## Programming Research Algorithms: Schedule for 5783

The schedule is subject to change.

|  |  |  |
| --- | --- | --- |
| **Week** | **Lecture topics** | **Assignment** **topics [each assignment has detailed instructions, which are currently written in Hebrew only]** |
| 1. | **Paper:** Reading a research paper: how do you start? What is the paper structure? What to note on first and second reading?  **Python 1:** operators, flow control, functions,args, kwargs, lambda, annotation, files, exceptions, doctest. | **Python**: functions.  **Paper**: choose a paper and get my approval [to week 2].  After approval: summarize paper in your own words [to week 3]. |
| 2. | **Python 2:** OOP, decorators, inheritance, encapsulation, abstract class, magic methods, operator overloading, context manager. | **Python**: OOP |
| 3. | [presentation of papers] | **Paper**: create detailed running examples for the algorithm in your paper. |
| 4. | **Python 3**: design patterns: cache, iterators, generators; strategy pattern. | **Python**: design patterns. |
| 5. | [presentation of running examples] |  |
| 6. | **Python 4**: development process: virtual environments, unitest, pytest, logging, github actions. | **Python+ Paper**: find an open-source library where your algorithm can fit; write headings and tests for your algorithm. |
| 7. | **Python 5**: libraries for scientific programming: numpy, scipy, matplotlib, networkx, cvxpy. | **Python:** num-stack. |
| 8. | [presentation of headings and tests] | **-** |
| 9. | **Python 7**: performance improvements: multithreading, multiprocessing, cython, cppyy, pypy, numba | **Paper:** implement your algorithm. |
| 10. | [presentation of implementations] | **Python+ Paper:** improve performance of your algorithm. |
| 11. | **Python 8**: building simple websites using flask and Google spread **.** | **Python+ Paper:** build a website for demonstrating your algorithm. |
| 12. | **Python 9**: publishing Python libraries in PyPI. | **Python+ Paper:** either pull-request your implementation into an existing library, or publish your algorithm as a new library in PyPI. |
| 13. | [final presentations] |  |