**Introduction to Computer Science**

Reichman University, 2024

Instructors: Kfir Bar and Shimon Schocken

“What I hear, I forget; What I see, I remember; What I *do*, I understand.”

(Confucius, 551-479 BC)

**Objectives:** The course exposes students to computer science fundamentals and equips them with solid programming skills, using a hands-on approach. Key elements of the theory and practice of computer science are presented in the context of programming examples and exercises. Through this experience, students develop an appreciation of the elegance and joy of computer science, and become competent Java programmers. The course assumes no previous experience. Students are expected to be curious and motivated to learn computer science not because it leads to great careers, but because it is fun and intellectually rewarding.

**Methodology:** 13 semester weeks, each consisting of two 2-hour lectures, one 2-hour recitation (תירגול), One 2-hours workshop (סדנה) and a homework assignment. Of these course elements, the homework assignments are by far the most important. Students should expect to work 6-12 hours a week on each homework assignment.

**The course grade** is based on a midterm exam (25%), final exam (60%), and homework assignments (15%). In order to pass the course, students must get at least 60 in the final exam. If the homework assignments are so important, how come they get such a small weight? The answer is that if you will not do the homework on your own, you will not pass the exams.

**Textbook:** *Introduction to Programming in Java: An Interdisciplinary Approach* by Sedgewick and Wayne (any edition is fine). Copies of this book are available at RUNI’s library, as well as in on-line bookstores. The textbook is recommended, but not required.

**English** is the official written language of this course. The spoken language in the lectures is Hebrew in the regular tracks and English in the international track; in all tracks though, all written materials (lecture notes, homework, web site, etc.) are in English. The course provides an excellent opportunity to improve your English in an applied context.

**Questions and answers**are welcome and encouraged. Post your questions on the course's Questions and Answers (Q&A) forum, and they will be answered shortly by the course staff, or by students who know the answer and want to help out.

**The course website** and the Q&A forum are the hub of all the course activities. The lecture slides, homework assignments and course materials for every week will be uploaded to the course website every Sunday. The course website is also the official repository of all the course's announcements, rules and regulations. If we say *x* in a class meeting and the course website says *y*, you must ignore *x* and do *y*. Each student is responsible for visiting the course website and Q&A forum daily and following the guidelines listed there.

**Civilized behavior**in and out the class activities is expected. Please refrain from doing things that would irritate you if you were to teach this course. Examples include being late to class meetings, posting silly messages in the Q&A forum, not following instructor requests during class activities, and so on. Use your judgment.

**Course Plan** (by week)

The plan listed below (next page) is tentative. The actual week-by-week plan will be published in the course website, one week in advance.  The reading references, which are optional, refer to page numbers from *Introduction to Programming in Java: An Interdisciplinary Approach* by Sedgewick and Wayne.

| **Topics**, more or less by week | | **Reading** |
| --- | --- | --- |
| 1 | Welcome to computer science, and to the course. | 1-37 |
|  | Programming basics: variables and data types. |  |
|  |  |  |
| 2 | Conditional and iterative processing. | 45-75 |
|  | Examples: The decimal and binary systems, Monte Carlo simulation. |  |
|  |  |  |
| 3 | Functions and the art of modular programming. | 85-100 |
|  | Examples: Approximation algorithms, string processing. |  |
|  |  |  |
| 4 | One- and multi-dimensional arrays. | 101-110 |
|  | Examples: prime numbers, the PageRank algorithm. |  |
|  |  |  |
| 5 | Multimedia I: handling text, graphics, and sound. | 119-150 |
|  | Multimedia II: handling animation and physical models. |  |
|  |  |  |
| 6 | Mid-semester review, more application examples. |  |
|  | Midterm examination. |  |
|  |  |  |
| 7 | Hardware architectures, machine languages. |  |
|  | Software architectures, compilation, computability. |  |
|  |  |  |
| 8 | Recursion: concepts and techniques. | 253-275 |
|  | Examples: fractals, permutations, artificial landscapes. |  |
|  |  |  |
| 9 | Object-oriented programming I: the client’s perspective. | 314-357 |
|  | Object-oriented programming II: the class-writer’s perspective. | 369-400 |
|  |  |  |
| 10 | Data structures: linked lists.  Sets, stacks, run-time analysis. |  |
|  |  |  |
| 11 | Sorting algorithms, trees. |  |
|  | Example: Compression using Huffman coding. |  |
|  |  |  |
| 12 | Inheritance: basic concepts and techniques. | 415-446 |
|  | Example: virus spreading and drug therapy. |  |
|  |  |  |
| 13 | Machine learning: Introduction and basic concepts. |  |
|  | Basic algorithms and implementations. |  |