

## **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**

**בילצ'יק יעל, גרין עדי**

## מעבדה – מתחילים לתכנת

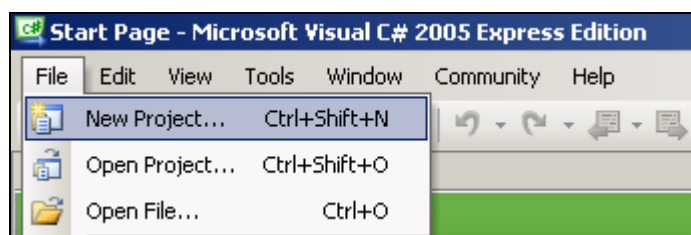
סביבת העבודה Visual C# Express מאפשרת לנו לכתוב תוכניות בשפת C#. במעבדה זה נלמד כיצד להשתמש בסביבת עבודה זו, כיצד נכתוב תוכנית, כיצד נכתוב פלט למסך, כיצד נקבל קלט מהמשתמש, וכיצד נוכל להשתמש במידע הקיים לנו

### פתיחת סביבת העבודה

כדי לפתוח את סביבת העבודה נבחר את Microsoft Visual C# 2005 Express Edition, דרך תפריט ההתחלה של "windows". יתקבל מסך הפתיחה של סביבת העבודה.

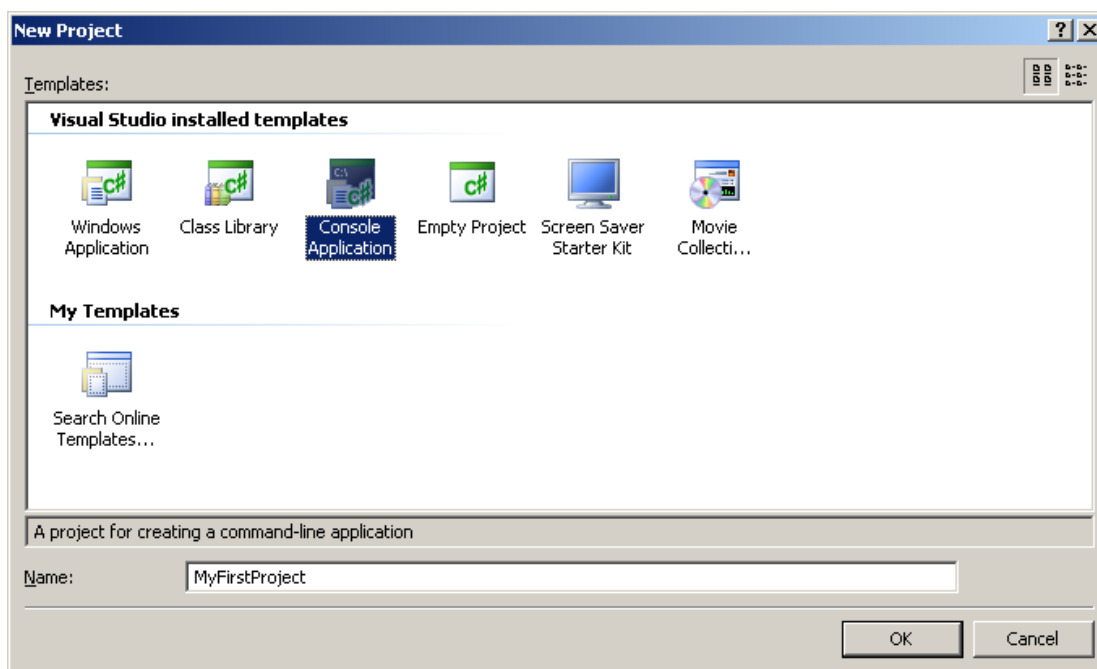
### יצירת פרויקט

כל תוכנית בשפת C# נמצאת בתוך פרויקט נפרד. לכן, כדי לכתוב תוכנית חדשה עלינו לפתוח פרויקט חדש. בחר בתפריט העליון את File, ושם בחר ב-New Project.



יפתח חלון המציע לך סוגי פרויקטים שונים. בחר ב-Console Application.

