



Visual Studio .Net 2013

חוברת זו נועדה ללמד אותך, הקורא, כיצד להשתמש בסביבת העבודה של ה-Microsoft Visual Studio .NET. סביבה זו תשתמש אותך לכל אורך הקורס לשם כתיבת תר"צים ומדרגות ולכן חשוב כי תכיר סביבה זו היטב ולעומק. סביבת ה-NET. עשירה מאוד, דינאמית ובעלת יכולות נרחבות. הסביבה מאוד נוחה ואסתטית, אך נדרשת מיומנות די מעמיקה כדי לשלוט בסביבה הזו ביד רמה, זריזה ומקצועית.

חוברת זו תשאר איתך לאורך הקורס ותשמש כמעין ספר עיון שיזכיר כיצד יש להשתמש בסביבת העבודה. מעקב אחר הצעדים הנלמדים בחוברת יבטיח הכרה טובה של הסביבה, דבר שהוא חיוני להצלחה בקורס.

רכישת מיומנות בסביבת העבודה תאפשר לך להתרכז בתוכנית אותה עליך לכתוב, ולא בעניינים חיצוניים כגון כתיבת טקסט, ניהול פרויקטים וכו'.

עליך לעבור על שני השיעורים הראשונים (פרק 1 ופרק 2) היום. הודעה על מתי לקרוא את שאר הפרקים תקבל במועד מאוחר יותר. אין צורך לקרוא את הפרקים הבאים לפני שקיבלתם את ההודעה על כך, עם כי הדבר מותר.

סגל המדריכים מאחל לך הצלחה רבה בתרגול החוברת ובקורס בכלל.

החוברת כתובה בלשון זכר, אך מכוונת לשני המינים.

תוכן עניינים

| | |
|----|----------------------------------|
| b | מבוא |
| c | תוכן עניינים |
| e | מטרות החוברת |
| e | למי מיועדת החוברת |
| 1 | איך ללמוד באמצעות החוברת |
| 1 | מוסכמות סימון בחוברת |
| 2 | הקדמה |
| 3 | הפעלת ה- Visual Studio |
| 4 | עורך הטקסט |
| 5 | עריכת תוכנית |
| 7 | עריכת טקסט |
| 7 | Insert/Overwrite |
| 7 | הדגשת טקסט - Syntax Highlighting |
| 8 | תנועה במסמך |
| 8 | עימוד טקסט |
| 9 | העתקה והעברה |
| 9 | מחיקה |
| 10 | ביטול |
| 10 | שימוש בעזרה |
| 11 | חיפוש והחלפה |
| 13 | סיכום |
| 13 | שאלות למחשבה |
| 14 | סביבת פיתוח |
| 15 | מהו פרויקט |
| 16 | טיפול בפרויקט |
| 16 | יצירת פרויקט חדש |
| 22 | הרצה |
| 23 | מהו Solution? |
| 23 | פתיחת/סגירת פתרון (Solution) |
| 24 | יצירת קבצי תוכניות |
| 26 | הידור, קישור, ריצה ומה שבאמצע |
| 26 | הידור וקישור |
| 28 | הרצה |
| 28 | מה שבאמצע |
| 29 | מתי נשתמש בכל פעולה? |
| 30 | סיכום |

31

32

32

32

32

35

36

36

36

36

36

37

38

39

39

39

41

42

43

44

44

45

46

47

47

48

49

עבודה יעילה

מקלדת

סימון

תנועה מהירה

קיצורי דרך

עכבר

טיפים כללים

סימניות

הגדלת המסך

חיפוש והחלפה

שמירה אוטומטית

השלמה אוטומטית

Debug בסיסי

תיקון שגיאות בתכנית

היכן נמצא הדיבאגר

שימוש בדיבאגר

נקודות עצירה (BreakPoints)

מעקב

טיפים

כיצד לבנות תהליך דיבאג

סיכום

Debug מתקדם

דיבוג שגרות

מפת התייחסויות

קריאות

נספחים

Help

מיטרת החוברת

למי מיועדת החוברת

החוברת מיועדת לחניכי קורס תכנות בסיסי בתחילת הקורס ובהמשכו. נדרשת הכרה של סביבת Windows לפני השיעור הראשון.

חוברת זו נועדה:

ללמוד ולתרגל עבודה בסביבת ה-Visual Studio .NET.

לאחר קריאת החוברת תוכל:

לערוך תוכניות באמצעות ה-Visual Studio .NET.

להכין תוכניות לריצה בסביבות C++.

לנפות שגיאות בתוכניות שכתבת באמצעות כלי ה-Debugger.

איך ללמוד באמצעות החוברת

חוברת זו מיועדת ללימוד עצמי. היא מכילה את כל הדרוש לך לצורך לימוד החומר.

החוברת מורכבת משיעורים. בדרך-כלל, לימוד השיעורים המוקדמים דרוש לצורך הבנת השיעורים הבאים. לכן, מומלץ שתתקדם בחוברת עפ"י סדר השיעורים, ולא תעבור לשיעור הבא לפני שוידאת שהבנת את כל החומר בשיעור הנוכחי. על-מנת להקל עליך, מופיעות בסוף כל שיעור שאלות חזרה על החומר הכלול בו.

בגוף החוברת משולבים תרגולים ושאלות. אל תתעצל ופתור את כולם. רק ביצוע של כל התרגולים יביא אותך לשליטה מספקת בחומר.

כל שיעור מחולק לפרקים. אין צורך לקרוא את כל השיעור בפעם אחת, ניתן לקרוא פרק אחד בכל פעם, אך מומלץ לא להפסיק לקרוא באמצע פרק. אם קראת פרק ולא הבנת, נסה שוב. אם החומר עדיין לא מובן, פנה למדריך התורן.

מוסכמות סימון בחוברת

לפני שתתחיל בלימוד, חשוב שתכיר את מוסכמות הסימון בהן נעשה שימוש בחוברת זו.

הערות ותוספות נכתבות בשוליים. דגשים, אזהרות וסיכומים חשובים מופיעים בשוליים בצורה מודגשת.

סימן הספר מפנה אותך למידע נוסף בספרות עזר.

מונחים חשובים מופיעים בצורה **נטויה**.

פעולות שעליך לבצע על המחשב מופיעות במסגרת, לצד סימן זה:

תווים אותם עליך להקליד מופיעים **בהדגשה**.

תרשימי תחביר (Syntax):

- טקסט המופיע באותיות **נטויות** יש להחליף בערך אותו הוא מייצג (לדוגמה, כתוב את שמך במקום **שםך**).
- טקסט בין סוגריים מרובעים הוא אופציונאלי. [טקסט זה לא חובה להקיש]
- רשימת ערכים המופיעה בין סוגריים מסולסלים מציינת שיש לבחור רק **אחד** מבין הערכים האלה. אני רוצה לאכול { תפוח, אגס }
- ערך ברירת מחדל מודגש בקו תחתון. { תפוח, **אגס** }
- כאשר מופיע תיאור של שני מקשים, וביניהם הסימן +, זה הסבר לכך שיש ללחוץ על שני המקשים בו-זמנית. לדוגמה, כאשר מוסבר כי יש להקיש **Ctrl+F5** יש להקיש על מקש ה-**Ctrl** ובו זמנית על מקש ה-**F5**.



הקדמה

אם נחזור על הנלמד בכיתה, הכנת תוכנית מוגמרת מורכבת ממספר שלבים :
השלב הראשון הוא כתיבת עיצוב התוכנית ותכנון האלגוריתם. דבר זה נעשה
לעיתים ע"י התוכניתן עצמו, ולעיתים ע"י מעצב.

השלב השני הוא תרגום האלגוריתם לתוכנית בשפת מחשב כלשהי. פעולה זו
מתבצעת על המחשב, ע"י התוכניתן.

השלב השלישי הוא הידור וקישור. בשלב זה נלקחת התוכנית שנכתבה והיא
מתורגמת לשפה אותה מבין המחשב.

השלב הרביעי הוא הדיבאג או הטסטינג. אותו תהליך של מציאת שגיאות
בתוכנית ותיקון.

השלב האחרון הוא הפצת התוכנית למשתמשים המיועדים.

ה-Visual Studio .NET הוא כלי המאחד בתוכו את השלב השני, השלישי והרביעי,
ואף מוסיף קריצה (בעזרת כלים נוספים שתלמדו בהמשך) גם לשלב החמישי.

צורת מחשבה זו, שמאפשרת לתוכניתן לעשות את כל השלבים מאותה תוכנה,
מקלה מאוד על חיינו כתוכניתנים. הדבר נכון כפליים ב-Visual Studio .NET. כלי
זה, בנוסף לכך שהוא מאגד תחת קורת גג אחת כלים רבים, מציע גם נוחות
מקסימאלית לכל תוכניתן, מתחיל ומתקדם.

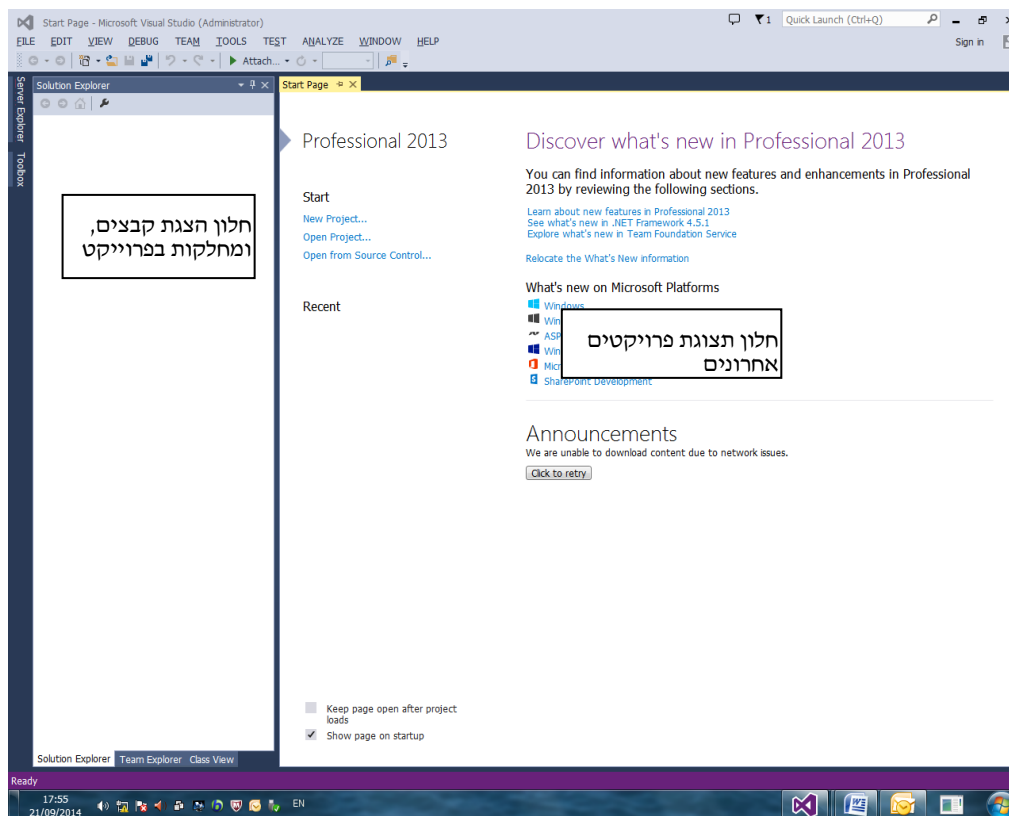
הפעלת ה- Visual Studio

ובכן לאחר שאתה יודע מהו ה- Visual Studio.NET השאלה המתבקשת היא כיצד להפעיל אותו.

