Efi Arazi School of Computer Science
Introduction to Computer Science

Lecture 13-2

Summary

public static void main(String[] args) {...
System.al.println ("Hello, Java)
JAVA

Reminder: Course objectives

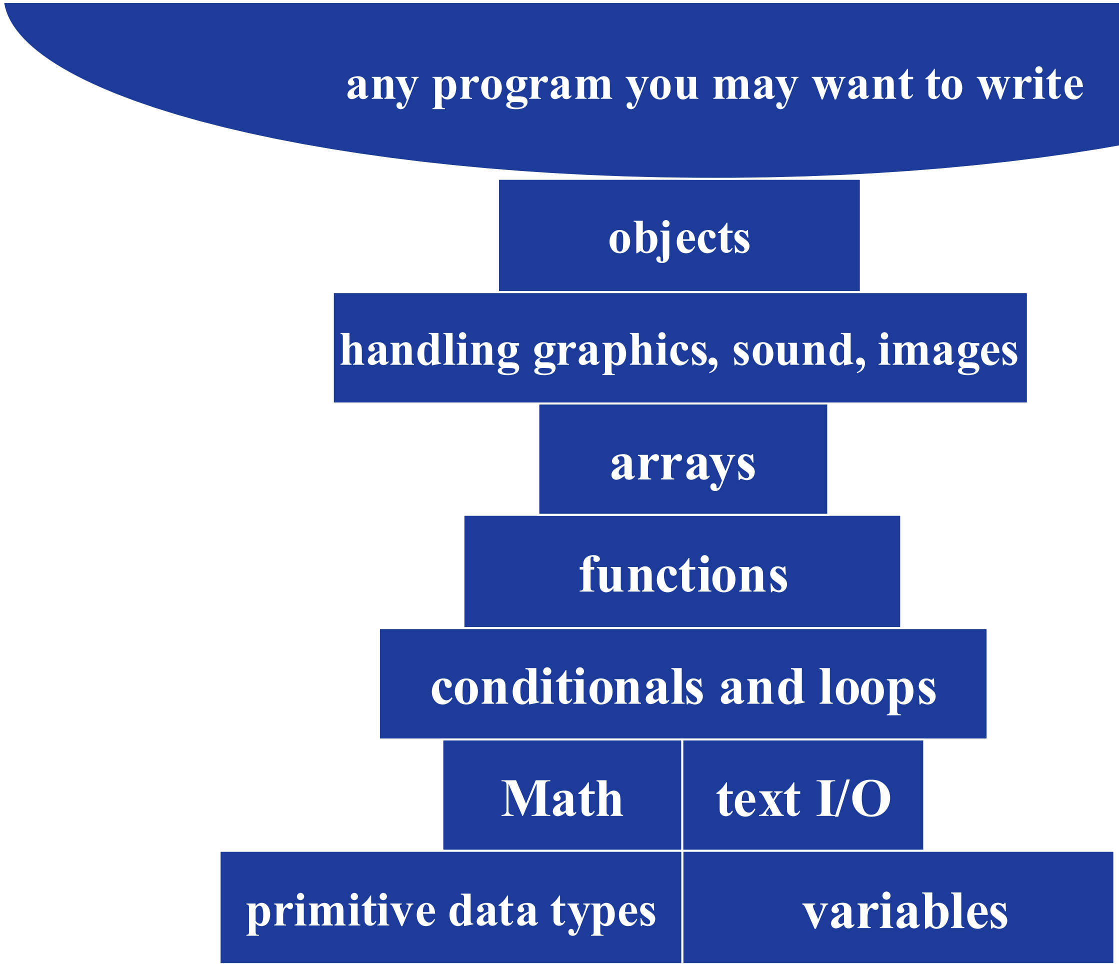
The course will give you …

- Basic exposure to Computer Science
- Basic programming skills

In addition, you will …

- Sharpen your analytic skills
- Appreciate clarity and elegance
- Develop a taste for beauty in science and engineering
- Learn how to learn and develop.