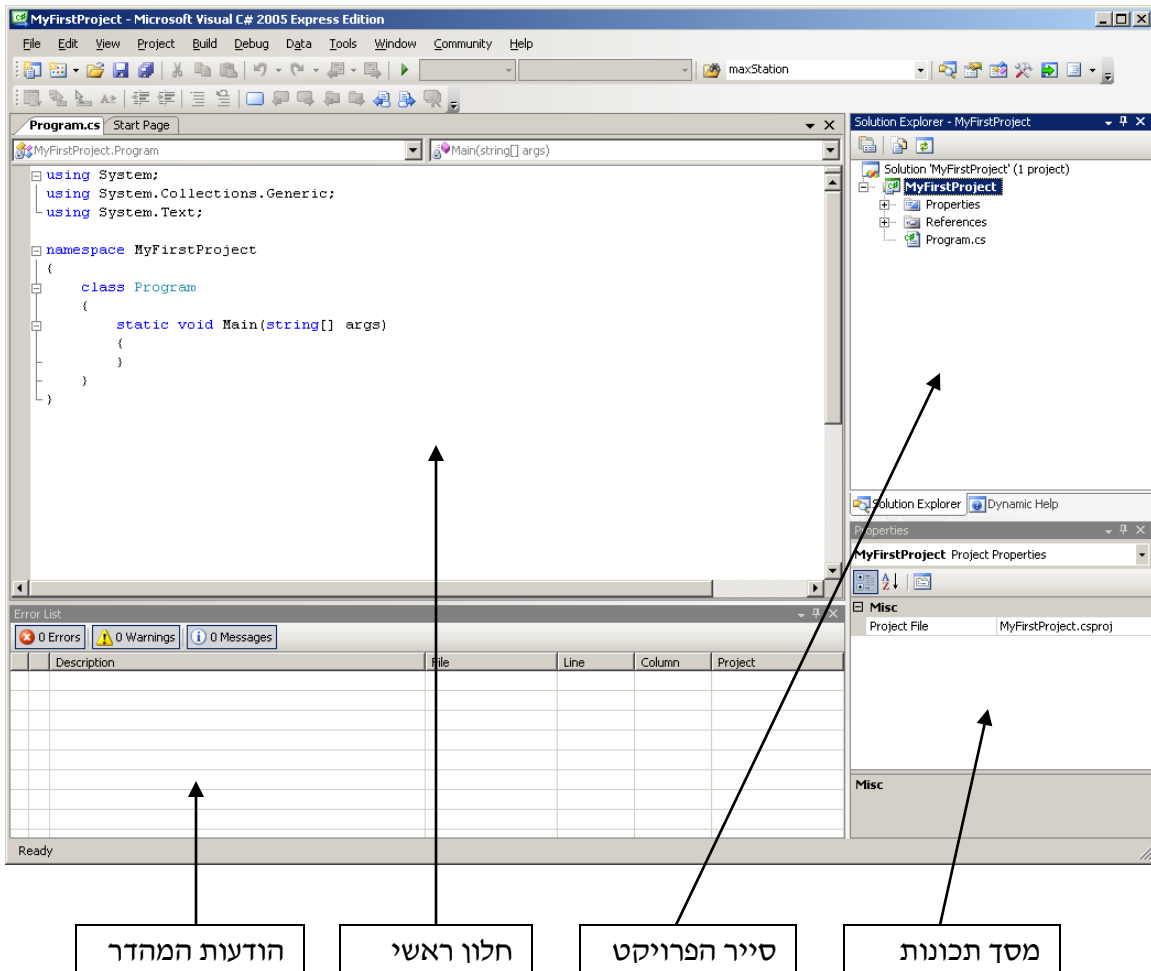
בשורת השם כתוב את השם שברצונך לקרוא לפרויקט. בדוגמה זו בחרנו בשם "MyFirstProject"



לאחר הלחיצה על "OK" יתקבל מסך סביבת העבודה, כאשר בחלון המרכזי כבר מופיע בסיס של תוכנית ריקה בשפת C#.

## חלונות סביבת העבודה

נכיר את החלונות השונים בסביבת העבודה:



