

מבוא לתכנות

הרצאה 4 – לולאות For ולדאבג

סמסטר 1

# חזרות – לולאות For ו While

- לרוב, עלינו לחזור על פעולה מספר פעמים.  
דוגמה פשוטה:

```
int i = 1;
while (i <= 5)
{
    Console.WriteLine("Hello " + name);
    i++;
}
```

- פתרון אחר:

```
for (int i = 1; i <= 5; i++)
{
    Console.WriteLine("Hello " + name);
}
```

הפתרון הנ"ל הינו אלגנטי וקריא.

# חזרות – לולאת For

הצהרת for –  
התחלת הלולאה

צעד האתחול – חד פעמי  
בתחילת הלולאה.  
המשתנה יוכר רק בתוך  
הלולאה

צעד הקידום –  
בסוף כל איטרציה

```
for (int i = 1; i <= 5; i++)  
{  
    Console.WriteLine("Hello " + name);  
}
```

קוד פנימי:  
הפעולות שיש  
לחזור עליהן

תנאי הלולאה –  
צריך להיות אמת  
על מנת להכנס  
לאיטרציה

- ניתן להשתמש במשתנה השליטה:

```
for (int i = 3; i < 8; i++)  
{  
    Console.WriteLine(i);  
}
```

- תוצאה:

```
3  
4  
5  
6  
7
```

# Example

- מה קורה בקוד הבא?

```
for (int i = 7; i > 2; i--)  
{  
    Console.WriteLine(i);  
}
```

- תוצאה :

```
7  
6  
5  
4  
3
```

# לולאה אינסופית

```
for (int i = 0; i < 5; i--)  
{  
    Console.WriteLine(i);  
}
```

- מה הבעיה עם הקוד הנ"ל?
- אם האיטרציה אמורה להפסיק בהתאם לתנאי כלשהו, עלינו לדאוג שהתנאי יהפוך ל"שקר" בשלב זה או אחר במהלך הלולאה.
- אחרת, הלולאה לא תפסיק לעולם!
- זו נקראת "לולאה אינסופית".
- יצירת לולאות אינסופיות בשוגג זו טעות קידוד נפוצה.

# הכרזת break

- אנו יכולים להפסיק את לולאת for באופן מידי באמצעות הכרזת :break

```
for (int i = 3; i < 8; i++)  
{  
    Console.WriteLine(i);  
    if (i == 5)  
        break;  
}
```

3  
4  
5

לא אותו דבר

```
for (int i = 3; i < 8 && i !=5 ; i++)  
{  
    Console.WriteLine(i);  
}
```

3  
4

- Break גורם לתוכנית להפסיק מידית את הרצת הלולאה ולהמשיך לשורות הקוד הבאות אחריה
- למרות ששימוש ב Break הוא לגיטימי, הוא נחשב ל"מוצא אחרון". אנו נעדיף בדרך כלל לעצור את פעולת הלולאה באמצעות כתיבת קוד אשר הופך את תנאי הלולאה ל"שקר".

# הכרזת continue

- אנו נוכל להפסיק רק את האיטרציה (הסיבוב) הנוכחית של הלולאה באופן מידי באמצעות שימוש ב:continue

```
for (int i = 3; i < 8; i++)  
{  
    if (i == 6)  
        continue;  
    Console.WriteLine(i);  
}
```

פלט:

3  
4  
5  
7

- continue גורם לתוכנית להפסיק את האיטרציה הנוכחית ולעבור לאיטרציה הבאה.



