

### הפשטה

**Knuth:** "They [computer scientists] are individuals who can rapidly change levels of abstraction, simultaneously seeing things 'in the large' and 'in the small'" (In Hartmanis, 1994; p. 39)

**Dijkstra:** "The effective exploitation of his powers of abstraction must be regarded as one of the vital activities of a competent programmer" (Dijkstra 1972; p. 864)

הפשטה היא אחד מהרעיונות היסודיים של מדעי המחשב (Schwill, 1994). מאמרים רבים מדגישים את החשיבות הרבה של הפשטה כחלק מהיכולות של מדעני מחשב (Wing, 2006) ואת הקושי של תלמידים לפתח יכולת זאת (Aharoni, 2000 ; Armoni & Gal-Ezer, 2006 ; Dijkstra , 1972).

אנו רואים חשיבות רבה בהקניית מיומנויות חשיבה ומתייחסים להוראת הרעיונות היסודיים של מדעי המחשב (כדוגמת הפשטה, רדוקציה, אי דטרמיניזם, אלגוריתם ועוד) כאל מרכיב מהותי בלימודי מדעי המחשב. רעיונות כאלו יש ללמד באופן ספירלי ומגיל צעיר (Bruner, 1960) ובכך להפוך את הלמידה למשמעותית.

### מסגרת המחקר

כחלק מלימודי מדעי המחשב בעתודה המדעית-טכנולוגית נלמדת היחידה "מבוא למדעי המחשב" המיועדת לתלמידי כיתות ז'. במסגרתה פוגשים התלמידים מושגי יסוד מעולם מדעי המחשב ומיישמים אותם בעולם הוויזואלי של שפת Scratch. ניתן ללמד את היחידה בעזרת הספר "רעיונות ומושגים במדעי המחשב דרך "Scratch" (Armoni & Ben-Ari, 2013).

היחידה בנויה סביב ניתוח ופתרון בעיות על ידי תכנון אלגוריתמי, והמושגים והרעיונות הנלמדים בה משתלבים באופן טבעי בתהליכי פתרון הבעיות. בין המושגים והרעיונות המוזכרים: ביצוע חוזר, תנאים, ביצוע מקבילי, משתנים ותקשורת על ידי מסרים.

כל הנושאים הנלמדים (לדוגמה, הבנה של ביצוע מקבילי, הבנה של משתנים או ביצוע חוזר אינסופי) מערבים הפשטה ולעתים רמות שונות של הפשטה. בפרט, תהליך פתרון בעיה דורש מעברים מרובים בין רמות הפשטה שונות, למשל, מעבר מבעיה נתונה לפתרון אלגוריתמי עבורה ומעבר מאלגוריתם לתוכנית מתאימה לו ב-Scratch.

המחקר מתמקד בשאלה האם ניתן ללמד ולשפר את יכולת ההפשטה במדעי המחשב בגיל צעיר, על ידי אסטרטגיית הוראה המבוססת על כמה עקרונות שמובילים להדגשה והדגמה מפורשת של הפשטה (Armoni, 2013). הניסוי נערך במתכונת השוואתית ובוחן את שיטת הלימוד המוצעת.

### רמות הפשטה בפתרון בעיות במדעי המחשב

בעבודה של Perrent ושותפיו (2006) הוגדר סולם למדידת רמות ההפשטה בהקשר של פתרון בעיות במדעי המחשב, כדלקמן:

1. בעיה
2. אובייקט (אלגוריתם)
3. תהליך (תוכנית מחשב)
4. ריצה

במחקר של Perrent (2006) ובעבודות נוספות נמצא שתלמידים בדרך כלל עובדים ברמה שתיים ושלוש ומתקשים לתפוס את ההבדל ביניהם ואת הרמה הגבוהה של הבעיה. בנוסף נצפה שלאורך לימודי תואר במדעי המחשב עולה ההבנה של הפשטה ותלמידים נוטים לעבוד יותר ברמת האובייקט ולהשתמש יותר בהפשטה.