החלון הראשי יישמש אותנו לכתיבת קוד התוכנית

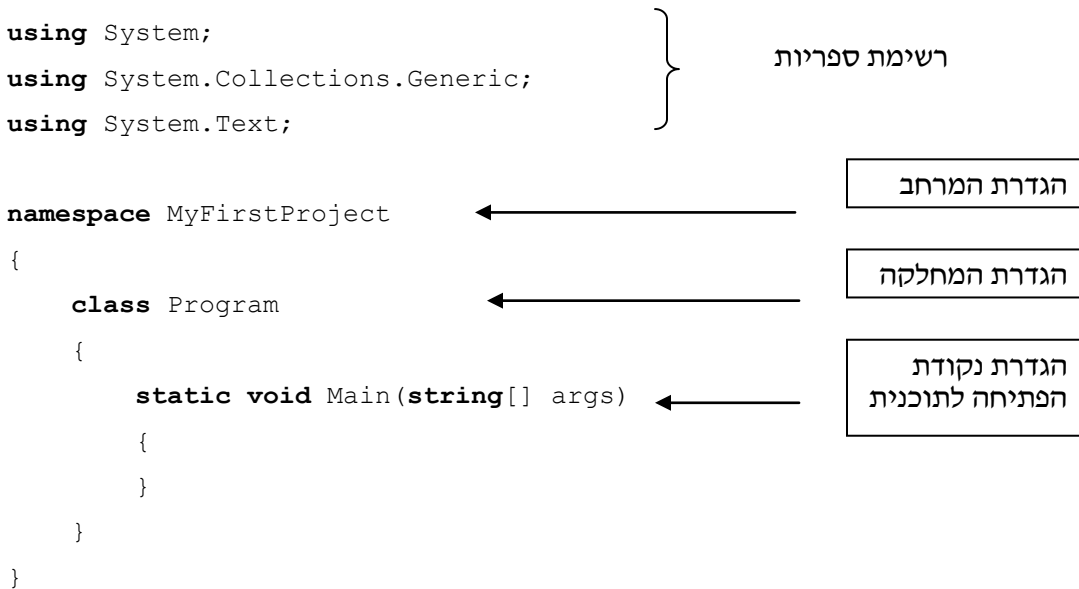
בחלון הודעות המהדר נצפה בשגיאות והערות המהדר (קומפיילר) לאחר כל הידור (קומפילציה).

בחלון סייר הפרויקט נשתמש על מנת לעבור בנוחות בין חלקי הפרויקט השונים.

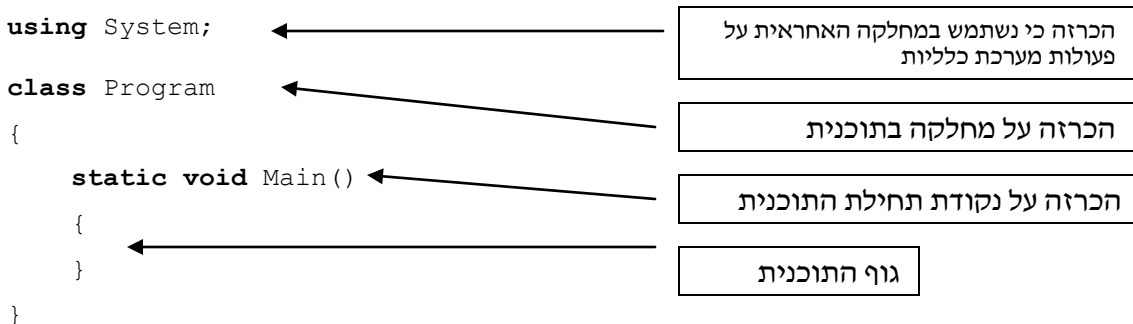
חלון מסך התכונות משמש בעיקר עבור תוכניות בהן נעשה שימוש בעזרים גראפיים ולכן לא נשתמש בו.

ניתן לסגור כל חלון אם אין בו שימוש, ולפתוח אותו שוב בעזרת התפריט View.

נתבונן בקוד (כלומר, רצף הוראות בשפת התכנות) אשר מופיע באופן אוטומטי במסך התוכנית עם פתיחת פרויקט חדש:



לא כל ההגדרות הכרחיות לצרכינו, לכן תוכל למחוק את ההגדרות שלא נזדקק להן ולהישאר עם ההגדרות הבאות המספיקות לצרכינו:



## כתיבת תוכנית ראשונה - פלט

כעת שלד התוכנית מוכן, אך גוף התוכנית ריק מתוכן. כעת אנו יכולים להקליד את התוכנית הראשונה, שתציג למסך את הפלט "Hello world".