The big picture



any program you may want to write

objects

handling graphics, sound, images

arrays

functions

conditionals and loops

Math

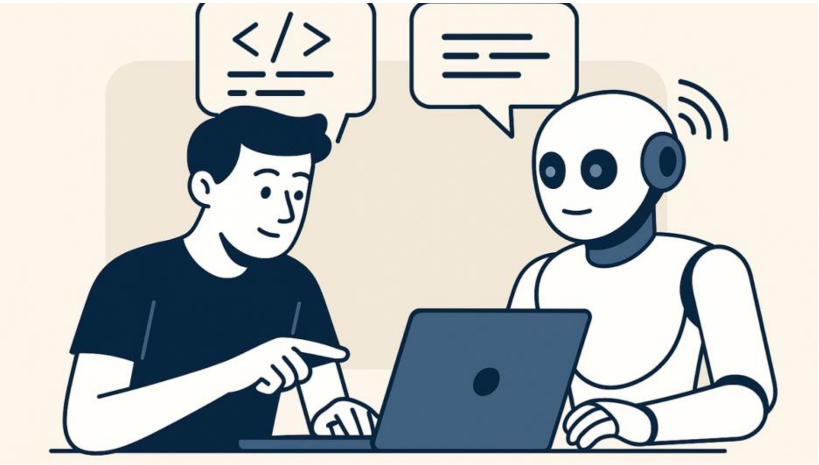
text I/O

primitive data types

variables

How people code today?

- Coding is no longer a solo activity
- Developers work with AI, not instead of thinking
- **Understanding *how* to use them well is now a core CS skill**



From “Writing Code” to “Designing Solutions”

Then **Think → write code → debug → repeat**

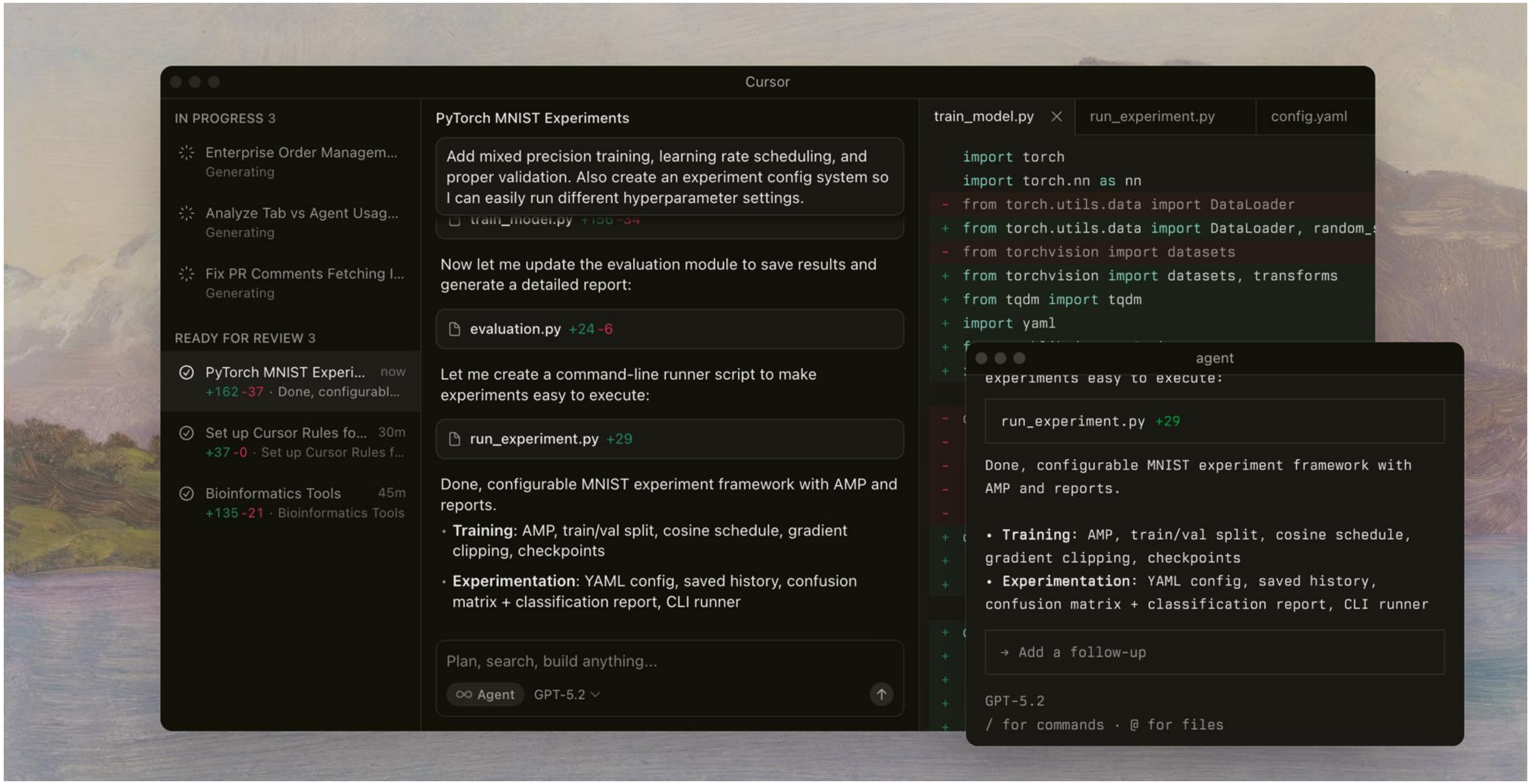
Now **Think → describe intent → evaluate AI output → refine → test**