הרץ את התוכנה Microsoft Visual Studio .NET הנמצאת בכפתור ה"התחל", תחת התפריט "תוכניות", ותת-התפריט "Microsoft Visual Studio .NET".



יוצג מסך הדומה למסך הבא :



הסבר על סביבת העבודה ושימוש בה יבוא בשיעורים הקרובים. תרגל היטב ובצמוד לחוברת, והכרה מקיפה של סביבת העבודה שלך - מובטחת.

עורך הטקסט

עריכת תוכנית

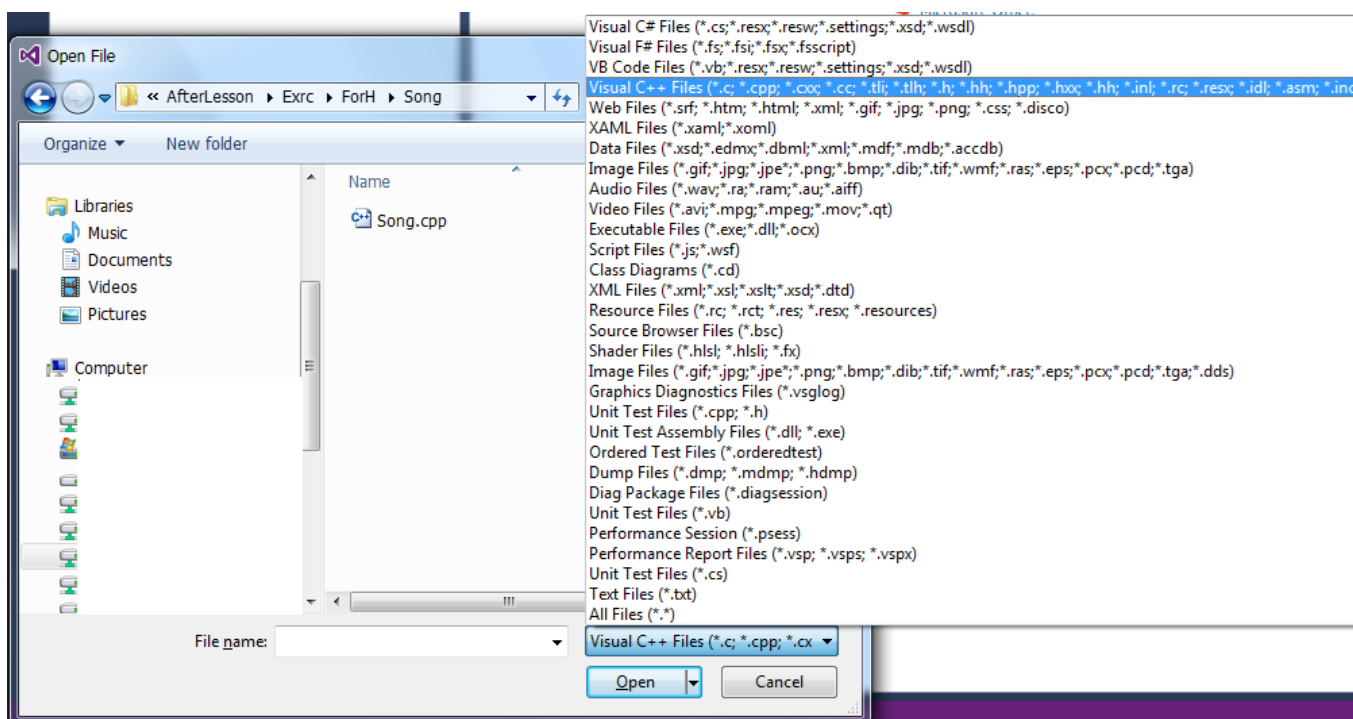
כפי שכבר אמרנו בהקדמה, השלב הראשון אותו עושה התוכניתן הוא שלב הכתיבה. ולכן השיעור הראשון מוקדש, בצורה לוגית מאוד, לעריכת תוכנית. חשוב לציין שיכולות העורך של ה-Visual Studio .NET הן מתקדמות מאוד. כל-כך מתקדמות עד שנוח מאוד גם לערוך קבצים רגילים, קבצים שאינם תוכנית בשפת מחשב, בעזרת ה-Visual Studio .NET.

ראשית, פתח את התוכנית שהוכנה לך מבעוד מועד. בעזרת תוכנית זו נדגים את יכולות ה-Visual Studio .NET. אל תדאג אם תעשה שגיאות, אינך יכול לעשות שום נזק במהלך התרגול.

בחר באופציה "File..." הנמצאת תחת תת-התפריט "Open" הנמצא תחת תפריט ה-"File". או לחץ על צירוף המקשים Ctrl+O.



תופיע לפניך תיבת השיחה הבאה :

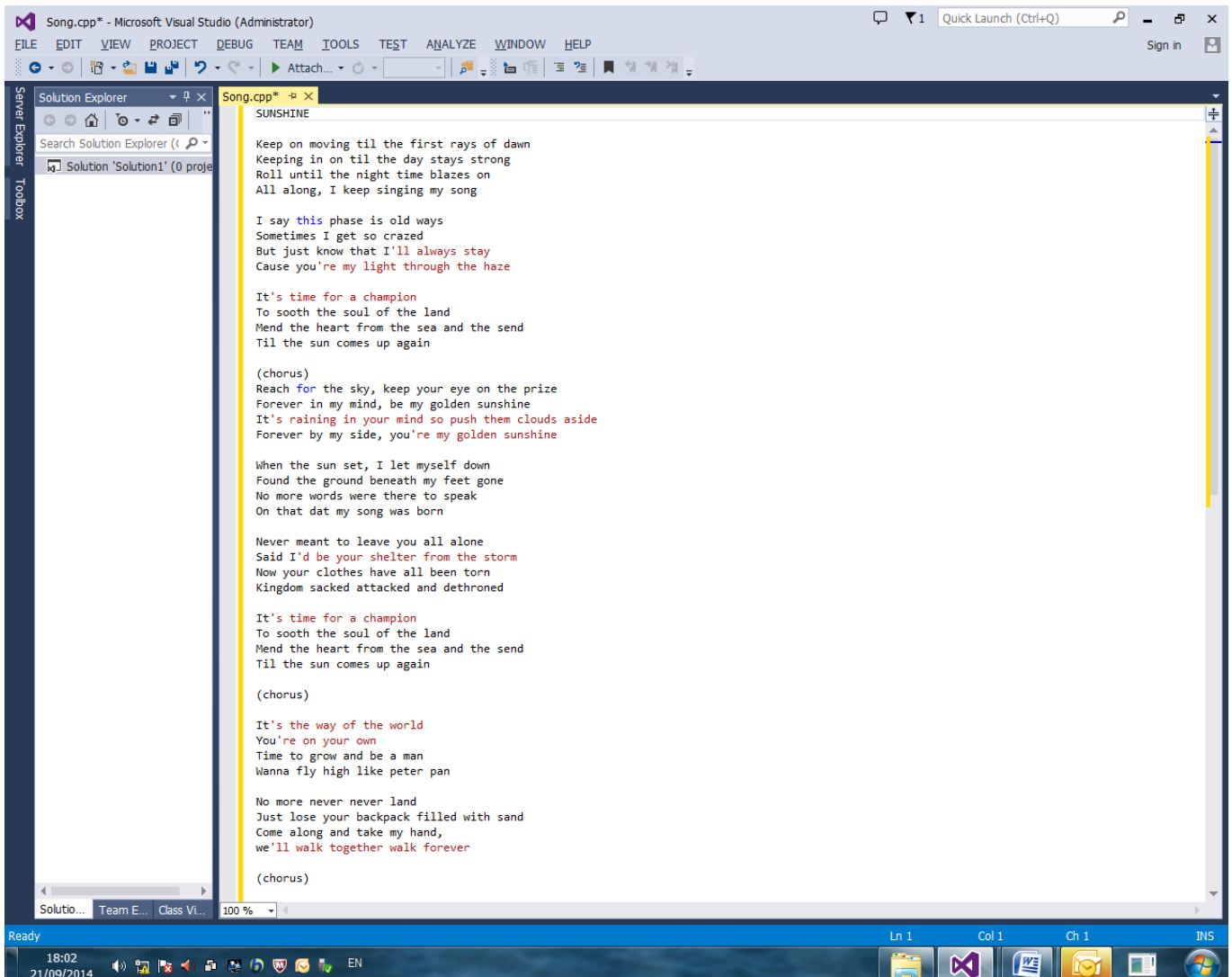


כדי להקל על המשימה, ביכולתך לבחור בסוגי הקבצים שיוצגו בתיבה : C++ Files לצורך העניין.

בחר בקובץ Song.cpp (נמצא בהערה בקובץ)



אם פעלת כשורה אתה אמור לראות את המסך הבא.