# תוכנית המסורין

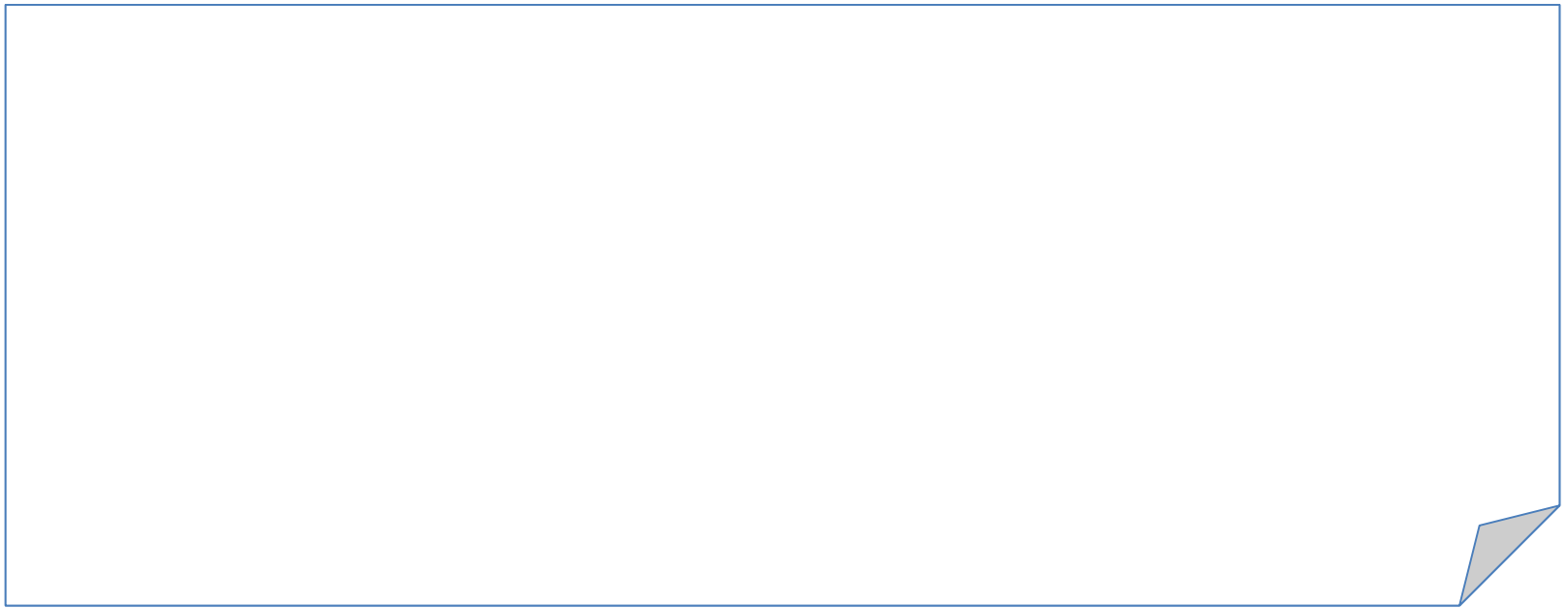
```
int sum = 0;
int n = int.Parse(Console.ReadLine());

for (int i = 0; i <= n; i++)
{
    sum += i;
}
Console.WriteLine(sum);
```

- מה התוכנית הזו עושה?
- מהו n? מהו i? האם אפשר להשתמש בו אחרי הלולאה?
- מהו sum?
- נוכל לכתוב אותה בדרכים אחרות ולקבל את אותה התוצאה?

# תרגיל 1

- כתבו את התוכנית `factorial_n`
- עצרת -  $Factorial\_n = n!$



## תרגיל 2

- כתבו את התוכנית 7-בום.
- קראו מספר מקסימלי מהמשתמש והדפיסו את כל המספרים מ1 ועד המקסימום, אך כאשר המספר להדפסה מתחלק ב7, הדפיסו "Boom" במקום.

# תרגיל 3

- כתבו תוכנית שקוראת מספרים מהמשתמש, עד שהוא רושם את המילה "done".
- ואז, התוכנית מדפיסה את הממוצע של כל המספרים שהוכנסו.

## תרגיל 4

- כתבו תוכנית שקוראת מספר מהמשתמש ומחליטה אם הוא ראשוני או לא.

- כלי אשר מאפשר לנו לראות כל דבר שהמחשב עושה בזמן ריצה.

- ערכים של משתנים שהמשתמש יצר

- ערכים של המשתנים הסביבתיים

- שימוש בזיכרון

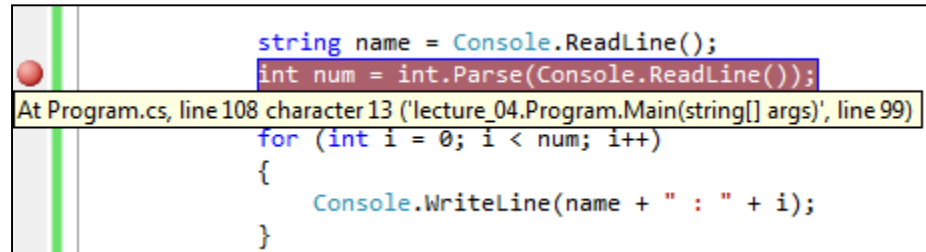
- שורת הקוד שמופעלת באותה עת

- Stack

- תוך שימוש במידע זה, ניתן בקלות למצוא את שגיאות התכנות.