שנה את שם המחלקה לשם שתרצה לקרוא לתוכנית, לדוגמא: HelloWorld (אין חובה לשנות את השם, אך רצוי תמיד לתת לתוכניות שמות משמעותיים, שמביעים את תפקידן), הוסיף את פקודת ההדפסה בתוך תחום ה-Main:

```

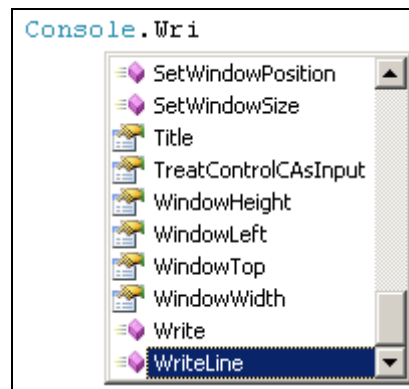
using System;

class HelloWorld
{
    static void Main()
    {
        Console.WriteLine("Hello world");
    }
}

```

## שימו ♥:

- ישנם צבעים שונים בקוד הכתוב: מילים שמורות נכתבות בכחול, מחרוזות באדום, שמות מחלקות בטורקיז.
- מייד לאחר כתיבת שם המחלקה Console נפתח חלון המציג את כל התכונות והפעולות של המחלקה בהן נוכל להשתמש. לפקודת הדפסה נבחר את הפעולה WriteLine.



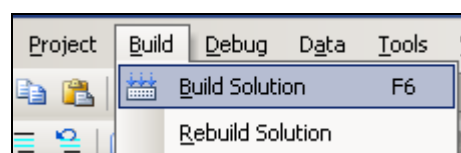
תוכלו לעבור ולבדוק תכונות ופעולות נוספות השייכות למחלקה Console.

- כדי לקבל הנחיות על כל תכונה ופעולה אפשר לעמוד עם הסמן על התכונה או הפעולה המבוקשת ולהקיש על F1.

## הידור תוכנית

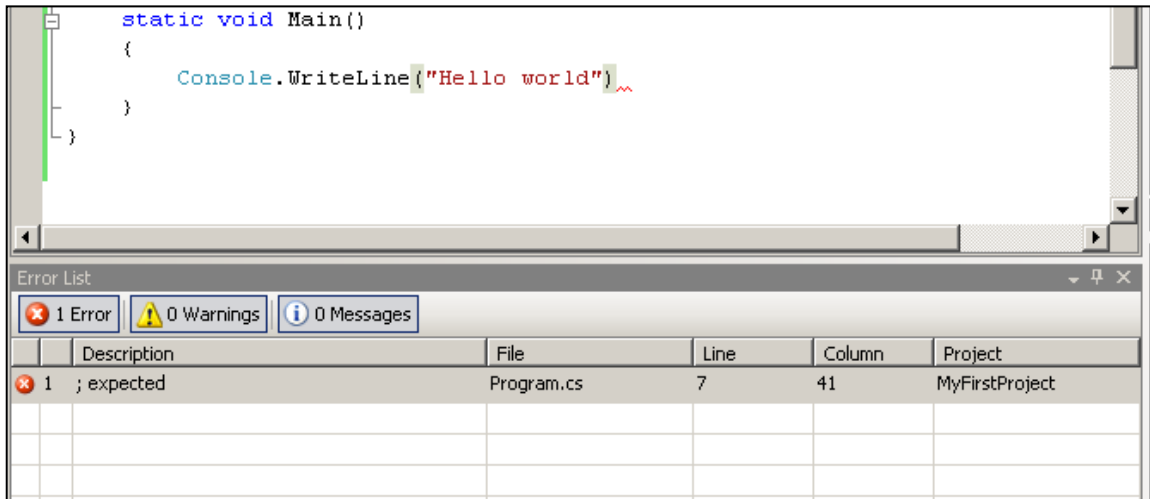
אחרי סיום הקלדת התוכנית, עלינו להדר (לקמפל) אותה כדי לבדוק שאין בה שגיאות תחביריות, ולהכין אותה לריצה. שלב זה יתבצע על ידי בחירת Build בתפריט, ואז בחירת Build Solution, או על ידי הקשה על המקש F6.

הדר את התוכנית שכתבת



אם אין כלל שגיאות תחביר בתוכנית, נקבל את ההודעה Build succeeded בצד השמאלי התחתון של המסך. אם קיימות שגיאות תחביר בתוכנית נקבל הודעות מתאימות בחלון הודעות המהדר. לחיצה כפולה על ההודעה תקפיץ את הסמן למקום בו ארעה השגיאה.