(ישנה אפשרות שהטקסט ייפרס על מעבר לעמוד אחד)

עריכת טקסט

עריכת טקסט היא דבר פשוט ואינטואיטיבי ביותר. ההקלדה נעשית ע"י לחיצה על מקשי המקלדת המתאימים (מן הסתם). לשם הכנסת שורה ריקה יש ללחוץ על מקש ה-ENTER בסוף השורה.

לשם הזזת הסמן במסך ניתן להשתמש בעכבר או במקשי החצים.

הכנס שורה ריקה בין שורה מספר 19 לשורה מספר 20.



חכה!!! לפני שאתה מתחיל לספור שורות מראש המסך, יש ב- Visual Studio .NET אופציה המאפשרת לראות את השורה והטור בה נמצא הסמן. הסט עיניך לצידו הימני תחתון של המסך. הטקסט Ln מציין את השורה שבה הסמן נמצא.

האם אתה מוצא את מקש ה-DEL במקלדת? אם לא, נסה לחפשו מעל חיצו התנועה שבמקלדת.

כדי להפריד שורה אחת לשתיים יש ללחוץ ENTER במקום המתאים בשורה. שתי השורות ייפרדו. כדי לחבר שתי שורות לשורה אחת, יש ללחוץ על מקש ה-DEL בסוף השורה.

פצל את שורה 29 לאחר המילה השנייה.
חבר את השורות שפיצלת לשורה אחת.



Insert/Overwrite

האם אתה מוצא את מקש ה-Insert במקלדת? הוא נמצא מעל מקש ה-Del.

בתור ברירת-מחדל כאשר כותבים תווים במסך הסורס (מהמילה Source כלומר קוד התוכנית המקורי אותו כותב התוכניתן) ב-Visual Studio .NET, שאר התווים שבשורה נדחפים. לעיתים אנו מעוניינים לכתוב על התווים הקיימים, ולא להוסיף להם. לשם כך יש מקש במקלדת בשם Insert שמשנה את המצב בין דחיפה להחלפה.

החלף בשורה 5 את המילה "time" במילה "star".
הוסף את המילה "while" לאחר המילה הרביעית בשורה 22.



הדגשת טקסט - Syntax Highlighting

אם שמת לב המילים שהקלדת בתרגיל הקודם נצבעו בכחול, בשונה משאר הטקסט השחור שבקובץ. הסיבה לכך נעוצה בעובדה שעורך התמליל חושב שאנו כותבים תוכנית ב-C++.

על מנת להקל על קריאות התוכנית יודע העורך לצבוע מילים שמוגדרות כפקודות בשפה בצבע מיוחד.

הדבר מסייע במיוחד לשם מציאת שגיאות בזמן הקלדת פקודות. כאשר שמים לב שהקשנו פקודה והיא לא נצבעה בכחול, ניתן להבין כי שם הפקודה לא הוקש בצורה נכונה.

תנועה במסמך

במקלדת ה-PC ישנם מספר מקשים במקלדת שמאפשרים את הזזת הסמן בצורה נוחה יותר מאשר הזזתו עם החצים.



המקלדת, כמו כל דבר בצבא, מחולקת ל-3 חלקים. החלק השמאלי מכיל את כל האותיות ובו משתמשים לשם הקלדה.

החלק הימני מכיל מספרים לשם הקשה מהירה ונוחה של מספרים.

החלק האמצעי מכיל את החצים, ועוד קבוצה של 6 מקשים. מתוכם הכרתם עד עכשיו שניים: Delete-Insert.

| שימוש | מקש |
|-----------------------------|-----------|
| מזיז את הסמן מסך אחד למעלה. | Page Up |
| מזיז את הסמן מסך אחד למטה. | Page Down |
| מזיז את הסמן לסוף השורה. | End |
| מזיז את הסמן לתחילת השורה. | Home |

אלה פעולות עריכה בסיסיות הבה נתקדם קצת.

עימוד טקסט

את השימוש לעימוד טקסט תכיר מאוחר יותר, אך בינתיים תרגל את אופן הביצוע.

כעקרון בעת כתיבת תוכנית C++ יודע העורך לעמד לבד את התוכנית.

נסה את הדוגמה הבאה:

הקלד, בסוף השיר, את קטע הקוד הבא:

```
if (a>5)
{
    a=b;
}
```

יש להקליד את הקוד ברצף, ללא שימוש במקשי החיצים או הרווח, תוך שימוש במקש ה-Enter בלבד לשם קפיצות שורות.



ובכן, כרגע ראית מהו העימוד האוטומטי של העורך. בהמשך הקורס הדבר יסייע לך רבות.

אם אתה רוצה לשנות את העימוד האוטומטי של העורך ניתן לעשות זאת באמצעות מקש ה-TAB. מקש זה גורם הכנסת מספר רווחים קבוע, ולכן נוח מאוד להשתמש בו לשם עימוד תוכנית.

עמד את שורות הפזמון (Chorus) פנימה.
עימוד בתחילת כל שורה בפזמון והקש על מקש ה-TAB.



מקש ה-BackSpace (מצוי במקלדת מעל מקש ה-ENTER) ומסומן ע"י התו הבא: \leftarrow , מאפשר לבטל את השפעת ה-TAB בלחיצה אחת.

החזר את עימוד הפזמון לאחור.
עימוד על תחילת המילה הראשונה בכל שורה בפזמון ולחץ על מקש ה-
.BackSpace



העתקה והעברה

לעיתים אנו כותבים קטע תוכנית מספר פעמים, או נאלצים להעתיק קוד דומה מאוד ממקום למקום. במקום לכתוב את הטקסט מחדש ניתן להשתמש בפעולות העברה והעתקה.

סמן את הפזמון באמצעות העכבר.
סימון טקסט נעשה ע"י גרירת סמן העכבר על הטקסט אותו מבקשים לבחור.



את פעולות העברה והעתקה ניתן לעשות גם על טקסט לא מסומן. אם נעמוד עם הסמן בשורה (לא משנה היכן), ונבצע פעולת העברה או העתקה, הפעולה תבוצע על השורה הנוכחית בשלמותה, והעורך ירד שורה באופן אוטומטי. העברה מוחקת את הטקסט מהקוד ומעבירה אותו לשטח עזר בזיכרון. העתקה עושה אותו דבר, אלא שהיא משאירה את הטקסט בתוכנית.

שטח העזר הזה נקרא ע"י Windows בשם ClipBoard. בעזרתו ניתן להעביר טקסט וגרפיקה בין מקומות שונים בתוכנית ואפילו בין תוכניות שונות.

כדי להעתיק את המידע מה-ClipBoard לתוכנית מבצעים פעולה הנקראת הדבקה.

פעולת העברה מתבצעת ע"י לחיצה על המקשים Shift+Delete.

פעולת העתקה מתבצעת ע"י לחיצה על המקשים Ctrl+Insert.

פעולת הדבקה מתבצעת ע"י לחיצה על המקשים Shift+Insert. יש לשים לב שהסמן יהיה במקום בו אנו מעוניינים להכניס את הטקסט, או את השורה.

לאחר פעולת ההדבקה הטקסט עדיין נשאר ב-ClipBoard, כך שניתן לשכפל טקסט מספר רב של פעמים.

שים לב שכל הפעולות שהוזכרו למעלה יכולות להיות מיושמות גם באמצעות לחיצה ימנית על כפתור העכבר כאשר הסמן נמצא מעל הטקסט המצוין וגם באמצעות התפריט עריכה (Edit). בחר אתה כיצד נוך לך להעתיק ולהעביר טקסט.

וכעת לתרגול.

העבר את הפזמון ל-ClipBoard. (פעולת העברה).
הדבק אותו מחדש בכל מקום בו כתוב Chorus.
העתק את השורה הראשונה של השיר ל-ClipBoard. שים לב, אין צורך לסמן את כל השורה אלא להעביר את הסמן לשורה ולבצע את הפעולה. (פעולת העתקה).
הדבק אותה גם בשורה בה כתוב THE END לפני המילה END. לחצת Enter? לא היית צריך.



מחיקה

מקש אחד משמש בעורך לכל פעולות המחיקה. מקש זה הוא מקש ה-Delete. בנוסף לכך שהוא יודע למחוק תווים בודדים מהשורה, הוא גם יודע, במקרה ויש טקסט מסומן, למחוק את כל הטקסט, בלי להעבירו ל-ClipBoard.

בחר את הטקסט הכתוב בסוף המסמך (אותו טקסט שכתבת בתרגול "עימוד טקסט") ומחק אותו לחלוטין מהעורך.
כעת עמוד בסוף המסמך ובצע פעולת הדבקה. מה הודבק?



ביטול

לעיתים אנו עושים שגיאות. כולנו בני אדם. החוכמה היא להתגבר על שגיאות. נניח שמחקנו קטע קוד עבה למדי בתוכנית שלנו. כיצד נשחזר את הנעשה?

הפקודה Undo (בטל) גורמת לביטול הפעולה האחרונה שנעשתה. ניתן להשתמש בה ע"י לחיצה על מקש ה-Ctrl במקלדת ותוך כדי לחיצה עליו לוחצים גם על המקש Z (Ctrl+Z).

ניתן לבטל יותר מפעולה אחת.

במידה והתחרטנו אחרי ביטול פקודה, ניתן להשתמש בפקודה Redo (בצע שוב) על מנת לשחזר את הפקודה שביטלנו. השימוש בפקודת ה-Redo נעשה ע"י לחיצה על מקש ה-Ctrl במקלדת תוך כדי לחיצה על המקש Y (Ctrl+Y).

בטל את הפעולה האחרונה שעשית.

בטל גם את השתיים שעשית לפני כן.

שחזר את הפעולה האחרונה שביטלת.