### עקרונות אסטרטגיית ההוראה

1. דיוק ועקביות בהפרדה בין רמות הפשטה שונות בזמן הוראה ודיון
2. הבדלים בשפה ובטרמינולוגיה - שימוש במערכות מושגים שונות לכול רמת הפשטה
3. דיון מהרמה הגבוהה לנמוכה -מהבנת הבעיה דרך האלגוריתם אל המימוש
4. העצמת השימוש והחשיבות של שלב התכנון האלגוריתמי והדגשת החשיבות שלו
5. התייחסות מפורשת לרמת ההפשטה בה עובדים או דנים - " בשלב זה נדון באלגוריתם לפתרון הבעיה"
6. שימוש ברפלקציה

### מערך הניסוי

גישת המחקר היא משולבת, כמותית ואיכותנית. הניסוי נערך מתחילת שנת הלימודים תשע"ז של העתודה המדעית-טכנולוגית, בחמישה בתי ספר ברחבי הארץ, במסגרת הוראת היחידה "מבוא למדעי המחשב". בתחילת השנה תלמידי הכיתות המשתתפות מילאו שאלון רקע וענו על שאלון ידע קצר לבחינת יכולת ההפשטה. בסוף השנה ייערך מבחן מסכם למדידת יכולת ההפשטה בסיום תהליך הלימוד של היחידה. בנוסף כלי המחקר כוללים גם תצפיות אורך, ראיונות עם תלמידים, שיעורי בית, ופרויקטים מסכמים.

סה"כ משתתפים כ-200 תלמידים, ב-9 כיתות של 6 מורים. איסוף נתונים במערך דומה מתוכנן גם בשנת הלימודים הבאה (תשע"ו).

### References

Aharoni, D. (2000). Cogito, Ergo sum! cognitive processes of students dealing with data structures. ACM SIGCSE Bulletin 32.1 , 26-30.  
 Armoni, M. (2013). On Teaching Abstraction in CS to Novices. Journal of Computers in Mathematics and Science Teaching, 32(3), 265-284.  
 Armoni, M., & Ben-Ari, M. (2013). Computer Science Concepts in Scratch. [http://stwww.weizmann.ac.il/g-cs/scratch/scratch\\_en.html](http://stwww.weizmann.ac.il/g-cs/scratch/scratch_en.html): Weizmann Institute of Science.  
 Armoni, M., & Gal-Ezer, J. (2006). Reduction--an abstract thinking pattern: the case of the computational models course. ACM SIGCSE Bulletin. Vol. 38. No. 1. ACM.  
 Bruner, J. S. (1960). The process of education. Harvard University Press.  
 Dijkstra, E. W. (1972). The humble programmer. Communications of the ACM 15.10 , 859-866.  
 Dijkstra, E. W. (1975). About robustness and the like. EWD 452. The Archive of Dijkstra's Manuscripts.  
 Hartmanis, J. (1994). Turing Award lecture on computational complexity and the nature of computer science. Communications of the ACM 37.10, 37-43.  
 Hazzan, O. (2003). How students attempt to reduce abstraction in the learning of mathematics and in the learning of computer science. Computer Science Education 13.2, 95-122.  
 Perrenet, J., & Kaasenbrood, E. (2006). Levels of abstraction in students' understanding of the concept of algorithm: the qualitative perspective. 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITICSE), 270-274.  
 Perrenet, J., & Kaasenbrood, E. (2006). Levels of abstraction in students' understanding of the concept of algorithm: the qualitative perspective. ACM SIGCSE Bulletin. Vol. 38. No. 3. ACM.  
 Schwill, A. (1994). "Fundamental ideas of computer science." BULLETIN-EUROPEAN ASSOCIATION FOR THEORETICAL COMPUTER SCIENCE 53 , 274-274 .  
 Wing, J. M. (2006). Computational Thinking. Communications of the ACM.