למשל, השגיאה הבאה נגרמה כיוון שלא נכתב הסימן ; בסוף משפט:



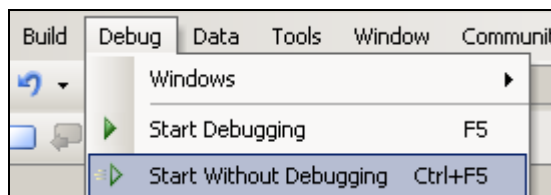
לא נוכל להריץ תוכנית לפני שנתקן את כל שגיאות התחביר. לאחר שנתקן את השגיאות, נהדר שוב את התוכנית, וכך נחזור על התהליך עד אשר לא יופיעו שגיאות בחלון הודעות המהדר, ונקבל את ההודעה Build succeeded.

**שימו ♥:** ייתכן שלאחר הידור של תוכנית נקבל **הערות הרצה**. ההערות מסומנות בסימן צהוב (בעוד השגיאות מסומנות באדום). ניתן להריץ תוכנית שהתקבלו עבורה הערות, אך יש לתת את הדעת על הערות אלה, משום שיתכן שהן מצביעות על טעות או על בעיה אפשרית אחרת בתוכנית. במידה וקבלת שגיאות, תקן אותן והדר את התוכנית שכתבת שוב עד אשר ההידור יעבור בהצלחה

## הרצת תוכנית

לאחר שהתוכנית עברה בהצלחה את שלב ההידור, היא מוכנה להרצה.

כדי להריץ את התוכנית נבחר בתפריט Debug את Start Without Debugging או נקיש על המקשים Ctrl ו-F5 בו זמנית.



כתוצאה מכך, יפתח חלון ריצת התוכנית, ייכתב בשורה נפרדת המשפט "Hello world", וריצת התוכנית תסתיים. בסיום ריצת התוכנית חלון ההרצה ייסגר רק לאחר הקשה על מקש כלשהו.

```

Hello world
Press any key to continue . . . _

```

**שימו ♥:** אם תנסו להריץ תוכנית אשר לא עברה הידור בהצלחה, או תוכנית שעברה הידור אך לאחר מכן עברה שינוי כלשהו ויש להדרה שוב, יתבצע הידור באופן אוטומטי.

הרץ את התוכנית שכתבת וודא כי הפלט זהה לפלט בו אתה מעוניין

**משימה:**

1. שנה את התוכנית כך ששמך יופיע על גבי חלון הפלט
2. שנה את התוכנית כך ששמך יופיע על גבי חלון הפלט מוקף בכוכביות. למשל, אם השם הוא יאיר, הפלט יהיה:

```

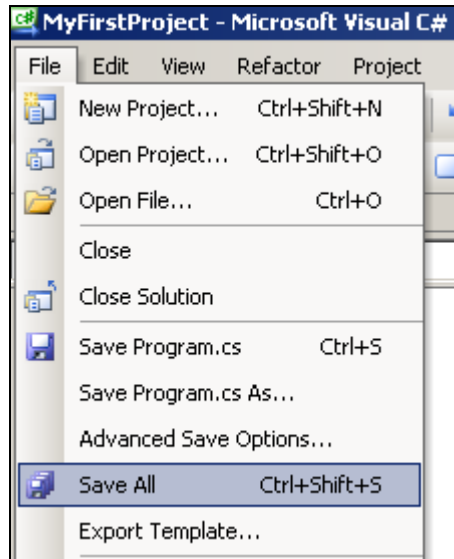
*****
*Yair*
*****

```

הדרכה: פקודת WriteLine מדפיסה שורה אחת למסך. בתוכנית זו עליכם להדפיס 3 שורות.

## שמירת תוכנית

לאחר שסיימנו לכתוב את התוכנית, להדר אותה (תוך תיקון שגיאות תחביר, במידת הצורך), להריץ אותה, ולבדוק שאין שגיאות לוגיות בתוכנית, נשמור את קובץ התוכנית לשימוש עתידי. בתפריט File בחרו את Save All או הקישו על המקשים Ctrl, Shift ו-S בו זמנית.



ייפתח חלון בו יופיע שם הפרויקט, הכתובת על גבי הדיסק הקשיח בה יישמר הפרויקט, ושם ה-Solution.

כאשר נרצה בעתיד לפתוח תוכנית שמורה נפתח את הפרויקט שלה על ידי Open project שבתפריט File, ונבחר את ה-Solution הנדרש לפי שמו בסיומת .sln.

