

## **מעבדות ליסודות מדעי המחשב בשפות החדשות**

כל החומרים פותחו ע"י המשתתפים בקורס מובילי יסודות.  
ניתן להשתמש בחומרים לצורכי הוראה בלבד.

**לא ניתן לפרסם את החומרים או לעשות בהם שימוש מסחרי כלשהו ללא רשות מראש של המחברים.**

**החומרים בקובץ זה פותחו ע"י**

**דורית כהן**

## מעבדה בנושא "מחרוזות"

חלק א'

```
public class AdvancedString
{
    public static void Main (string[] args)
    {
        string st = "spring";
        Console.WriteLine (st[1] + st[4]);
        Console.WriteLine (st);
        st = "hello"; // ***
        Console.WriteLine (st[1] + st[4]);
        Console.WriteLine (st);
    }
}
```

- הפעילו C# Express
- צרו פרויקט חדש בשם AdvancedString והקלידו את הקוד שבמסגרת
- הפעילו את הפרויקט בעזרת כפתור ההרצה או בעזרת צירוף המקשים Ctrl + F5
- עקבו אחר תוצאות ההרצה.
- שימרו את התכנית (Ctrl + S)
- הריצו את התכנית ורישמו את הפלט המתקבל: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

ההתייחסות למחרוזת היא כמו אל מערך חד-מימדי המכיל תווים. ניתן לגשת למקום מסויים במחרוזת כדי לקרוא את התו הנמצא שם.

- החליפו בתכנית הנ"ל את השורה המסומנת ב \*\*\* בשורה: `st[1] = 't';` והריצו שוב את התכנית. מה היתה תגובת המחשב? \_\_\_\_\_

הגישה למקום מסויים במחרוזת אפשרית רק לצורך קריאה מאותו מקום, אך לא ניתן לשנות כך את ערך המחרוזת. שינוי מחרוזת יכול להיעשות רק על-ידי השמה של ערך מחרוזתי חדש לתוכה. פעולות שמשנות מחרוזת יוצרות מחרוזת חדשה, כפי שנלמד בהמשך...

### הקדמה להמשך המעבדה – המחלקה string

כפי שראינו **מחרוזת** היא סוג נתונים היכול לאחסן מידע המורכב מרצף של תווים. string הוא שם של מחלקה בשפת C# שמגדירה את טיפוס הנתונים מחרוזת ומכילה את כל התכונות והפעולות הקשורות להגדרה זו, וקיימות בשפת C#. להבדיל מטיפוסי הנתונים הבסיסיים בהם השתמשנו עד כה (int, double וכדומה), הערכים של טיפוס הנתונים string הם עצמים. דבר המחייב שימוש שונה בפעולות שמוגדרות על טיפוס נתונים זה. אין אמנם צורך להשתמש בהוראה new ביצירת עצם מטיפוס מחרוזת (כפי שראינו שדרוש ביצירת עצם מטיפוס מערך...), אך כשרוצים להפעיל פעולות של המחלקה string יש לבצע זאת דרך העצם שעליו רוצים להפעיל את הפעולה. דוגמאות:

ההגדרה `string st = "abc";` יוצרת הפנייה לעצם מטיפוס מחרוזת שהערך שלו הוא "abc". את אופן הפעלת הפעולות נכיר מחלק ב' ואילך...

## חלק ב'

```
public static void Main (string[] args )
{
    string st = "abc";
    Console.WriteLine ("first="+ st.Length );
    st = "abcdefg";
    Console.WriteLine ("2nd="+ st.Length);
}
```

- שנו את התכנית בהתאם לנדרש.
- הריצו את התכנית ורישמו את הפלט המתקבל: \_\_\_\_\_
- שנו את הוראת ההשמה השניה למשתנה st כך שתודפס ההודעה 2nd=0
- הגדירו במילים שלכם, מה משמעות הביטוי **st.Length**

משמעות ההוראה: st.Length היא: בדיקת ערך התכונה Length של העצם st.

Length היא תכונה של מחרוזת, שערכה הוא מספר התווים במחרוזת, כלומר: אורך המחרוזת. לכל מחרוזת יש אורך. האורך של המחרוזת הריקה "" הוא 0. שימו לב: את התכונה Length ניתן רק לקרוא ואי אפשר לשנות את ערכה.

## חלק ג'

```
public static void Main (string[] args)
{
    string st1, st2;
    Console.WriteLine ( "Enter 2 strings" );
    st1 = Console.ReadLine();
    st2 = Console.ReadLine();
    if (st1 == st2)
        Console.WriteLine (st1+" is equal to"+st2);
    else Console.WriteLine ("different strings");
    if (st1.Equals(st2))
        Console.WriteLine (st1+" is equal to"+st2);
    else Console.WriteLine ("different strings");
    if (st2.Equals(st1))
        Console.WriteLine (st2+" is equal to"+st1);
    else Console.WriteLine ("different strings");
}
```

- השוואה בין מחרוזות:
- שנו את התכנית בהתאם לנדרש.
  - שימרו את התכנית.
  - הריצו את התכנית מספר פעמים, עם קלטים שונים בכל פעם. למשל: "abcd" ו-"abcd", "abc" ו-"abb", עד שתבינו מה תפקיד ההוראה Equals, וכיצד היא פועלת... ומה ההבדל (אם קיים) בינה לבין השוואה ע"י ==.

