השלמות

<u>קיצורי דרך בחישובים מספריים</u>

```
a += b;    // The same as:   a = a+b;
a  -= b;   // The same as:  a = a-b;
a  /= b;   // The same as:  a = a/b;
a  *= b;   // The same as:  a = a*b;
```

לדוגמה: אם a=5 ו- b=6 ונכתוב: a+=b; הערך של a ישתנה ל- 11 ו- b ישאר ללא שינוי.
אם a=5 ו- b=6 ונכתוב: a *=b; הערך של a ישתנה ל- 30 ו- b ישאר ללא שינוי.

הפעולה ++ (פלוס פלוס) המוצמדת למשתנה, מעלה את ערכו באחד.

הפעולה - - (מינוס מינוס) המוצמדת למשתנה, מורידה את ערכו באחד.

למשל:

```
int k1 = 10,  k2 = 8;   //   הגדרה של 2 משתנים מוג שלם עם השמת ערך לכל אחד
++k1;                   //   k1  is now 11
- -k2;                  //   k2  is now 7
```

את סימני הפלוס פלוס (וגם מינוס מינוס) אפשר גם לשים לאחר המשתנה כך: k1++ או k2- -

ההבדל בין הצמדת הקיצור לפני או אחרי המשתנה הוא כזה:

- ++a פירושו: הגדל את a, השתמש בערך החדש לפעולה,
- a++ פירושו: השתמש בערכו הנוכחי של a, אח"כ הגדל את a,
- --a פירושו: הקטן את a, השתמש בערך החדש לפעולה,
- a-- פירושו: השתמש בערכו הנוכחי של a, אח"כ הקטן את a,
- למשל:

```
a = 1;
b = a++ + 5;
(a = 2,b = 6)
```

עוד פעולה מתמטית שמושית בשפה היא פעולת האחוז (המודולו) - כלומר השארית בחלוקת שלמים:

11 % 5 = 1

כאן חילקנו את 11 ב-5 התוצאה בשלמים היא 2 והשארית היא 1

דוגמאות נוספות:

20 % 2 = 0
15 % 6 = 3

שימוש אפשרי של פעולה זו היא לבדוק אם מספר הוא זוגי או אי-זוגי - כיצד נעשה זאת?

כתבו תוכנית שקולטת מספר שלם ומדפיסה אם הוא זוגי: even , אחרת תדפיס: odd

<u>שמות משתנים וסימני יחס</u>

שם משתנה יכול להכיל אותיות, ספרות וגם קווים תחתונים.שם משתנה אינו יכול להתחיל במספר או להכיל רווח. שם משתנה אינו יכול להיות מילה שמורה בשפה, כמו למשל: class

אגלה לכם סוד שהרבה מתכנתים לא יודעים. אפשר ליצור משתנה שהוא מילה שמורה אם נצמיד לו בתחילה את הסימן @ למשל: @else הוא משתנה חוקי לחלוטין.

מצד שני אם נגדיר 2 משתנים כך:

```
string name;
string @name;
```

נקבל הודעת שגיעה שהגדרנו 2 משתנים בעלי אותו שם, כלומר תוספת ה-@ אינה יוצרת משתנה חדש.

כאשר אנו משוים בין שני ערכים, כמו למשל בפקודת התנאי: if (a > b) הפעולות העומדות לרשותנו הן:

a > b	האם a גדול מ- b
a>=b	האם a גדול או שווה ל- b
a<b	האם a קטן מ- b
a<=b	האם a קטן או שווה ל- b
a==b	האם a שווה מ- b
a !=b	האם a שונה (לא שווה) מ- b

<u>גורמי הדפסה</u>

על מנת להדפיס גם הודעות וגם ערכי משתנים בפקודת ה- Console.Write או Console.WriteLine אפשר למשל לעשות את הדבר הבא:

```
int k3=7;
double num= 3.4;
Console.WriteLine (“ The first number is: {0}  and the second is:  {1}” , num,  k3);
```

מה שיודפס זה:

```
The first number is: 3.4 and the second is: 7
```

כלומר הסוגריים המסולסלים מכילים ״תופסי מקום״ עבור המשתנים שבאים לאחר המחרוזת המודפסת. המשתנה מספר אפס {0} בדוגמה מתייחס לראשון אחרי המחרוזת שהוא: num והשני הוא מספר אחד {1} שמתייחס למשתנה השני שהוא k3.

ההבדל בין Write ובין WriteLine מתבטא במה שקורה לאחר ההדפסה. אם משתמשים ב- WriteLine המחשב ירד לשורה הבאה ולכן אם נדפיס דבר נוסף, הוא יודפס בתחילת השורה הבאה. על ידי שימוש ב- Write המחשב לא ירד שורה אלא רק ׳זז׳ ימינה ולכן בהדפסה הבאה נמשיך באותה שורה מימין לערך שהודפס קודם.

לדוגמה:

```
Console.WriteLine( “After this line, go to the next line”);
Console.Write( “After this - no new line ==> ”);
Console.Write( “Same line as before”);
```

כך יראו התוצאות:

```
After this line, go to the next line
After this - no new line ==> Same line as before
```

<u>סוגי משתנים נוספים</u>

ישנו סוג של משתנה שיכול להכיל תו אחד (כל תו שרוצים). זהו הסוג: char (קיצור של המילה character). כדי להציב ערך, משתמשים בצמד של גרשים בודדים כך:

```
char a1= ‘B’;
```

זה בניגוד לסוג שכבר למדנו שנקרא מחרוזת string שמצויין בעזרת צמד של גרשיים. במילים אחרות אם יש צורך ביותר מתו אחד, נשתמש ב- string.

אם תרצו לקרוא מהקלט תו אחד בודד, תצטרכו להשתמש ב- char.Parse (בדומה ל- int.Parse עבור שלמים).

עוד סוג משתנה שהשפה מספקת הוא bool. זהו סוג משתנה (שנקרא גם בוליאני) שיכול להכיל רק שני ערכים: true או false

משתנה מסוג זה משמש לפעמים עבור החלטות כאלה או אחרות בתוכנית.

דוגמה לשימוש: אם רצינו להרחיב את תוכנית ההמרה מצלזיוס לפרנהייט ולהדפיס שהיום הוא ׳יפה׳ על פי הטמפרטורה וגם במקום אחר בתוכנית היינו בודקים אם ירד או לא ירד גשם, ורק הצירוף של טמפרטורה נוחה ומצב שאין גשם היו התנאים להדפסת ׳יום יפה׳. במצב כזה יכולנו להגדיר משתנה בוליאני שנקרא : nice

שיקבל אמת (true) אם הטמפרטורה נוחה, ובהמשך בודקים אם לא ירד גשם וגם nice הוא אמת, אז נדפיס: nice day

זה היה נראה בערך כך:

```
bool nice= false;   //first we make it false

//Here is the conversion fahr_temp=C*9/5 + 32

if (fahr_temp >= 60 && fahr_temp < 80)
{
   nice=true;
}
//Check if raining here

if (no_rain && nice)
{
   Console.WriteLine(“It’s a nice day”);
}
```