שמור את התוכנית שכתבת בשם HelloWorld על גבי הדיסק הקשיח, וסגור את הפרויקט על ידי **File שבתפריט Close Solution**



## כתיבת תוכנית שנייה - קלט

פתח פרויקט חדש כפי שעשינו בתחילה, ומחק את שורות הקוד המיותרות עבור צרכינו. כעת, עלינו לכתוב תוכנית המקבלת מידע (קלט) שני מספרים שלמים (המופרדים ברווח), ומדפיסה כפלט את ההודעה: "שני המספרים שנקלטו הם: " ואת שני המספרים. הנתונים הנקלטים נשמרים בתאי הזיכרון המכונים "תאי משתנים" או בקיצור "משתנים".

**הגדרה:** משתנה (variable) הוא תא זיכרון אשר ניתן לשמור בו ערך ולקרוא את הערך השמור בו במהלך ביצוע אלגוריתם. למידע השמור בתוך המשתנה קוראים **ערך** המשתנה. פנייה למשתנה נעשית באמצעות שמו, שהוא **שם** המשתנה.

הקלט בבעיה זו הוא שני נתונים נפרדים (שני מספרים), ולכן נגדיר שני משתנים ונקרא להם: num1, num2. בכל אחד מהמשתנים יישמר ערך אחד בלבד. קליטת הנתונים מתבצעת באמצעות הוראה: `int.Parse(Console.ReadLine());` שם משתנה פקודה זו קולטת ערך ומכניסה אותו לתוך המשתנה ששמו כתוב בצד שמאל. בעת ביצוע התוכנית, כאשר תתבצע פעולה זו, תעצור התוכנית ותחכה לקלט מהמשתמש. נהוג להוסיף לפני כל הוראת קלט, הדפסה למסך המסבירה למשתמש בתוכנית מה מצפים ממנו להקליד.

**כתוב בגוף התוכנית את ההוראות הבאות:**

```
Console.WriteLine("Insert first number Please: ");
num1 = int.Parse(Console.ReadLine());
Console.WriteLine("Insert second number Please: ");
num2 = int.Parse(Console.ReadLine());
```

כעת נכתוב את ההוראות הדרושות לכתיבת פלט התוכנית ונדפיס את ערכי המשתנים על המסך

**הוסף את ההוראה:**

```
Console.WriteLine("The two numbers are: {0} {1}", num1, num2);
```

להוראה זו שני חלקים: חלק ראשון – המשפט שיודפס על גבי המסך, וחלק שני – רשימת המשתנים אשר ישתלו במשפט: num1, num2 .

הוראה זו כותבת על גבי המסך את המשפט: The two numbers are: ומייד אחריו שותלת במשפט את שני הערכים במקום בו קיים סמן מתאים. המשתנה num1 יישתל במקום הסמן {0} כיוון שהוא המשתנה הראשון ברשימה, והמשתנה num2 יישתל במקום הסמן {1} כיוון שהוא השני ברשימה.

**הדר את התוכנית (על ידי ctrl+F5)**

האם זיהית כי התוכנית לא סיימה את שלב ההידור בהצלחה? זאת כיוון שיש להצהיר על כל משתמש לפני השימוש בו. הצהרה נעשית ע"י כתיבת **טיפוס** המשתנה ושמו.

הגדרה: **טיפוס** (type) הוא סוג הערכים שיכול משתנה להכיל בתוכו.

כיוון שבחרנו במספרים שלמים, נגדיר את שני המשתנים כטיפוס **int** (מספר שלם = integer).  
הצהרה מתבצעת בשורה הראשונה של התוכנית (בתוך תחום ה-Main)