Key shift

- From syntax-first to reasoning-first
- From “How do I write this loop?” to “What should the system do?”

AI Driven IDEs

Cursor



IN PROGRESS 3

- Enterprise Order Managem... Generating
- Analyze Tab vs Agent Usag... Generating
- Fix PR Comments Fetching I... Generating

READY FOR REVIEW 3

- PyTorch MNIST Experi... now +162 -37 · Done, configurabl...
- Set up Cursor Rules fo... 30m +37 -0 · Set up Cursor Rules f...
- Bioinformatics Tools 45m +135 -21 · Bioinformatics Tools

PyTorch MNIST Experiments

Add mixed precision training, learning rate scheduling, and proper validation. Also create an experiment config system so I can easily run different hyperparameter settings.

train_model.py +156 -34

Now let me update the evaluation module to save results and generate a detailed report:

evaluation.py +24 -6

Let me create a command-line runner script to make experiments easy to execute:

run_experiment.py +29

Done, configurable MNIST experiment framework with AMP and reports.

- **Training**: AMP, train/val split, cosine schedule, gradient clipping, checkpoints
- **Experimentation**: YAML config, saved history, confusion matrix + classification report, CLI runner

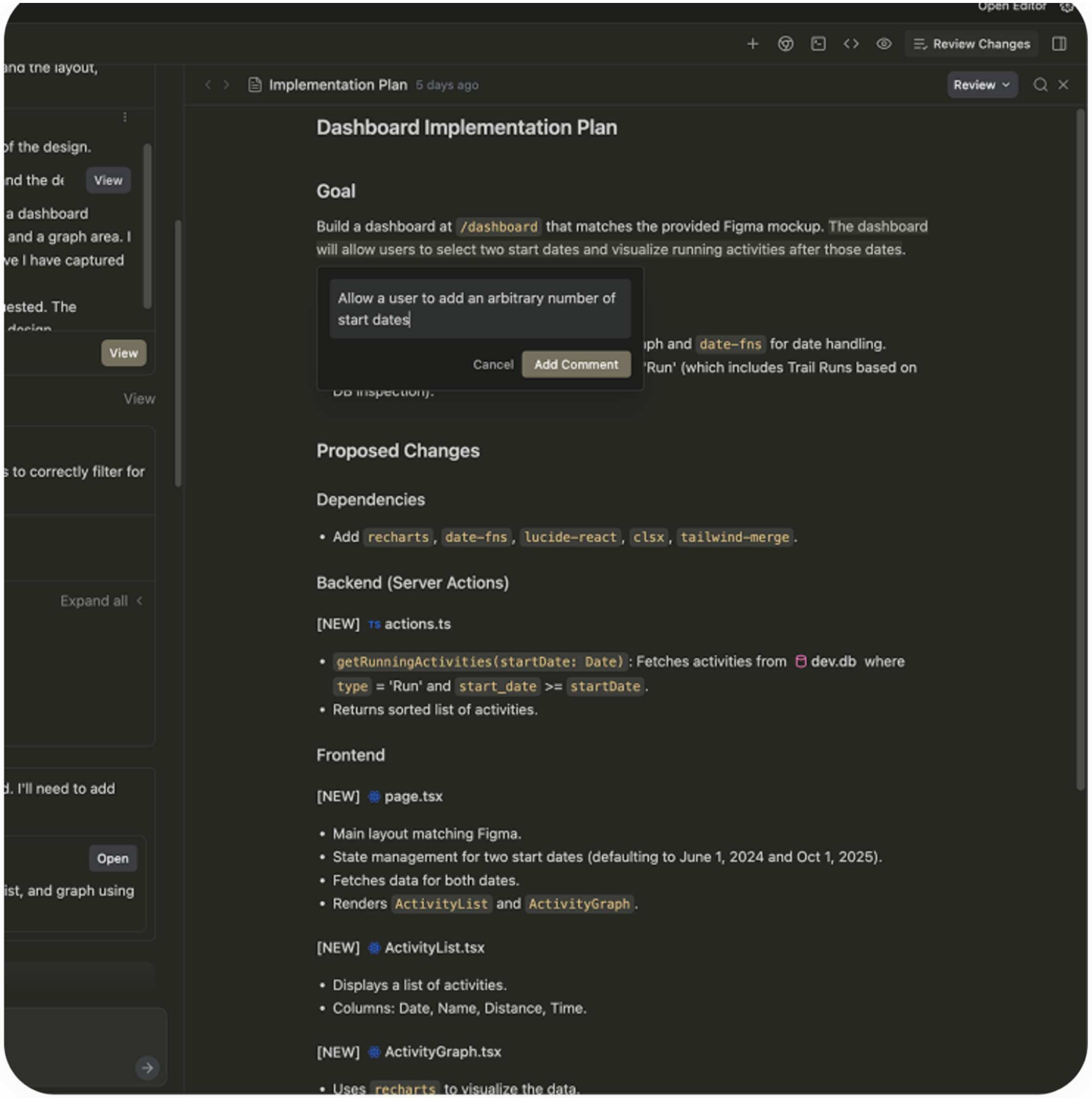
Plan, search, build anything...

```
  import torch
  import torch.nn as nn
- from torch.utils.data import DataLoader
+ from torch.utils.data import DataLoader, random_s
- from torchvision import datasets
+ from torchvision import datasets, transforms
+ from tqdm import tqdm
+ import yaml
```

```
experiments easy to execute:

run_experiment.py +29

Done, configurable MNIST experiment framework with
AMP and reports.

• Training: AMP, train/val split, cosine schedule,
gradient clipping, checkpoints
• Experimentation: YAML config, saved history,
confusion matrix + classification report, CLI runner

→ Add a follow-up

GPT-5.2
/ for commands · @ for files
```

AI Driven IDEs

Anti Gravity



...and the layout,

...of the design.

...nd the de

...a dashboard ...and a graph area. I ...ve I have captured

...ested. The ...design

...s to correctly filter for

...d. I'll need to add

...ist, and graph using

Implementation Plan 5 days ago

Dashboard Implementation Plan

Goal

Build a dashboard at `/dashboard` that matches the provided Figma mockup. The dashboard will allow users to select two start dates and visualize running activities after those dates.