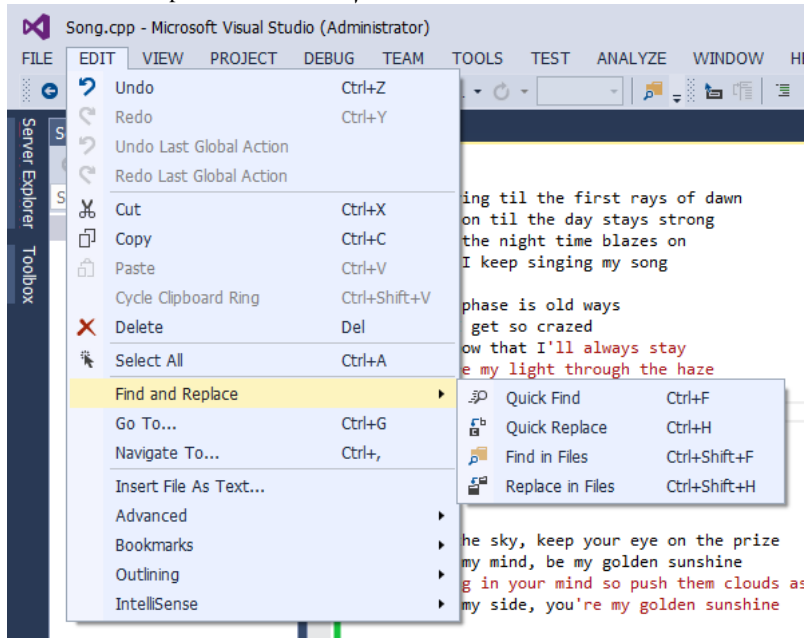
שימוש בעזרה

בנוסף לקומפיילר, עורך הטקסט ושאר המטעמים שקיימים ב-Visual Studio .NET יש גם מערכת עזרה מתקדמת שנקראת MSDN. אנו נרחיב עליה את הדיבור בנספחים, אבל בינתיים רק שתדע, כדי להכנס ל-MSDN יש להכנס לתיקיית icons שעל הdesktop (או בכונן ה-w) וללחוץ על הקישור ל-MSDN.

חיפוש והחלפה

נניח כי יש לך תוכנית בת 1000 שורות ואתה לא זוכר איפה בדיוק כתבת קטע קוד מסוים. בעיה. והנה הפתרון.

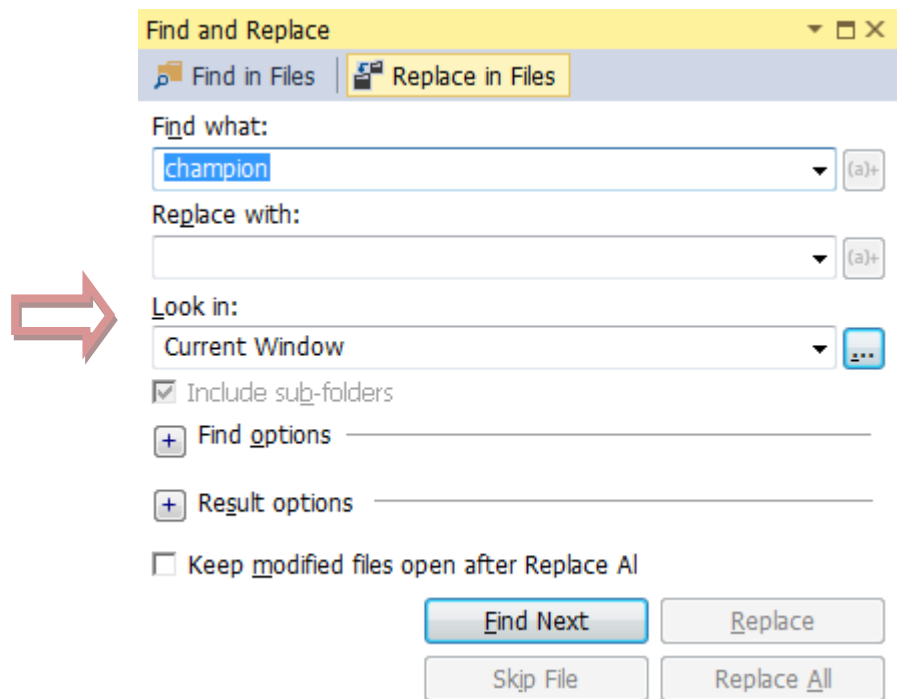
תחת תפריט ה-Edit מצויה הישועה. הפקודות Find ו-Replace יפתרו את הבעיה.



הפקודה Find מוצאת טקסט מסוים בתוכנית. פקודת ה-Replace מאפשרת החלפת טקסט מסוים בתוכנית בטקסט אחר, ניתן לגשת פקודות אלו גם בעזרת הקיצורים המופיעים מימין לפקודה בתפריט המצולם למעלה.

מכיוון שהשימוש בפקודות אלה דומה אנו נתמקד בפקודת ה-Replace in Files.

פקודת ה-Replace in Files גורמת להופעתה של תיבת השיחה הבאה:




• שימו לב שהסעיף Look in צריך להיות על ה־Current Window או על All Open Documents.

כיוון שאנחנו עדיין לא עובדים עם Solution אלא רק עם קובץ, במידה והחפוש יהיה על Entire Solution הוא לא ימצא.

יש להכניס את הטקסט המקורי, את הטקסט בו רוצים להחליף ואיפה אנחנו רוצים לחפש ולשנות את המילים. בנוסף, אם נלחץ על ה'+' נוכל לבחור אופציות מסוימות - האם להתייחס לאות גדולה כאות קטנה, האם להתייחס לטקסט כמילה, וכו'.

הכפתורים מאפשרים קביעה מדויקת יותר של מה להחליף. כפתור ה-Find Next מוצא את המופע הבא של המילה. לחיצה על כפתור ה-Replace תחליף אותה במילה המבוקשת. כפתור ה-Replace All עובר, ללא מגע מצידנו, על כל האזור המבוקש ומחליף את כל המופעים של המילה במילה החדשה.

פתח את תיבת ה-Replace. 


בשדה ה-Find What כתוב את הטקסט it (או שתפתח את תיבת ה-Replace כאשר אתה עומד עם הסמן על המילה it שכתובה בעורך).

בשדה ה-Replace with כתוב את הטקסט eat.

וודא כי שלושת תיבות האופציות המתקדמות לא מסומנות (Match whole word , Use , Match case , only).

לחץ על כפתור ה-Find Next.

התבונן במסמך. האם אתה רואה היכן נמצא הסמן?

לחץ על כפתור ה-Replace. וודא כי אכן השתנה הטקסט במסמך. 


לחץ על כפתור ה-Find Next.


לחץ עליו שוב. מה קרה?

לחץ על כפתור ה-Replace All.

עכשיו נסה למצוא את הטקסט it במסמך בעזרת תיבת ה-Find. כמה מחרוזות מצאת?

ועכשיו לבד.

החלף את כל המופעים של הטקסט I בטקסט IBM MF. (השתמש ב-Replace All). 

המחשב אינו חכם במיוחד. הוא יחליף את כל המופעים של I שהוא מוצא. כולל כאלה באמצע מלה. יש להשתמש באופציה Replace All בזהירות. 

שים לב

סיכום

מה למדנו עד עכשיו? קודם כל איך לפתוח קובץ ב-Visual Studio .NET. אין צורך לפרט את חשיבות הפעולה הזו.

אחר כך עברנו קצת על פעולות עריכה בסיסיות - הזזת הסמן, מחיקה והכנסת שורות.

ואז, אז התקדמנו קצת ולמדנו על עימוד טקסט, העתקה והעברה, וגם על חיפוש והחלפה אוטומטים. אם אינך זוכר משהו מבין הנושאים המוזכרים, חזור על הפרק שוב.

שאלות למחשבה

האם נוח לך בינתיים עם הכלי?

האם אתה מרגיש את הכוח המתחבא בקצות האצבעות שלך?

אם לא, קרא במועד מאוחר יותר את שיעור 4 - "עבודה יעילה" אשר נמצא בעמוד 31.

סביבת פיתוח

מהו פרויקט

לפני שנתחיל לעבוד בצורה מסודרת עם ה-Visual Studio .NET יש להכיר מושגים הנקראים פרויקט ופתרון.

ההנחה הבסיסית היא כי כל פרויקט מורכב ממספר תוכניות המרכיבות, ביחד, את הפרויקט. פרויקט ה"חץ" לדוגמה מורכב ממספר גדול של קבצים שכותבים מספר גדול של אנשים. האיחוד של כל הקבצים הללו הוא הפרויקט. פרויקט יכול להיות מורכב ממספר גדול מאוד של קבצים, או תתי פרויקטים אחרים. למעשה פרויקט יכול להיות מורכב מקבצים במספר שפות - C, ++C, אסמבלר ועוד.

גם ה-Visual Studio .NET יודע לעבוד עם פרויקטים. פרויקט ב-Visual Studio .NET מכיל למעשה מידע על הקבצים המרכיבים את הפרויקט וכן כיצד לבנות את הפרויקט.

ב-Visual Studio .NET המציאו את המושג פתרון (Solution). פתרון מורכב ממספר פרויקטי משנה, היכולים להיכתב במספר שפות שונות (C++, VB, C#, Java ועוד).

לעת עתה מספיק להבין כי פרויקט הוא בסה"כ רשימה של כל הקבצים המשתתפים בתוכנית.

חשוב להדגיש כי הפרויקט אינו מכיל את הקבצים עצמם, אלא רק מידע על המיקום שלהם בדיסק הקשיח. אם נמחק קובץ מרשימת קבצי הפרויקט (עבור פרויקטים משפת ++C), הקבצים לא ימחקו ישירות מהדיסק, אלא רק מהרשימה.

הפרויקטים הראשונים שתעבדו עליהם בקורס יהיו קטנים. הווה אומר פרויקטים המכילים רק קובץ אחד בהם (ב-++C או באסמבלר). מדוע ליצור פרויקט לקובץ אחד (הרי משמעות הפרויקט היא לאגד מספר קבצים השותפים ביצירת תוכנית)? ובכן Visual Studio .NET מאפשר לנו להדר ולקשר אך ורק קבצים אשר שותפים לפרויקט מסוים.