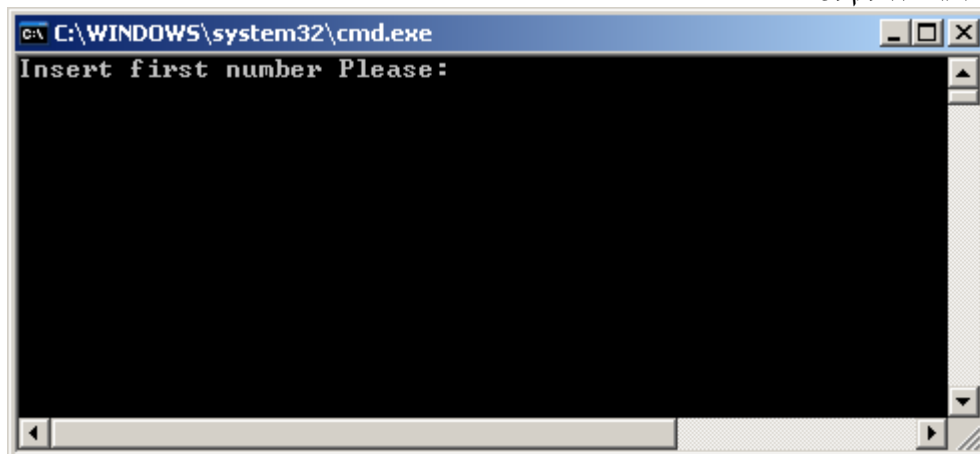
```
int num1;  
int num2;
```

להלן התוכנית השלמה כפי שהיא צריכה להיות:

```
using System;  
  
class HelloWorld  
{  
    static void Main()  
    {  
        int num1;  
        int num2;  
  
        Console.WriteLine("Insert first number Please: ");  
        num1 = int.Parse(Console.ReadLine());  
        Console.WriteLine("Insert second number Please: ");  
        num2 = int.Parse(Console.ReadLine());  
  
        Console.WriteLine("The two numbers are: {0} {1}", num1, num2);  
    }  
}
```

הדר שוב את התוכנית והרץ אותה

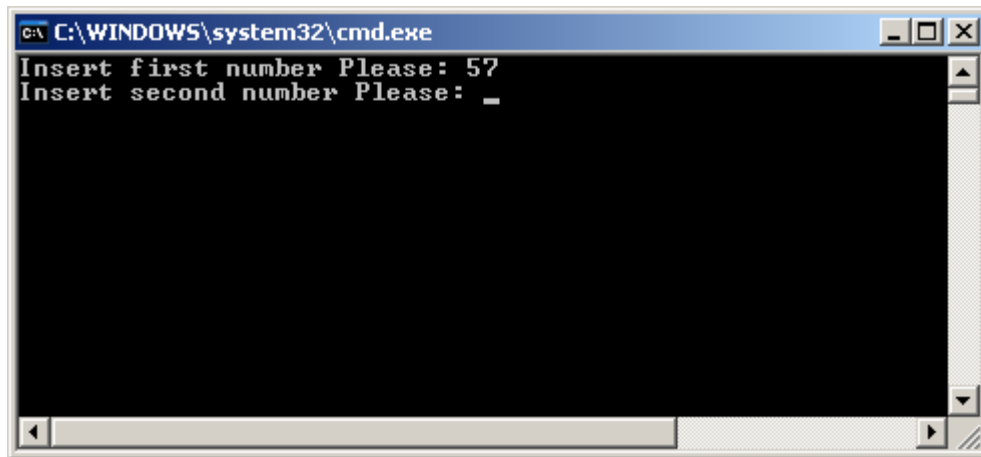
כעת התוכנית רצה, אך לא הסתיימה מייד כפי שקרה עד עכשיו, וזאת כיוון שהתוכנית נעצרה באמצע ריצתה וממתינה לקלט:



על מנת להזין קלט לתוכנית, הקש את המספר 57 ואחריו הקש על Enter

**שימו** ♥: אם התוכנית ממתינה לקבל מספר שלם, אך אנו נקליד קלט שאינו מספר שלם, יגרום הדבר לשגיאת ריצה. ייפתח מסך בו תופיע השאלה: האם ברצונכם לנפות את השגיאות באמצעות כלי לניפוי שגיאות? לחצו על "No". אז תופיע הודעת השגיאה על גבי מסך התוכנית, והתוכנית תפסיק את ריצתה.

כעת שוב נעצרה פעולת התוכנית, והיא ממתינה לקלט נוסף:



הקש מספר שלם נוסף ואחריו הקש על Enter

הערכים אותם הקשנו נקלטו על ידי התוכנית ונשמרו במשתנים המתאימים. בדוגמא זו המשתנה הראשון, num1 קיבל את הערך 57, ואילו המשתנה השני num2 קיבל את הערך השני אותו בחרת. מיד לאחר קליטת שני הערכים המשיכה התוכנית בפעולתה והדפיסה על המסך את משפט הפלט המציג את ערכי שני המשתנים ששמרה.

הרץ את התוכנית פעם נוספת, הזן קלטים שונים מאלה שהזנו בהרצה הראשונה וודא כי התוכנית משיגה את מטרתה ומדפיסה את הערכים הנכונים של המשתנים.