Allow a user to add an arbitrary number of start dates

Cancel | Add Comment

...ph and `date-fns` for date handling.

...'Run' (which includes Trail Runs based on DB inspection).

Proposed Changes

Dependencies

- Add `recharts`, `date-fns`, `lucide-react`, `clsx`, `tailwind-merge`.

Backend (Server Actions)

[NEW] actions.ts

- `getRunningActivities(startDate: Date)`: Fetches activities from dev.db where `type` = 'Run' and `start_date` >= `startDate`.
- Returns sorted list of activities.

Frontend

[NEW] page.tsx

- Main layout matching Figma.
- State management for two start dates (defaulting to June 1, 2024 and Oct 1, 2025).
- Fetches data for both dates.
- Renders `ActivityList` and `ActivityGraph`.

[NEW] ActivityList.tsx

- Displays a list of activities.
- Columns: Date, Name, Distance, Time.

[NEW] ActivityGraph.tsx

- Uses `recharts` to visualize the data.

What Skills Matter More Than Ever?

Still essential

- Algorithms & data structures
- Debugging
- Reading code
- Testing

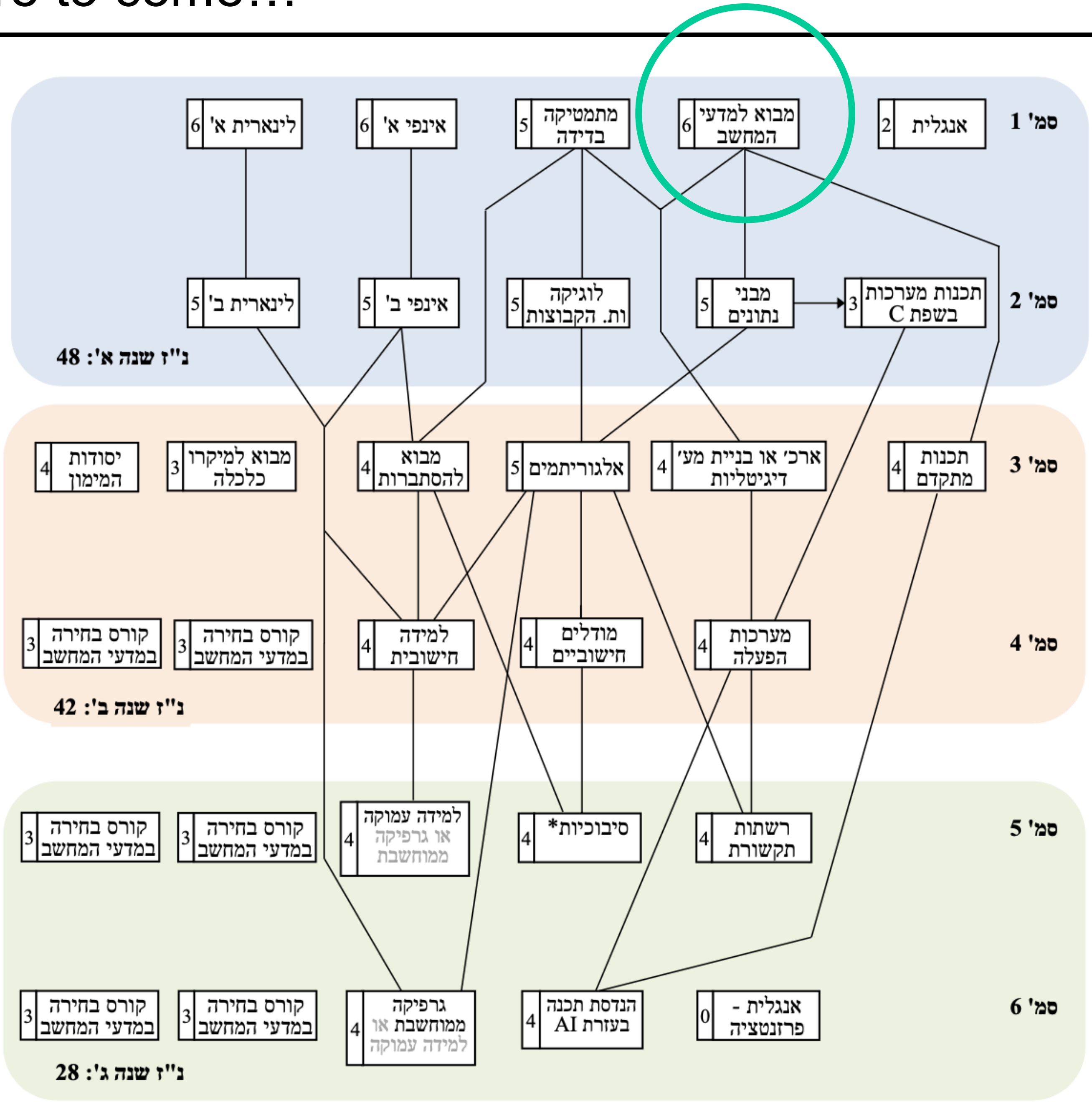
More important than before

- Problem decomposition
- Precise communication
- Asking good questions
- Understanding failure modes

Less central (but not gone)

- Memorizing syntax
- Writing everything from scratch

Much more to come…



סמ' 1
- 2 אנגלית
- 6 מבוא למדעי המחשב
- 5 מתמטיקה בדידה
- 6 אינפי א'
- 6 לינארית א'

סמ' 2
- 3 תכנות מערכות בשפת C
- 5 מבני נתונים
- 5 לוגיקה ות. הקבוצות
- 5 אינפי ב'
- 5 לינארית ב'

נ"ז שנה א': 48

סמ' 3
- 4 תכנות מתקדם
- 4 ארכ' או בניית מע' דיגיטליות
- 5 אלגוריתמים
- 4 מבוא להסתברות
- 3 מבוא למיקרו כלכלה