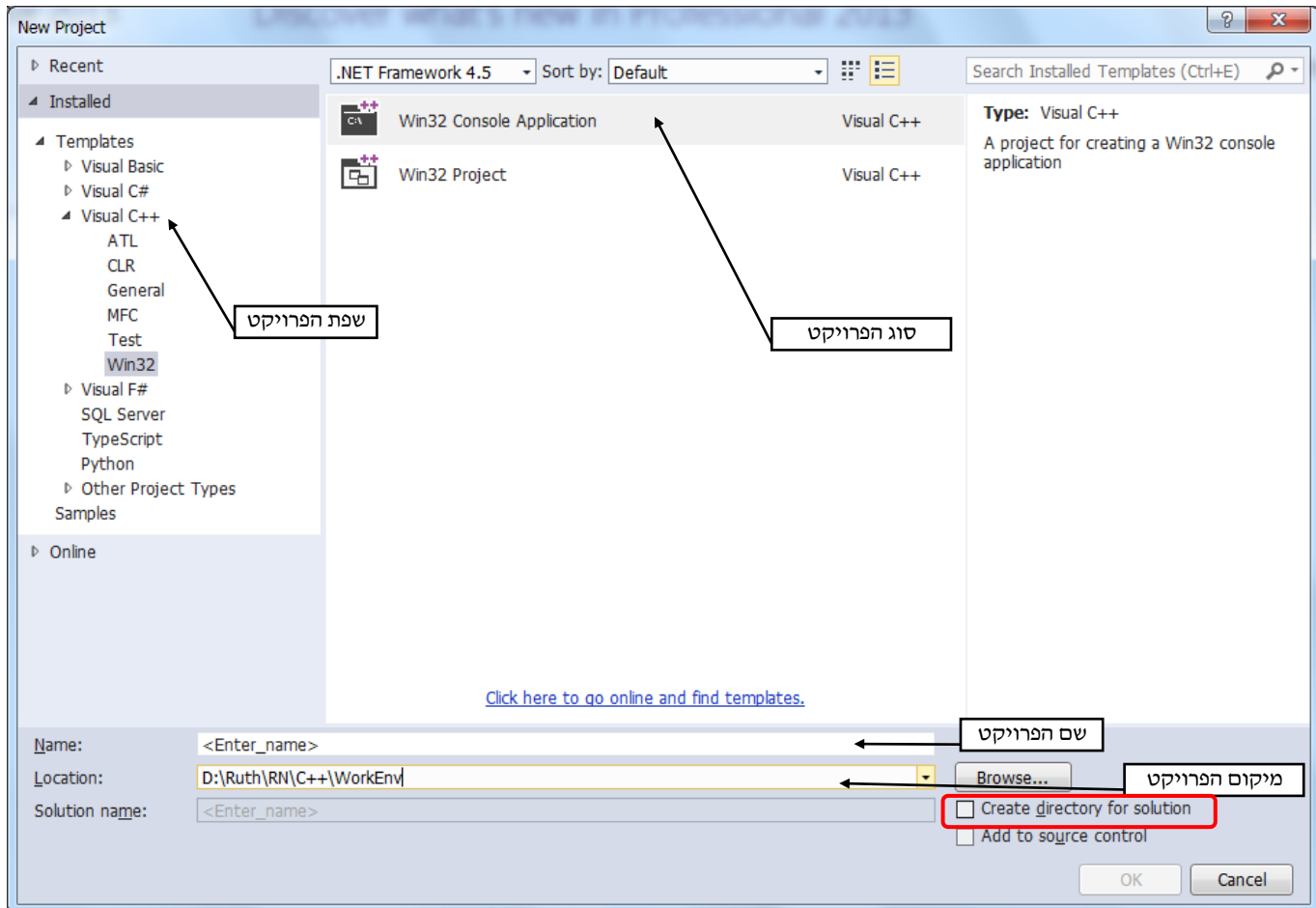
טיפול בפרויקט

יצירת פרויקט חדש

גש לשורת התפריטים וממנה בחר את האופציה File->New->Project (או שלחץ על Ctrl+Shift+N).



תפתח לך תיבת השיחה הבאה :



למה דווקא פרויקט? שים לב שבחרת ב
.New Project

Win32 Console Project

סוג פרויקט מיוחד, בו נשתמש לאורך
הקורס, המאפשר לנו כתיבת תוכניות
ל-DOS.

במידה והוספת את הפרויקט באמצעות
לחיצה ימנית על הפתרון, הפרטים
המסומנים במסגרת לא יופיעו.

תיבת שיחה זו שואלת אותנו לגבי מאפייני הפרויקט אותו ברצוננו ליצור.

יש להקפיד לבחור מתוך רשימת שפות הפרויקט בפרויקט משפת C++, ומתוך
רשימת סוגי הפרויקטים את הסוג "Win32 Console Project" (הסוג המסומן
בתמונת המסך).

**שים לב שאינך יוצר תיקיה עבור ה-Solution (ישנו הסבר על Solution ים בהמשך),
אלא רק תיקיה עבור הפרויקט.** זאת אומרת שהתיבה הימנית הממוסגרת,
אינה מסומנת.

בכל פעולה שהינך עושה, כאשר מופיע חלון הפעולה לא תסתיים עד שתלחץ על כפתור OK אישור וכדומה. לחיצה על ה X (בקצה הימני), Cancel או ביטול תגרום לביטול הפעולה!



קטע עלינו להכניס את שם הפרויקט. שם הפרויקט מורכב משני חלקים. שם הפרויקט, וכן מיקומו בדיסק. יש לוודא, לפני יצירת כל פרויקט, כי יצרת אותו במקום בו נתבקשת.

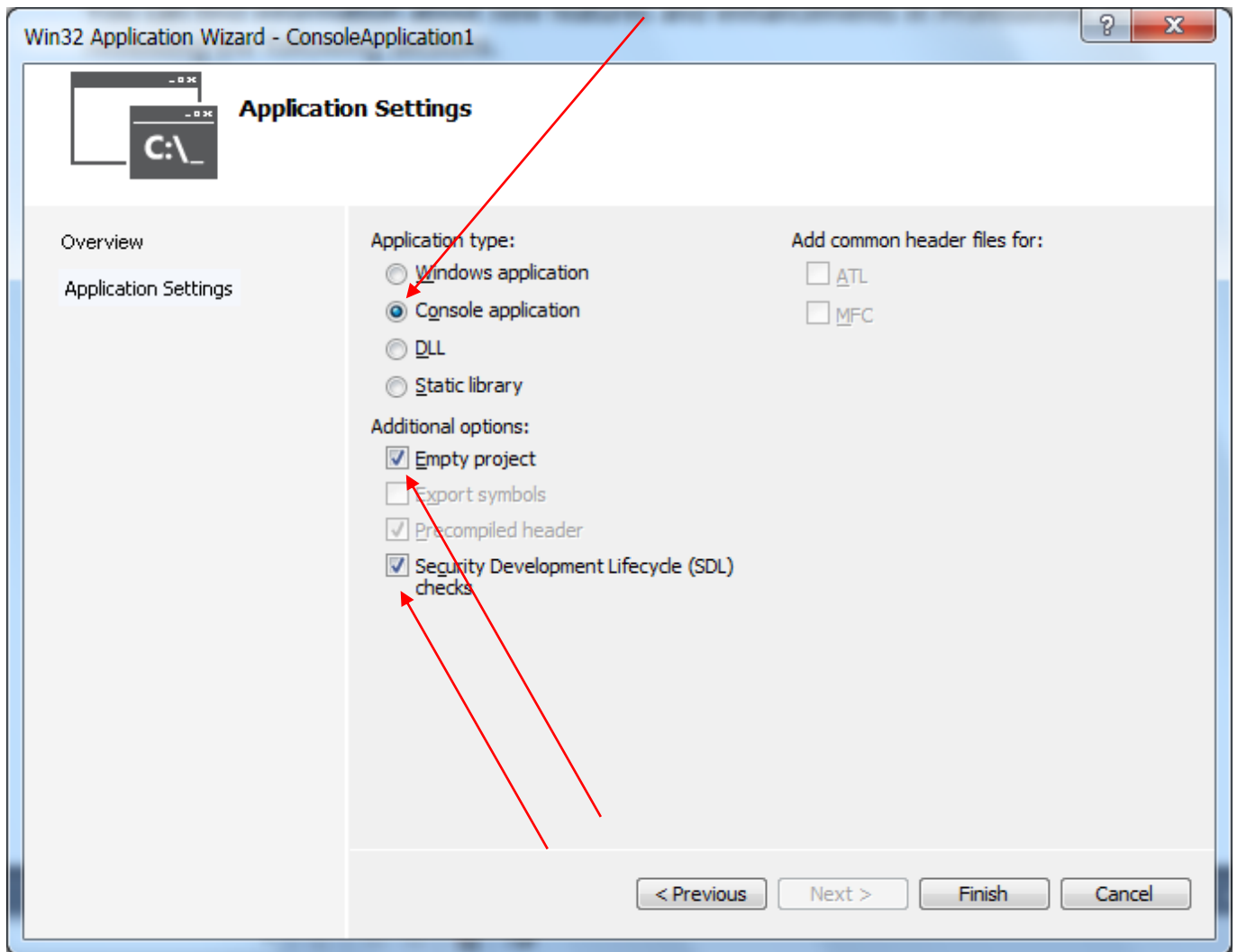
צור פרויקט חדש.
מלא את שדה "שם הפרויקט" במילה Tichnut.
שים לב כי שם הספרייה לתוכה יכנס הפרויקט מתעדכן אוטומטית כאשר מכניסים את שמו. (בשדה "מיקום פרויקט").
לאחר שבצעת פעולות אלה - לחץ על כפתור האישור (OK).



במהלך התרגולים אין בשום פנים ואופן לשמור את הפרויקט על כונן C. יש להשתמש אך ורק בכונן H לצורך כך.



אם בצעת את הפעולות כראוי, אמורה להופיע תיבת השיחה הבאה - לאחר שלחצת על Next או על Application Settings בצד שמאל:



ברצוננו לפתוח פרויקט ריק, ללא קבצים, ולכן נבחר באופציה התחתונה (סימון Empty project בV).

תיבת שיחה זו מאפשרת לנו ליצור פרויקטים שמכילים כבר קבצים עם תוכנית בסיסית ביותר, ממנה נוכל להמשיך לפתח את האפליקציה שלנו.

נכון לעכשיו בחר באפשרות "Console application", וודא שהאופציה התחתונה במסך (Empty project) מסומנת וגם האופציה של SDL, לחץ על Finish.



ובכן, כרגע יצרת את הפרויקט הראשון שלך.

כזכור הפרויקט מכיל רשימה של קבצים המרכיבים את התוכנית. כרגע רשימת הקבצים בפרויקט ריקה. ולכן אם נבקש מה-Visual Studio .NET להריץ את הפרויקט, נקבל הודעת שגיאה שכן אין ה-Visual Studio .NET יודע אילו קבצים עליו להריץ. עלינו לומר ל-Visual Studio .NET אילו תוכניות מרכיבות את הפרויקט שלנו, כלומר – להכניס קבצים לרשימה.