ניתן להשוות בין שתי מחרוזות ע"י שימוש בסימן השוויון הכפול, כמו כל השוואה ב C#.

ההוראה Equals משווה גם בין שתי מחרוזות. שתי ההשוואות הללו (== ו- Equals) בודקות אם שתי המחרוזות מכילות את אותו ערך. כאשר שני הערכים שווים – יוחזר הערך true, וכאשר אינם שווים – יוחזר הערך false.

```

public static void Main (string[] args)
{
    string st1, st2;
    Console.WriteLine ( "Enter 2 strings" );
    st1 = Console.ReadLine();
    st2 = Console.ReadLine();
    Console.WriteLine (st1.CompareTo(st2));
    Console.WriteLine (st2.CompareTo(st1));
}

```

#### חלק ד'

- הקלידו את התכנית ושימרו אותה.
- הריצו את התכנית עם כל אחד מזוגות הקלטים הבאים, ורישמו הפלט שהתקבל.
  1. הקלט: "abc" ו-"abc" הפלט: \_\_\_\_\_
  2. הקלט: "abc" ו-"cba" הפלט: \_\_\_\_\_
  3. הקלט: "cba" ו-"a" הפלט: \_\_\_\_\_
  4. הקלט: "ab" ו-"hello" הפלט: \_\_\_\_\_
- הריצו את התכנית עם זוגות קלטים נוספים וכיתבו במילים שלכם את מסקנתכם: ההוראה `CompareTo` מבצעת \_\_\_\_\_ ומחזירה: \_\_\_\_\_ כאשר \_\_\_\_\_ או: \_\_\_\_\_ כאשר \_\_\_\_\_ או: \_\_\_\_\_ כאשר \_\_\_\_\_
- תנו דוגמה נוספת של זוג קלטים, שעבורם זימון ההוראה `st1.CompareTo(st2)` יחזיר את הערך 0.
- תנו דוגמה נוספת של זוג קלטים, שעבורם זימון ההוראה `st1.CompareTo(st2)` יחזיר את הערך -1.
- תנו דוגמה נוספת של זוג קלטים, שעבורם זימון ההוראה `st1.CompareTo(st2)` יחזיר את הערך 1.

ההוראה `CompareTo` מבצעת השוואה לפי סדר לקסיקוגרפי (מילוני) בין הערכים של שתי מחרוזות. כאשר יש שוויון בין המחרוזות, ההוראה מחזירה 0. כאשר המחרוזת הראשונה גדולה מהמחרוזת השניה יוחזר ערך חיובי. כאשר המחרוזת הראשונה קטנה מהמחרוזת השניה יוחזר ערך שלילי.

## חלק ה'

```
public static void Main (string[] args)
{
    string st1 = "happy birthday";
    string st2, st3, st4;
    st2 = st1.Substring (6);
    st3 = st1.Substring (11);
    st4 = st2.Substring (5);
    Console.WriteLine ("st1= " + st1);
    Console.WriteLine ("st2= " + st2);
    Console.WriteLine ("st3= " + st3);
    Console.WriteLine ("st4= " + st4);
}
```

- הקלידו את הקוד שבמסגרת.
- הריצו את התכנית וצפו בפלט.
- הקלידו מחרוזת אחרת כרצונכם לתוך st1 וצפו בפלט.
- כיתבו את מסקנתכם: ההוראה Substring(n) המופעלת על מחרוזת, מחזירה

בהתאם למספר n שקיבלה כפרמטר.

הפעולה Substring תת-מחרוזת מהמחרוזת המקורית, החל ממיקום מוגדר. תזכורת: המקום של התו הראשון במחרוזת הוא 0 (אפס).

```
public static void Main (string[] args)
{
    string st1 = "happy birthday";
    string st2, st3;
    st2 = st1.Substring (6, 5);
    st3 = st2.Substring (3, 2);
    Console.WriteLine ("st1= " + st1);
    Console.WriteLine ("st2= " + st2);
    Console.WriteLine ("st3= " + st3);
}
```

- שנו את התכנית בהתאם לנדרש:
  - הריצו את התכנית וכיתבו מה יהיה הפלט.
  - הקלידו מחרוזת אחרת כרצונכם לתוך st1 וצפו בפלט.
  - כיתבו את מסקנתכם: ההוראה Substring(n, m) המופעלת על מחרוזת, מחזירה
- כאשר n קובע את \_\_\_\_\_
- ו- m קובע את \_\_\_\_\_.

הפעולה Substring(n, m) מחזירה תת-מחרוזת באורך m מהמחרוזת המקורית, החל ממיקום מוגדר n.

עבודה נעימה!