#### משימה

כתוב תוכנית שתקבל כקלט שלושה מספרים שלמים, והפלט שלה יהיה המספר האמצעי שהוקלד.  
למשל, עבור הקלט: 9 5 3 יודפס: " המספר האמצעי שהוקלד הוא: 5 "

#### חישוב עם משתנים

פתח פרויקט חדש כפי שעשינו בתחילה, ומחק את שורות הקוד המיותרות עבור צרכינו.

כעת, עלינו לכתוב תוכנית המקבלת שני מספרים שלמים כקלט, (כמו בתרגיל הקודם) ומדפיסה כפלט את ההודעה: "סכום המספרים שנקלטו הם: " ואת סכומם.

גם כאן, כמו בבעיה הקודמת, עלינו לקלוט שני מספרים שלמים ולכן נכתוב את שורות הקוד הבאות:

```
Console.WriteLine("Insert first number Please: ");
num1 = int.Parse(Console.ReadLine());
Console.WriteLine("Insert second number Please: ");
num2 = int.Parse(Console.ReadLine());
```

כעת, ברצוננו לחשב את סכום המספרים, לשם כך עלינו להיעזר במשתנה נוסף לצורך שמירת הסכום. נוכל לקרוא לו sum (סכום = sum). משתנה זה אינו משתנה קלט, כי אין אנו קולטים את ערכו אלא הוא מחושב במהלך ריצת התוכנית.

לאחר שיש בידינו את שני הערכים של num1, num2 נרצה לשים בתוך המשתנה sum את סכומם. לשם כך נזדקק לפעולת השמה. פעולת ההשמה מתבצעת על ידי הסימן =. כך למשל אם נרצה לשים בתוך המשתנה x את הערך 7, נכתוב:  $x=7$ .

הוסף את ההוראה המבצעת הוראת השמה:

```
sum = num1 + num2;
```

הוסף לתוכנית הוראת פלט שתדפיס את ערכו של המשתנה sum.

הגדרה: בפעולת השמה, המחשב מבצע חישוב כלשהו וְשֵׁם (שומר) את התוצאה בתוך משתנה.

פעולת ההשמה מיושמת ב-C# כך:  $y = \text{expression};$

משפט זה מורה על השמת ערך הביטוי expression בתוך המשתנה y.

הביטוי expression יכול להיות ביטוי פשוט (כמו למשל מספר או שם של משתנה) או ביטוי המורכב מפעולות חשבוניות שונות, כמו למשל סכום.

הנה התוכנית השלמה כפי שהיא צריכה להיות:

```
using System;

class CalculateSum
{
    static void Main()
    {
        int num1;
        int num2;
        int sum;

        Console.Write("Insert first number Please: ");
        num1 = int.Parse(Console.ReadLine());
        Console.Write("Insert second number Please: ");
        num2 = int.Parse(Console.ReadLine());

        sum = num1 + num2;
        Console.WriteLine("The sum is: {0}", sum);
    }
}
```

הדר את התוכנית והרץ אותה

הרץ את התוכנית מספר פעמים נוספות, הזן קלטים שונים וודא כי התוכנית משיגה את מטרתה ומדפיסה את הסכום המבוקש.

#### משימה

כתוב תוכנית שתקבל כקלט שלושה מספרים שלמים, והפלט שלה יהיה מכפלת שלושת המספרים שהוקלדו.

למשל, עבור הקלט: 2 5 3 יודפס: " המכפלה היא: 30 "

הדרכה: לשם הכפלה בין מספרים השתמש בסימן \*

## בנוסף למתקדמים!

ניתן לשנות את צבע הטקסט וצבע הרקע של טקסט הכיתוב:

? `Console.ForegroundColor = ConsoleColor.Red;` מה תעשה ההוראה

---

---

? `Console.BackgroundColor = ConsoleColor.Green;` מה תעשה ההוראה

---

---

? `Console.ResetColor();` מה תעשה ההוראה

---

---

הדרכה: שנה את גוף התוכנית להיות

```
Console.WriteLine("some output");
Console.ForegroundColor = ConsoleColor.Yellow;
Console.WriteLine("some other output");
Console.ForegroundColor = ConsoleColor.DarkGreen;
Console.WriteLine("some different output");
Console.BackgroundColor = ConsoleColor.Red;
Console.WriteLine("something else");
Console.ResetColor();
```

? `Console.Beep();` מה תעשה ההוראה

---

---

# בהצלחה ובהנאה!