לצורך התרגול הכנו לכם תוכנית מראש בשם Start1.cpp.

אנו יכולים כעת להוסיף את קובץ זה לפרויקט, אך דבר זה אינו נוח. מדוע? הקובץ Start1.cpp שניתן לכם. כזכור, פרויקט יכול להכיל מספר קבצים, ולא רק קובץ אחד. מה היה קורה לו היה לנו פרויקט בעל 100 קבצים, וכל קובץ היה יושב בספרייה אחרת?

על מנת למנוע את חוסר הסדר הזה, Visual Studio .NET יוצר לנו ספרייה אחת בה אנו יכולים להשתמש כדי לאחסן את כל הקבצים המשתתפים בפרויקט (יכולים, אך לא חייבים). ספרייה זאת נקראת "ספריית הפרויקט". כאשר יצרנו את הפרויקט, אמרנו ל-Visual Studio .NET מהו מיקום הפרויקט בדיסק. במיקום זה נוצרה ספריית הפרויקט.

למען הסדר הטוב, אנו תמיד נמקם את כל הקבצים המשתתפים בפרויקט בספריית הפרויקט. לכן יש להעתיק את התוכנית Start1.cpp לספריית הפרויקט.

העתק את הקובץ START1.CPP.

השתמש ב"מחשב שלי" כדי לבצע פעולה זו (מזער את החלון של ה-Visual Studio .NET).



מלבד השימוש שלנו בספריית הפרויקט לאחסון קבצים, משתמש בה גם ה-Visual Studio .NET כדי לשמור מספר קבצים המכילים נתונים חיוניים על הפרויקט. הסתכל ב"מחשב שלי" על הקבצים הנמצאים בספריית הפרויקט. תוכל למצוא שם קבצים בשם TICHNUT.SLN, TICHNUT.VCPROJ ו-TICHNUT.NCB. בקבצים אלו שומר ה-Visual Studio .NET את אותם נתונים. במידה ואינך רואה אחד מהקבצים הנ"ל, אל חשש. הם יופיעו רק לאחר שתצא ה-Visual Studio .NET.

הזהר לא למחוק קבצים שאינך מכיר מספריית הפרויקט. שכן ה-Visual Studio .NET יוצר אוסף קבצים שונים לצורך ניהול הפרויקט.

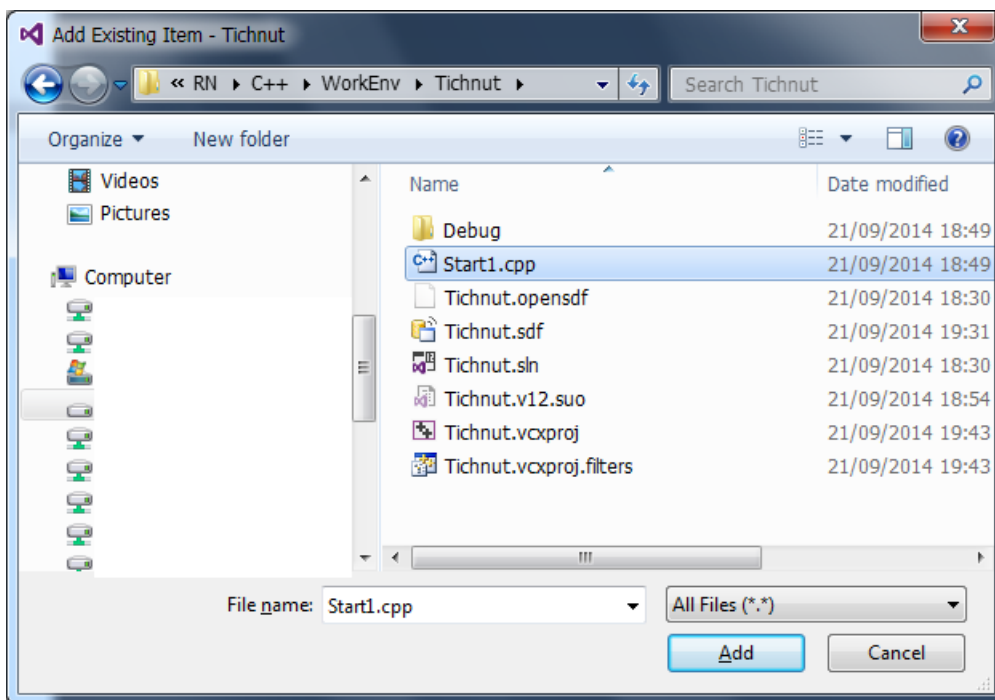


כפי שנאמר, בספריית הפרויקט אנו יכולים לשים, לנוחיותנו, את הקבצים המשתתפים בפרויקט (בקורס תמיד נעשה זאת). אך העובדה שקבצים אלו נמצאים בספרייה זו, אינה אומרת כי הם מוכללים עדיין בפרויקט. כלומר – הם עדיין אינם חלק מאותה רשימת קבצים שמחזיק הפרויקט.

להזכירך – כל פעולות העתקת הקבצים והעברתם לספריות אחרות, כמו שעשינו עבור הקובץ Start1.cpp, נעשות דרך "המחשב שלי". הפרויקט אינו מושפע מפעולות אלה. כעת עלינו להודיע לפרויקט כי אנו רוצים להוסיף את קובץ זה לרשימת הקבצים אותה הוא שומר.

חזור לחלון ה-Visual Studio .NET.
בחר באופציה Project→Add Existing Item... מתוך שורת התפריטים.
(זו השורה המופיעה בראש המסך).
ניתן גם לבצע את הפעולה : Alt+Shift+A.
מומלץ לבצע את פעולת ההוספה כאשר אחת מתיקיות הפרויקט מסומנות
(בד"כ Source Files).

אופציה זו גורמת להכנסת קובץ לתוך הפרויקט שלנו.
תופיע לעיננו המשתאות תיבת השיחה הבאה :



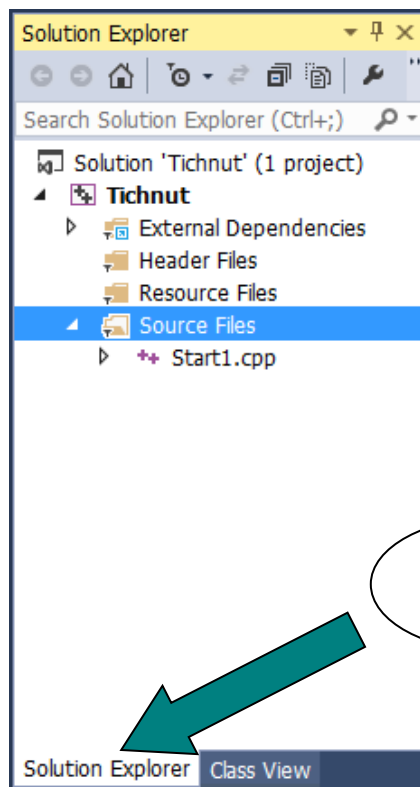
תיבה זו מציגה את רשימת הקבצים הקיימים כרגע בספרית הפרויקט.
אנו נבחר בקובץ אותו העתקנו לספרייה לפני שתי קטנות בדיוק.

הכנס בתיבת שם הקובץ את שם הקובץ אותו ברצונך להכניס לפרויקט. בחר
בקובץ Start1.cpp. (זכור לחוץ ADD בכדי לאשר את הפעולה)

מדוע סיומת הקובץ למעשה היא CPP?? הסיומת CPP אומרת ל- Visual Studio
.NET באיזה מהדר יש להשתמש כדי להדר את הקובץ.

**שים לב שעליך לציין את הסיומת CPP . במידה ולא תציין שום סיומת תילקח
סיומת ברירת המחדל (CPP)**

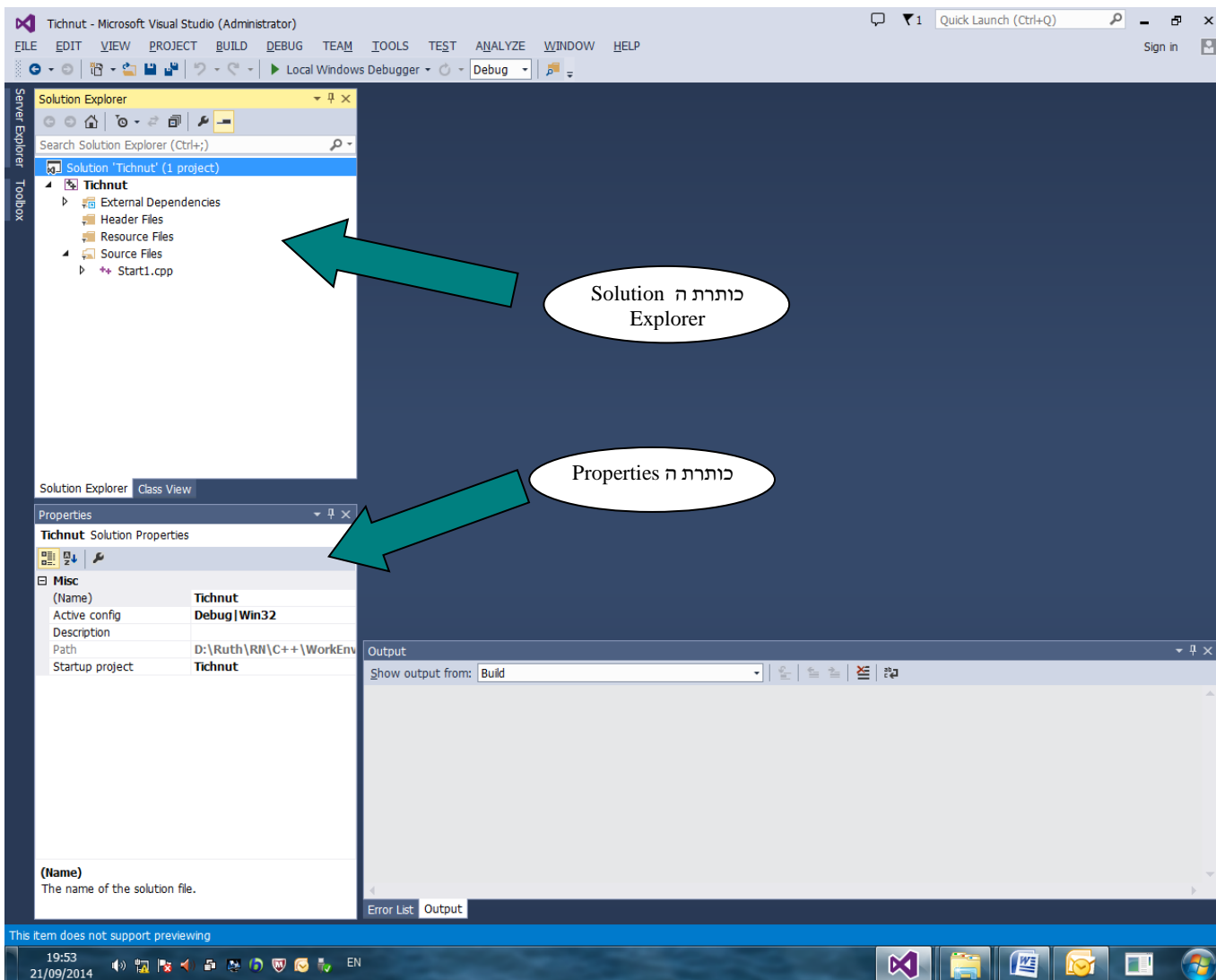
כדי לראות אילו קבצים נמצאים בפרויקט, יש לבחור, מתוך כפתורי החוצצים בתחתית חלון הפרויקט את החוצץ
Solution Explorer, כפי שמתואר בתמונה הבאה (ניתן להיעזר בתמונה שבעמוד 3 תחת המסגרת "חלון הצגת קבצים,
ומחלקות בפרויקט"):