- 4 יסודות המימון

סמ' 4
- 4 מערכות הפעלה
- 4 מודלים חישוביים
- 4 למידה חישובית
- 3 קורס בחירה במדעי המחשב
- 3 קורס בחירה במדעי המחשב

נ"ז שנה ב': 42

סמ' 5
- 4 רשתות תקשורת
- 4 סיבוכיות*
- 4 למידה עמוקה או גרפיקה ממוחשבת
- 3 קורס בחירה במדעי המחשב
- 3 קורס בחירה במדעי המחשב

סמ' 6
- 0 אנגלית - פרזנטציה
- 4 הנדסת תכנה בעזרת AI
- 4 גרפיקה ממוחשבת או למידה עמוקה
- 3 קורס בחירה במדעי המחשב
- 3 קורס בחירה במדעי המחשב

נ"ז שנה ג': 28

Directly related courses

→ Data Structures (מבני נתונים) –LinkedLists, Sets, Queues, Maps, Binary Trees… much more to learn and analyze.

→ Digital Systems (מערכות דיגיטליות) – We learned Vic. Much more to learn how a real system is designed and implemented.

→ Advanced Programming (תכנות מתקדם) - We learned how to program any program we like. But there are more topics to cover, including design patterns and advanced OOP techniques.

→ Software engineering with AI (הנדסת תוכנה בעזרת בינה מלאכותית) – You will learn how to use AI for designing and building systems.

What’s left

- Assignment 9
- Final Exam

Exam

- 3 hours.
- Same format of the midterm (main file + help pages).
- Answers are provided only on the main file.
- Topic coverage: Everything we learned in class, recitations and homework assignments (excluding Binary trees). Data structures, efficiency, Vic, recursion, are all included among all other things. We will not ask you questions about graphics, animation and sound processing.
- Sample exams from previous years will be made available on Moodle, soon.

Congratulations!
Kol Hakavod ☺

Good luck, stay in touch.