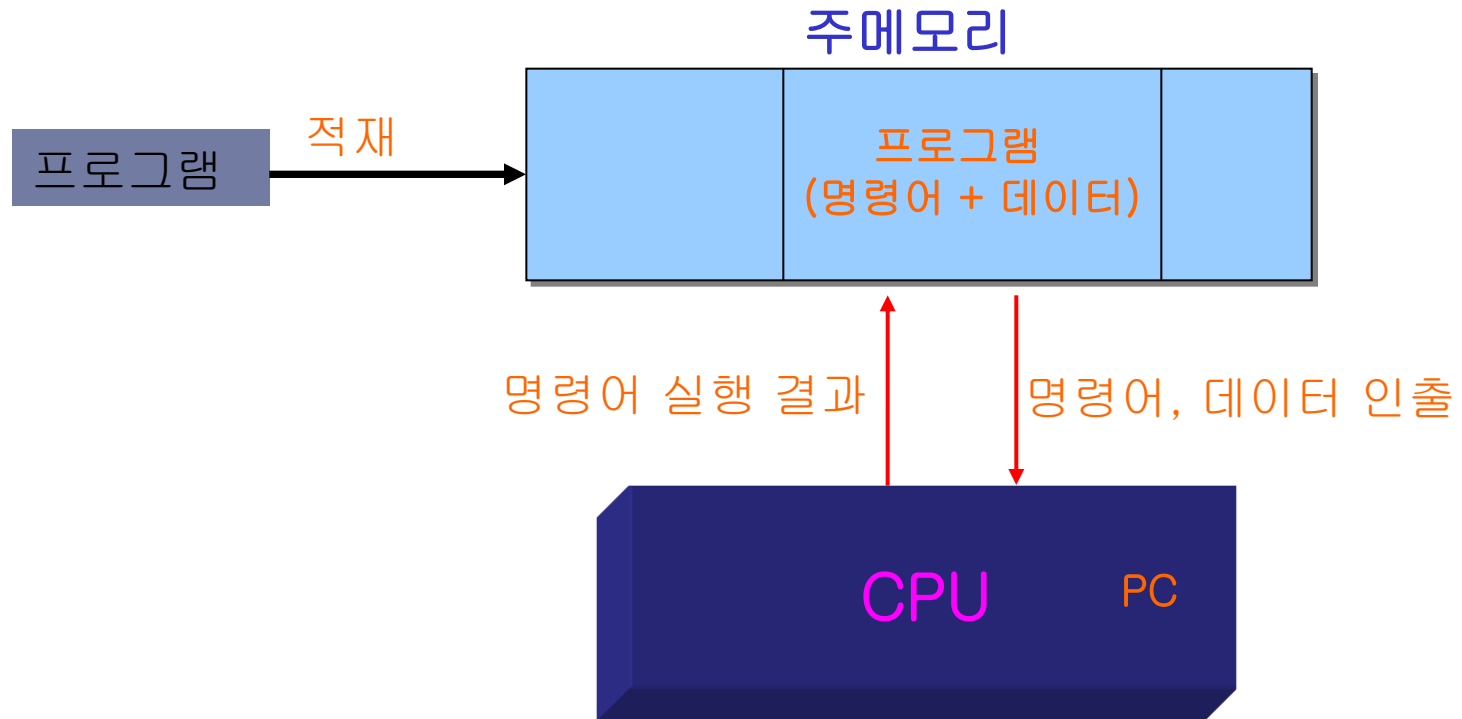
# 프로그래밍 언어 발전 과정

---

- 역사적 발전 과정
  - 최초의 컴퓨터(ENIAC, EDVAC, UNIVAC)가 만들어지면서
  - 프로그래밍 언어가 개발되기 시작했을 것이다.
- 두 가지 질문?
  - 그때 컴퓨터는 어떤 컴퓨터였을까요?
  - 어떤 프로그래밍 언어가 개발되었을까요?
- 컴퓨터
  - Von Neuman model computer
  - 폰 노이만 모델 컴퓨터
- 초창기 프로그램
  - 컴퓨터에 명령하는 기계어 명령어들의 집합

# Von Neuman 모델 컴퓨터

---



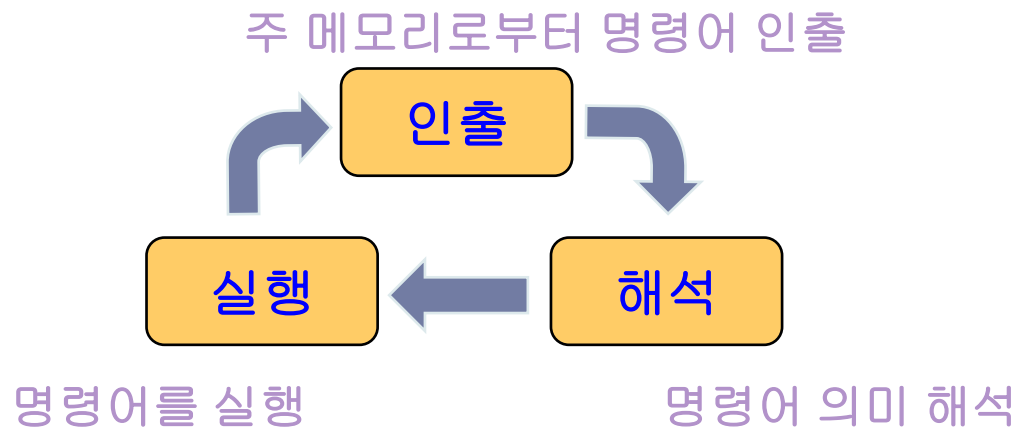
# Von Neuman 모델 컴퓨터

---

- 프로그램 내장 방식 컴퓨터
  - stored program computer
  - 메모리에 프로그램(명령어와 데이터) 저장
- 메모리에 저장된 명령어 순차 실행
  - a single CPU sequentially execute instructions in memory
  - PC  
다음 실행할 명령어를 가리킨다.
- 명령어
  - 메모리에 저장된 값을 조작 혹은 연산
  - instructions operate on values stored in memory

