

# AI활용프로그래밍

## Week 15. 기말시험

범위 요약 + 샘플 문제 + 준비 전략

## 학습 목표

기말시험의 범위/형식/규칙을 이해한다.  
Week 9~14 핵심 개념을 요약하고 실전 문제에 적용한다.  
샘플 문제로 시간 관리 전략과 디버깅 전략을 점검한다.  
시험 전에는 AI로 연습하되, 시험 중 규정을 준수한다.

## 오늘의 구성

---

기말시험 안내(형식/범위/준비물)  
핵심 개념 요약(Week 9~14)  
샘플 문제(실기/이론/디버깅)  
자주 하는 실수 & 시간 관리  
Q&A

## 기말시험 개요

- 평가 방식/시간/제출 방식은 담당 교원 공지를 따른다.
- 문제 유형: 데이터 처리 + 시각화 + 파일 I/O + (선택)쓰레드

Tip: 프로젝트 코드를 복습하면 점수가 오른다(실전 기능이 그대로 등장).

## 시험 규칙(예시)

- 허용/금지 사항은 담당 교원 공지를 따른다.
- 일반적으로 시험 중 AI 도구 사용은 제한될 수 있다.
- 부정행위는 학칙에 따른다.

시험 전에는 AI로 연습하되, 코드는 반드시 이해하고 검증!

## 시험 범위(Week 9~14)

---

Week 9 : 파일 입출력(text/csv/json)

Week 10: NumPy, Pandas

Week 11: Matplotlib 그래프

Week 12: 필터링·정렬·groupby

Week 13: 스레드, 미니게임

Week 14: 개인 과제(설계/문서화/AI 활용 기록)

## 준비물 & 환경 점검

- Python, pandas, matplotlib 설치 확인
- 파일 경로(현재 작업 폴더) 확인
- CSV/JSON 샘플을 읽고 저장해보기
- 그래프 저장(savefig) 경로 확인

## 핵심 요약: 파일 입출력(Week 9)

- with open(...): 안전한 I/O
- try/except로 파일 없음/깨짐 처리
- csv/json 모듈로 표/구조 데이터 처리
- encoding='utf-8', newline="" 주의



## 핵심 요약: Pandas(Week 10)

- read\_csv / DataFrame 기본 조작
- 컬럼 선택(df['col'], df[['a','b']])
- 결측치 처리(isna, dropna, fillna)
- 간단한 통계(mean, value\_counts)

## 핵심 요약: Matplotlib(Week 11)

- line/bar/hist/scatter 그래프 선택
- title/xlabel/ylabel/legend
- savefig로 파일 저장
- 그래프 해석: 축/단위/추세/이상치

## 핵심 요약: 필터링·정렬(Week 12)

- `df[cond]` 필터링, (`&`, `|`) 사용 시 괄호
- `sort_values(ascending=False)`
- `groupby` + `agg`로 요약 테이블
- Top-N + `value_counts` 패턴

## 핵심 요약: 스레드(Week 13)

- Thread(target=...), start/join
- 종료 플래그(bool/Event)
- 공유 데이터는 Lock/Queue로 보호
- 동시성은 타이머/입력 처리에 유용

## 핵심 요약: 프로젝트/발표(Week 14)

- 기능뿐 아니라 구조/문서/테스트가 중요
- README: 실행 방법/예외/제약
- AI 활용 기록: 프롬프트→검증→수정의 흔적

## 자주 나오는 실수 TOP 10

- 1) 파일 경로 오류(작업 폴더 착각)
- 2) encoding/newline 처리 누락
- 3) pandas 조건 괄호 누락(cond1 & cond2)
- 4) 컬럼명 오타(KeyError)
- 5) 결측치 처리 미흡
- 6) 그래프 라벨/단위 누락
- 7) 정렬 후 head/top-N 실수
- 8) 쓰레드 종료 조건 없음(프로그램 안 끝남)
- 9) 출력 형식(공백/줄바꿈) 오류
- 10) 테스트 케이스 부족

## AI로 공부하는 방법(시험 전)

- '시험 스타일 문제 5개 만들어줘(해설 포함)'
- '내 풀이를 채점하고 개선점 제안해줘'
- '옛지 케이스를 더 만들어줘'

주의: 시험 중 규정은 반드시 준수!

## 샘플 이론 문제

Q) pandas에서 두 조건을 결합할 때 올바른 코드는?

A) `df[(a>0) and (b>0)]`

B) `df[(a>0) & (b>0)]`

C) `df[a>0 & b>0]`

정답: B (괄호 + & 사용)



## 샘플 실기 1: CSV → 요약

```
1 import pandas as pd
2
3 df = pd.read_csv("scores.csv")
4 # TODO: score 평균, 최고점 학생(name, score) 출력
```

### 핵심 포인트

df['score'].mean(), idxmax() 활용 가능  
빈 데이터/결측치 있을 때 처리 규칙  
정하기  
출력 형식을 문제 요구에 맞추기

## 샘플 실기 2: 필터 + Top-N

```
1 # TODO:  
2 # - score >= 80 인 학생만 필터링  
3 # - score 내림차순으로 상위 3명 출력
```

### 핵심 포인트

cond 변수로 분리  
sort\_values + head(3)  
동점 처리(선택): rank로 동일 점수  
포함

## 샘플 실기 3: 그래프 저장

```
1 import matplotlib.pyplot as plt
2
3 x = [1,2,3,4]
4 y = [10,20,15,25]
5 # TODO: 선 그래프 + 라벨/제목 추가 후 png로 저장
```

### 핵심 포인트

xlabel/ylabel/title/legend  
savefig('out.png', dpi=150)  
파일이 저장된 위치 확인

## 샘플 실기 4: JSON 저장

```
1 import json
2
3 result = {"avg": 82.5, "top": {"name": "kim",
4 # TODO: result를 report.json으로 저장(한글 포함 가
   "score": 95}}
```

### 핵심 포인트

json.dump(..., ensure\_ascii=False,  
indent=2)  
파일 경로/권한 확인  
저장 후 다시 load해서 검증

## 샘플 디버깅: SettingWithCopyWarning

```
1 # 잘못된 예(체인 인덱싱)
2 high = df[df["score"] >= 80]
3 high["flag"] = True
```

### 핵심 포인트

체인 인덱싱은  
SettingWithCopyWarning 유발  
해결: high =  
df.loc[df['score']>=80].copy()  
시험에서는 '정확히 동작'하는 코드  
가 중요

## 시간 관리 전략

- 1) 요구사항을 체크박스로 분해
- 2) 데이터 읽기 → 중간 출력(head)로 확인
- 3) 핵심 로직 구현 후, 형식/예외 처리 보완
- 4) 최소 2개 테스트(정상/엣지) 실행

## 실전 체크리스트

- 경로/파일명 확인(대소문자 포함)
- 컬럼명 확인(df.columns)
- 조건 괄호 확인
- 그래프 라벨/제목/저장 확인
- 저장한 파일을 다시 열어 확인(load/read)

## Q&A

---

궁금한 점을 질문하세요.

- 범위/형식
- 실습 환경
- 채점 기준
- 준비 전략



# 마무리

기말은 '데이터 처리 + 분석 + 시각화 + 파일 저장'이 핵심이다.  
프로젝트를 복습하고, 직접 손으로 코딩하며 경계값 테스트!

Good luck 🍀