וודא כי סימנת בחלון הפרויקט את חוצץ ה-Solution Explorer. החוצצים הם אותם כפתורים קטנים שמופיעים בתחתית חלון הפרויקט. חוצץ ה-Solution Explorer הוא החוצץ השמאלי. יש ללחוץ עליו באמצעות כפתור העכבר השמאלי.



אם פעלת כשורה המסך שלך אמור להראות כך :



במקרה שאינך רואה תחת שמות הקבצים את הקובץ Start1.cpp, וודא כי לחצת על חוצץ ה-Solution Explorer. אם לא לחץ עליו. אם הוא כבר לחוץ, התחל את כל פעולת הכנסת הקובץ מחדש.

מה זה החלון הזה שנפתח למטה בצד ימין?? זהו חלון המאפיינים. החלון הזה מציג מאפיינים של כמעט כל עצם שנשמך. נסה ללחוץ על הפרויקט עצמו (Tichnut) ב Solution Explorer, ושים לב שהמאפיינים בחלון הזה משתנים (נשתמש בו ה-מ-ו-ן, במקצוע אחר). ראה מה ההבדל בחלון המאפיינים כאשר בוחרים ב Start1.cpp מתוך ה Solution Explorer.

ניתן לאחד בין 2 החלונות ע"י לחיצה וגרירת כותרת ה- Properties לעבר כותרת ה Project Explorer (המלבן המסומן הכחול בתמונה, אבל בדיוק לשם).
כדי להגיע לחלון ה Properties : F4

כדי לראות את התוכנית עצמה, יש לבצע לחיצה כפולה על שם התוכנית.

הצג את התוכנית Start1.cpp.
שנה את שם התוכנית ואת מספר החניך שבהערה הראשית לשמך ולמספר החניך שלך.



ובכן, ביצענו שינויים בתוכנית. עלינו לשמור אותה.

שמור את התוכנית.
התמקד בחלון הסורס ובחר, מתוך שורת התפריטים את התפריט File→Save Start1.cpp (או שלחץ על Ctrl+S כמו המקצוענים מקורס תכנות).



אתה ודאי שואל את עצמך כיצד שומרים את הפרויקט עצמו. ובכן הפרויקט נשמר אוטומטית לאחר כל שינוי שעורכים בו.

כזכור הפרויקט אינו מכיל את התוכניות עצמן אלא רק אינפורמציה עליהן. ולכן השמירה האוטומטית של הפרויקט לא שומרת אוטומטית את התוכניות שבו.



נסה לבצע את המשימות האלו ללא שימוש בעכבר.

הרצה

כעת נרצה להריץ את הפרויקט. נכון שאנו מדלגים על שלב ההידור והקישור (שמכניסים בעצם את התוכנית לריצה) אולם נתעכב על כך לאחר מכן.

לחץ על צירוף המקשים Ctrl+F5.
בתיבת השיחה המופיעה בחר את כפתור ה"כן".



הפלא ופלא. התוכנית רצה. במקרה ואינך מצליח להריץ את התוכנית פנה למדריך הנמצא בכיתה.

מהו Solution ?

כמו שפרויקט מאגד מספר קבצים הקשורים לאותה תוכנה, פתרון מאגד מספר פרויקטים הקשורים זה לזה באיזשהו אופן.

אין דבר כזה פרויקט ללא פתרון (Solution).

אנחנו כבר יודעים שה Visual Studio .NET יודע להדר ולקשר אך ורק פרויקטים. אז אנחנו חייבים פרויקט (שאותו למדנו איך ליצור לפני מספר עמודים).

אולם, פרויקט יימצא תמיד בתוך פתרון. קודם אנחנו לא יצרנו את הפתרון, ולכן ה-Visual Studio .NET יצור את הפתרון עבורנו בצורה אוטומטית.

ולכן, כשנתחיל פרויקט חדש אין צורך לפתוח פתרון חדש לפני כן, הוא נוצר לנו אוטומטית.

ניתן ליצור פתרונות ריקים באמצעות התפריט File – New – Blank Solution, או באמצעות תיבת הדיאלוג אותה ראינו קודם.


פתיחת/סגירת פתרון (Solution)

סגירה של פתרונות (ואיתם סגירת הפרויקטים) מתבצעת ע"י התפריט File→Close Solution.

פתיחה של פתרון, אם כן, מתבצעת ע"י האופציה File→Open Solution... (ולא File→Open→File...).

כדי לפתוח פרויקט בחר בתפריט
File->Open->Project...

סגור את הפתרון Tichnut.

פתח את הפתרון (בעזרת OPEN SOLUTION - פתיחת פתרון) BigMama. שים לב שגם הפתרון הוא קובץ בסיומת SLN שלידו הציור 



יצירת קבצי תוכניות

עד עכשיו למדנו כיצד ליצור פתרון ופרויקט ולהכניס לתוכו קבצים שכבר יושבים בדיסק. כיצד ניצור את התוכנית עצמה? כלומר איך נוסיף לפרויקט קובץ חדש?

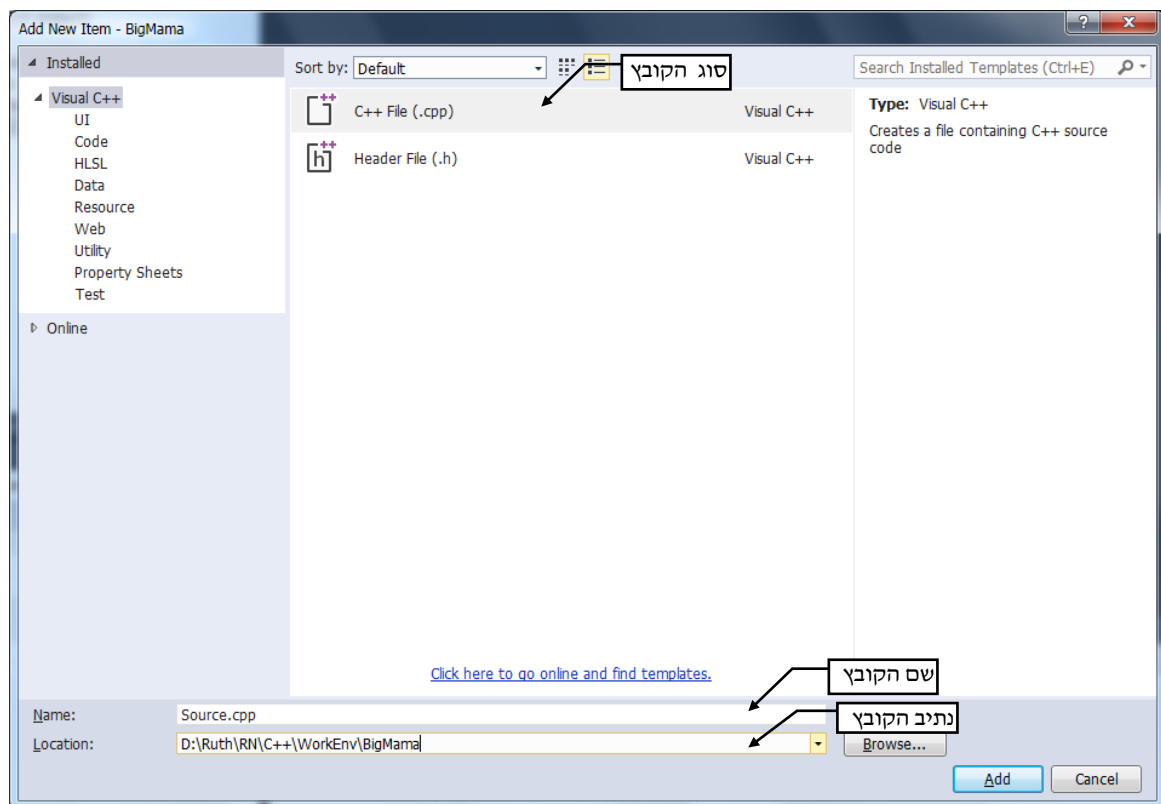
הפרויקט שפתחתם (BigMama) מכיל מספר קבצים. עם זאת יש בו מספר בעיות. אחת מהן היא העובדה כי קובץ אחד בפרויקט חסר. את הקובץ הזה, תוכנית למעשה, אתה תכתוב.

ראשית כל יש ליצור את הקובץ.

סמן את תיקיית ה Source Files וצור קובץ מקור חדש בפרויקט ע"י בחירת האופציה Project→Add New Item..., או ע"י לחיצה על **Ctrl+Shift+A**.



נפתחה תיבת שיחה דומה לזו שנפתחה כאשר יצרנו את הפרויקט שלנו :



ראשית, עלינו לבחור מתוך רשימת האופציות את סוג הקובץ. **שימו לב:** אם לא מופיעה הרשימה המלאה, וודאו שבחלון הימני מסומנת הקטגוריה Visual C++ כולה.