# Von Neuman 컴퓨터 프로그램 실행

- CPU의 인출-해석-실행(fetch-decode-execute) 주기 반복
  - CPU는 주 메모리 내에 저장되어 있는 프로그램의 명령어를
  - 한 번에 하나씩 읽어 들여 해석하고 실행한다.





# 컴퓨터의 예: 기계어 코딩의 실습

---

- 세자리 수(3 digits) 컴퓨터(십진수)
- 메모리
  - 1,000개의 공간, 주소는 0 ~ 999
  - 한 주소에 3 자리 숫자까지 저장
- 레지스터 10개
  - R0, R1, R2, ..., R9
  - 한 레지스터에 3 자리 숫자까지 저장
  - 모든 레지스터는 000으로 초기화
- 명령어 10개
  - 한 명령어는 3자리 숫자로 구성

# 명령어 집합 (Instructions Set)

---

- 종료
  - 100 : halt
- 레지스터와 상수 연산
  - $2dn$  : set register  $d$  to  $n$  299
  - $3dn$  : add  $n$  to register  $d$  399
  - $4dn$  : multiply register  $d$  by  $n$  492
- 레지스터와 레지스터 연산
  - $5ds$  : set register  $d$  to the value of register  $s$  521
  - $6ds$  : add the value of register  $s$  to register  $d$  621
  - $7ds$  : multiply register  $d$  by the value of register  $s$  721

# 명령어 집합 (Instructions Set)

---

- 적재(load)
  - 메모리 값을 레지스터에 적재(load)
  - 8da : set register  $d$  to the value in RAM whose address is in register  $a$
  - *892*
- 저장(store)
  - 레지스터 값을 메모리에 저장(store)
  - 9sa : set the value in RAM whose address is in register  $a$  to that of register  $s$
  - *992*
- Goto
  - 0ds : goto the location in register  $d$  unless register  $s$  contains 0
  - *082*

# 프로그램

---

- 프로그램

- 일련의 연속된 명령어들 + 데이터
- 메모리의 0번지부터 적재된다.
- 명시되지 않는 부분의 메모리는 000으로 초기화

# 예제 프로그램

- $R0 \leftarrow -1$  (R0에 1 저장)  $\begin{matrix} 0: 201 \\ 1: 100 \end{matrix}$
  - $R1 \leftarrow -10$  (R1에 10 저장)  $\begin{matrix} 0: 215 \\ 1: 315 \\ 2: 100 \end{matrix}$
  - $R2 \leftarrow -3; R2 \leftarrow -R2 * R1$   $\begin{matrix} 0: 215 \\ 1: 315 \\ 2: 223 \\ 3: 721 \\ 4: 100 \end{matrix}$
  - $R2 \leftarrow R2 - 1$ 
    - 빼기 명령어가 없는데 어떻게?
    - 오버플로우 이용:  $999 + 20 = 1019$
- |     |     |
|-----|-----|
| 0:  | 299 |
| 1:  | 492 |
| 2:  | 495 |
| 3:  | 399 |
| 4:  | 492 |
| 5:  | 495 |
| 6:  | 399 |
| 7:  | 215 |
| 8:  | 315 |
| 9:  | 223 |
| 10: | 721 |
| 11: | 629 |
| 12: | 100 |

# 예제 프로그램

---

0: 299

1: 492

2: 495

3: 399

4: 492

5: 495

6: 399

7: 283

8: 279

9: 689

10: 078

11: 100

12: 000

13: 000

0: 299

1: 492

2: 495

3: 399

4: 492

5: 495

6: 399

7: 280

8: 279

9: 687

10: 679

11: 269

12: 067

13: 100

0: 299

1: 492

2: 495

3: 399

4: 492

5: 495

6: 399

7: 281

8: 276

9: 787

10: 679

11: 269

12: 067

13: 100

R8의 값은?

# 세자리수 컴퓨터 프로그램

---

- 100번지의 값에 1 더하기
- 100번지와 101번지의 값을 더해서 102번지에 저장
- 101번지에서 150번지의 값을 더해서 200번지에 저장
- 1에서 50까지의 합을 200번지에 저장

# 명령형 언어(Imperative language)

---

- 명령형 언어의 발전

Imperative programming languages began by **imitating** and **abstracting** the operations of von Neuman model computer

- 예

- **Fortran, Basic, C**

- 특징: 폰 노이만 모델 컴퓨터의 특징을 많이 갖고 있겠지요.

- 순차적 명령어 실행
- 메모리 위치를 나타내는 변수 사용
- 배정문을 사용한 변수 값 변경

- 그러나, 사람의 필요보다는 컴퓨터 모델을 기반으로 한 언어