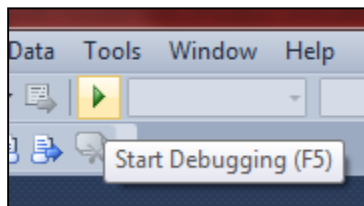
# דיבאג

- כדי להתחיל עלינו לקבוע "נקודת שבירה" (BP-*breakpoint*) בכל מקום שבו נרצה להתחיל את הדיבאג. לחצו על F9 או על הצד השמאלי של החלון. השורה תיציב באדום.



```
string name = Console.ReadLine();
int num = int.Parse(Console.ReadLine());
At Program.cs, line 108 character 13 ('lecture_04.Program.Main(string[] args)', line 99)
for (int i = 0; i < num; i++)
{
    Console.WriteLine(name + " : " + i);
}
```

- הצעד הבא הוא להתחיל לדאבג עם F5, לחצן ה"Play" למעלה או בתפריט ה-Debug. התוכנית תתחיל לרוץ ותעצור ב-BP.

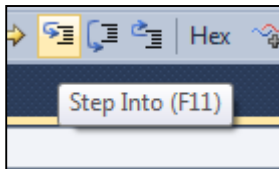


# דיבאג

- ניתן לראות כי שורת הקוד שעתידה להתבצע צבועה בצהוב.

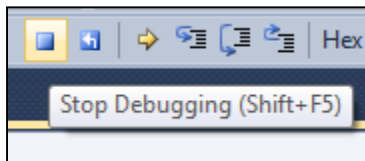
```
string name = Console.ReadLine();  
int num = int.Parse(Console.ReadLine());  
  
for (int i = 0; i < num; i++)  
{  
    Console.WriteLine(name + " : " + i);  
}
```

- נוכל להריץ את השורה באמצעות F10/F11 או כפתורי *step over\into*. כל לחיצה כזו תקדם את התוכנית הלאה בשורה אחת.



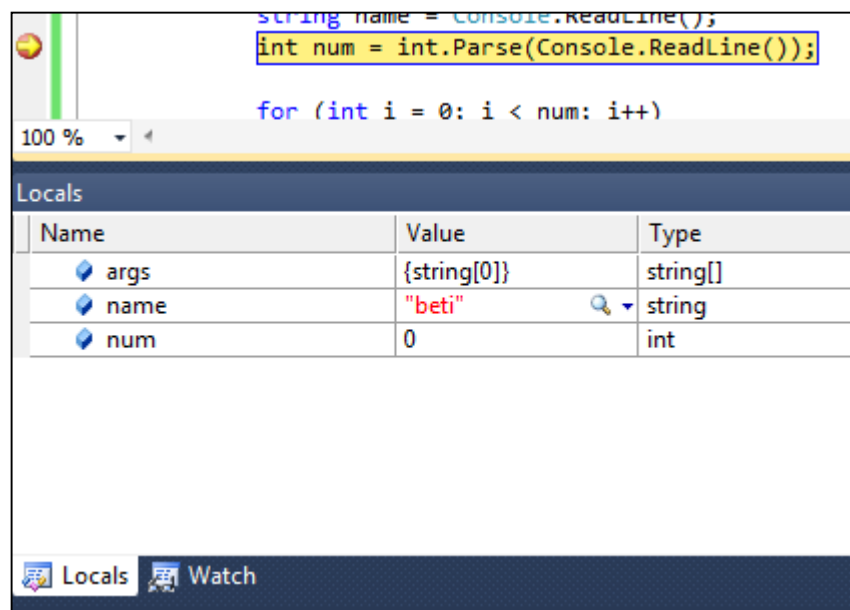
- כדי להגיע ישירות אל ה BP הבאה ניתן ללחוץ על F5 או כפתור ה "play" שוב.

- ניתן לעצור על ידי shift+F5 או כפתור ה "stop".





- בחלון ה *Locals* ניתן לראות את כל המשתנים שהגדרנו (ועוד כמה), וכמו כן את ערכם הנוכחי.



- באמצעות שימוש בחלון ה *Watch* נוכל "לשחק" עם משתנים ספציפיים לבחירתנו.

