

יום ראשון

נושאים ומושגי יסוד

- מושגים בסיסיים השגורים בפי מי שידוע תכנות מחשבים.
- דיברנו על המבנה הסכמטי של המחשבשבו ישנה יחידת עיבוד מרכזית (CPU), זיכרון וציוד היקפי כמו: מקלדת, מסך, עכבר וכד' (אפשר לכתובם אמצעי קלט/פלט). אמרנו שה-CPU הוא "המפקד העליון" של המחשב, והוא קובע מייעשה מה ומתי.
- 1. מערכת הפעלה - Operating System או בקיצור OS - זוהי תוכנית שבדרך כלל מותקנת על המחשב בזמן הקנייה שלו והיא מאפשרת לאנשים לתקשר עם המחשב. היא למשל שולטת על המקלדת ועל המסך.
- 2. דוגמאות למערכות הפעלה שקיימות היום: Windows, Mac OS, Unix, Linux
- 3. תוכנית מחשב זהו בעצם קובץ שניתן 'לבצע'. הביצוע של התוכנית ניקרא: הרצה (Run)
- 4. ישנן שפות תכנות רבות, אחת מהן שאנחנו נילמד היא: סי שרפ (ניכתב: C#). בשפת התכנות ישנן פקודות שאת התחביר שלהן (Syntax) נילמד בהמשך וגם נילמד מה כל פקודה מבצעת כמובן ומתי להשתמש בה.
- בצורה הפשוטה ביותר ניתן לומר שתוכנית מחשב היא תירגום אלגוריתם כלשהו לשפה שהמחשב מבין, כלומר מי שידוע לתאר את הפעולות הפשוטות של כל צעד באלגוריתם, יוכל לכתוב תוכנית שתיישם את האלגוריתם.

מהו אלגוריתם?

- אלגוריתם - סידרת הוראות מדוייקת ומפורטת מספיק עבור המשתמש, על מנת לבצע פעולה מסוימת. למשל: אלגוריתם לעשיית קפה נמס

1. הרתח מים בעזרת קומקום שמספיקים לפחות לכוס אחת
2. שים כפית של קפה נמס בספל
3. שפוך מהקומקום מים לתוך הספל עם כפית הקפה.
4. הוסף מעט חלב.
5. ערבב היטב.
6. סיים.

- תכונות האלגוריתם: מדוייק וחד משמעי, מפורט מספיק, סופי, פותר בעיות מאותו סוג עם נתוני התחלה שונים, למשל אלגוריתם לחיבור 3 מספרים צריך לתת תשובה נכונה לכל צירוף של 3 מספרים. הנתונים ההתחלתיים של האלגוריתם ניקראים: קלט.
- אלגוריתם יכול להיות מבוסס על הסבר מילולי (כמו ההוראות לעיל), או על ידי תרשים עם צורות שמייצגות פעולות שונות וחיצים שמראים את כיוון זרימת הפעולות - ניקרא תרשים זרימה.
- הרעיון של פיתרון בעיה בעזרת מחשב יכול להיות מורכב ומעניין. אחרי שהגענו לאלגוריתם הנכון, הכתיבה של התוכנית זהו תהליך מכני לא קשה, אם אנחנו בקיאים בשפת התכנות.
- בלימוד תכנות קורה שלא מבינים דברים מסוימים ורק מניחים שהם עובדים כפי שאנו מצפים. בשלבים מאוחרים יותר מתבהרת התמונה ואז חוזרים ומבינים את החלקים שלא הבנו בהתחלה. זה אומר שהלימוד אינו תמיד מ-אלף עד תו, כי אחרת זה יהיה יבש מידי ומשעמם ולכן עושים קיצורי דרך. אם לומדים בצורה טובה, בסופו של דבר הכל הופך להיות מובן וברור.

זה מוביל לתוכנית הראשונה שכתבנו יחד, למרות שעדיין לא ברור מה עשינו בה, אבל הנה תיאור מפורט של מה שעשינו:

הסיבה לשימוש בסביבה, היא כדי להקל על תהליך כתיבת התוכניות. נדבר על נושא זה יותר בהמשך.

כאשר פתחנו את סביבת העבודה, ובחרנו פרויקט מסוג: Console application, התקבל לפנינו מסך דומה לזה (אנחנו כתבנו את השורה האחרונה):

```
using System;

namespace ourprograms
{
    class MainClass
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

הסבר של מה שראינו:

כללי, כל המילים שמופיעות בתכלת למלה (כחול בסביבת העבודה), הן מילים ששייכות לשפה C# (ניקראות 'מילים שמורות').

המשפט הראשון: `using System;` אומר למחשב שאנו מעוניינים להשתמש בתוכנות שקיימות כחלק מהשפה של C# – הפקודה מסתיימת בתו 'נקודה פסיק'

המשפט: `namespace ourprograms` בעצם אומר שכל מה שיבוא לאחריו יהיה שייך לאוסף של תוכניות שניקרא: `ourprograms`. שימו לב שלאחרי שורה זו ניפתחות סוגריים מקושטות – זה אומר שניפתח קטע או בלוק של תכנות, שיסתיים כאשר יופיע סוגר מקושט, כרגע ניכתוב תוכנית אחת בלבד.

לאחר מכן ישנה שורה: `class Mainclass` – השם `Mainclass` זה שם שאפשר לשנות (משהו שאנחנו מגדירים). המילה `class` היא מילה 'שמורה' של השפה C#

השם `Mainclass` הוא בעצם שם התוכנית שלנו. שימו לב שוב לסוגריים המקושטות שמראות היכן מתחילה התוכנית והיכן היא מסתיימת. סביבת העבודה מסדרת לנו יפה לכל פותח { סוגר שמתאים מתחתיו } (בצורה מקוננת).

אחרי השורה הבאה: `public static void Main(string[] args)` היא בעצם המקום שבו התוכנית שלנו מתחילה.

התוכנית שלנו כולה בנויה **מפקודה אחת** בלבד, שאומרת למחשב להדפיס שורה:

```
Console.WriteLine("Hello World!");
```

לאחר מכן יש רק סוגריים מקושטות שסוגרות את כל הקטעים הנ"ל. מה שאמור להיות מודפס זה מה שכתוב בצהוב: `Hello World`. על כל יתר המבנה ואיך אפשר להבין אותו נדבר בשיעור הבא.