סמן את האופציה C++ File (.cpp). כעת נזין שם לקובץ, ונוודא שהוא נשמר תחת נתיב הפרויקט. (שם הקובץ First.cpp)



שים לב כי באפשרותך להוסיף את הקובץ לפרויקט, או סתם ליצור קובץ ולערוך אותו להנאתך באמצעות עורך הטקסט המצוין של Visual Studio .NET באמצעות **File→New→File...** או **Ctrl+N**.

כאשר תלחץ על אישור, יופיע לפניך חלון חדש, ריק ולבן. בחלון זה תכתוב את התוכנית.

הקלד את השורות הבאות בחלון:



```
#include <iostream>
using namespace std;
void print(int x)
{
    cout << x << endl;
}
```

אתה עדיין לא מודע לעובדה זו אבל השורות שכתבת מרכיבות תוכנית ב-C++.

שמור את התוכנית שכתבת (זוכר? ע"י File→Save).



מאחר וקודם בחרנו באפשרות של להוסיף קובץ לפרויקט, הקובץ נשמר אוטומטית בספריית הפרויקט.

הידור, קישור, ריצה ומה שבאמצע

פתח את הפרויקט Debug.
והוסף את הקובץ First.cpp כקובץ נוסף לפרויקט (ctrl+shift + A).



הידור וקישור

לאחר שלמדנו לערוך תוכנית, הגיע הזמן להעביר אותה את שלב הקומפילציה. דבר זה מתבצע ע"י לחיצה על צירוף המקשים Ctrl+f7. ניתן לעשות פעולה זו גם דרך שורת התפריטים, ע"י תפריט Build→Compile.

פעולה זו תתחיל את תהליך ההידור. בניגוד לסביבות ישנות בהן היה צורך לחכות בזמן עריכת ההידור, ניתן להמשיך לעבוד על התוכנית ב- Visual Studio .NET. את הודעות קומפילציה אנו נראה בחלון המופיע בתחתית המסך (חלון (.Errors

קמפל את התוכנית המופיעה על המסך. (וודא כי זוהי התוכנית (First.cpp).



הבעיה עם אופציית ההידור היא שלמעשה היא מהדרת רק את הקובץ בו אנו נמצאים - First.cpp. אך בפרויקט זה ישנו עוד קובץ אחד - Main.cpp (תוכל לראות את שמו בחלון הפרויקט - זה שמצד שמאל). אנו רוצים להדר גם אותו, או במילים אחרות - להדר את כל הפרויקט ולא רק קובץ אחד.

לשם כך אנו נבחר, מתוך שורת התפריטים את האופציה Build→ReBuild Solution אופציה זו תהדר את כל הפרויקט, ואם ההידור עבר בהצלחה, גם תקשר אותו.

במידה ועולה בעיה כלשהי במהלך בניית הפרויקט תתקבל, בחלון התחתון של ה- Visual Studio .NET, הודעת שגיאה מתאימה.

בנה את הפרויקט (ע"י האופציה Build→ReBuild All).

ניתן לעשות זאת גם ע"י Ctrl+Shift+B.



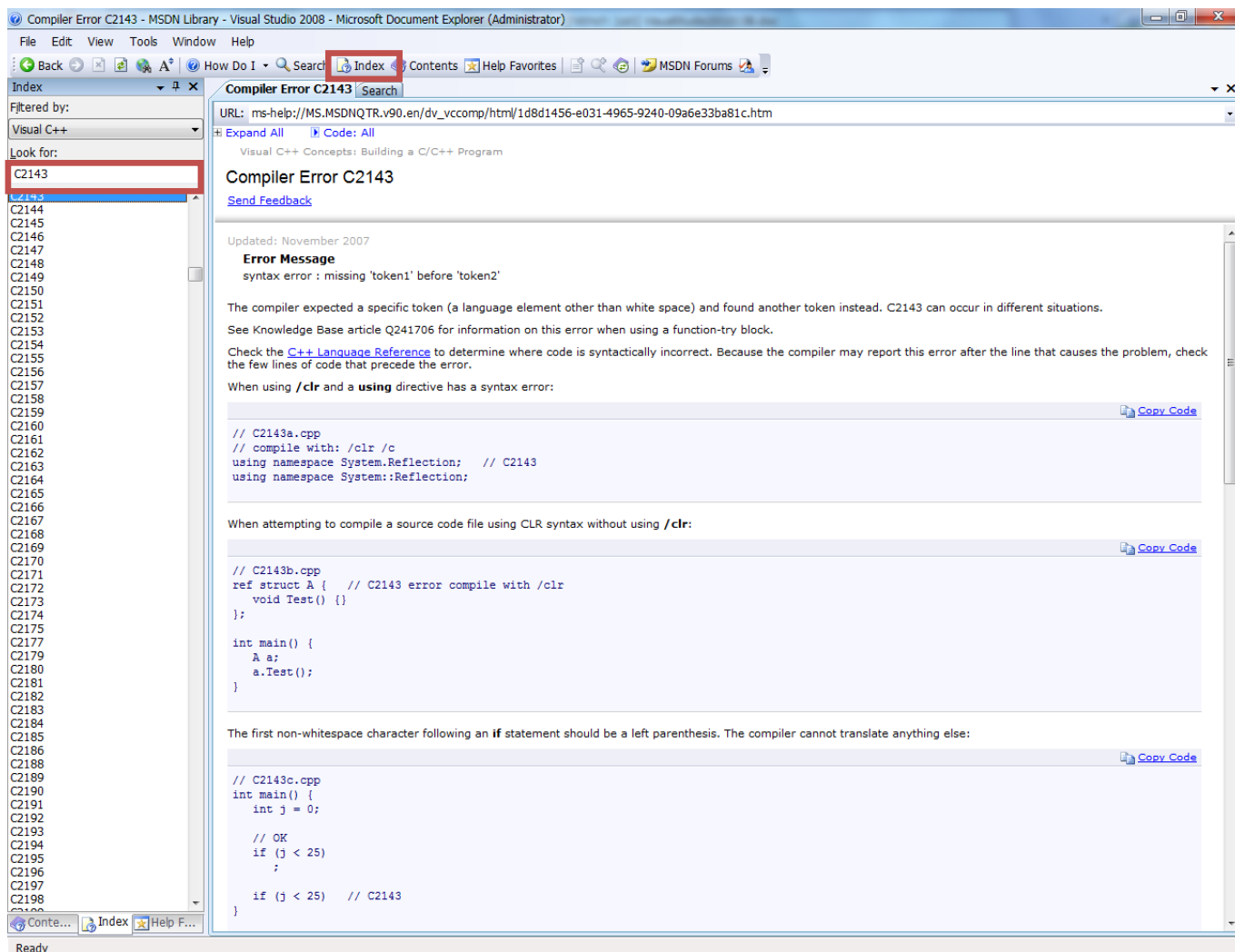
הפלא ופלא קיבלנו הודעת שגיאה. ליתר דיוק קיבלנו כמה. הראשונה שבהן היא:

```
error C2143: syntax error : missing ']' before ')'
```

הודעת השגיאה לא מפרטת במדויק מה הבעיה. כיצד נקבל הסבר מעמיק על הסיבה לשגיאה שקיבלנו?

העתק את מספר השגיאה (C2143 במקרה זה) והדבק אותו בשורת החיפוש ב- MSDN (אחרי שווידאתם שאתם מחפשים ע"י שימוש באופציית index) *תמונת הסבר בע"מ הבא







אוקיי. מה שקיבלנו הוא חלון המפרט את השגיאה, דוגמאות מובחרות למה השגיאה נוצרה, ודרכים לתיקון.

המידע בעזרה יכול להכיל הרבה אודות דברים נוספים ודברים שאנחנו לא משתמשים בהם, אם ניתן, צריך למצוא את המידע הרלוונטי אלינו בלבד. נסה להבין היכן נמצאת השגיאה ומדוע היא נגרמה.

הצלחת? יופי. לא הצלחת? לחיצה כפולה על שגיאת המהדר תביא אותנו לשורה בה קיימת השגיאה.

לחץ לחיצה כפולה על שורת השגיאה. שים לב כי החלון המוצג הוא של תוכנית אחרת בפרויקט. 

שים לב לחלון הסורס. האם אתה רואה את השורה מסומנת?

עשה את השינויים המתאימים בקובץ כדי לתקן את הטעות (יש להפוך את הסוגר העגול בשורה 31 לסוגר מרובע). 
בנה שוב את הפרויקט.

מזל טוב, בנית את הפרויקט הראשון שלך.

אם תהית, מעכשיו אנו נהדר את התוכנית שלנו רק ע"י שימוש ב **ctrl + shift + B**


הרצה

בתור עזרה למפתח, הרצת פרויקט ב-
Visual גוררת בדיקה של הקבצים
בפרויקט, והידורם וקישורם מחדש
במקרה והם אינם מעודכנים.

לאחר שהעברנו הידור וקישור ללא שגיאות נרצה להריץ את התוכנית. הקש
Ctrl+F5 וצפה בהנאה כיצד התוכנית שכתבת רצה.

מה שבאמצע

למרות שאנו מריצים בד"כ את הקובץ מתוך ה-Visual Studio .NET הרי שאין
מניעה להריץ אותו ידנית. מה זאת אומרת?
לאחר סיום שלב ההידור והקישור נוצר קובץ הרצה (בעל סיומת EXE). קובץ זה
ניתן להריץ עצמאית, בלא שימוש ב-Visual Studio .NET או כלי אחר, ממש כפי
שניתן להריץ כל תכנית אחרת תחת Windows. הבה נעשה זאת.

עבור לספרייה בה נמצא הפרויקט. 

לצורך כך, מזער את תוכנת ה-Visual Studio .NET (אנו הולכים להריץ את
התוכנית שלנו בלי השימוש של ה-Visual Studio .NET).

לחץ לחיצה כפולה על הקובץ Debug.exe.

צפה בהרצה של התוכנית.

חזור ל-Visual Studio .NET.

למעשה פקודת ההידור יוצרת בדיסק עבור כל תוכנית בפרויקט קובץ בשם זהה
לקובץ המקורי, אלא שהסיומת שלו היא Obj. לדוגמה עבור Start.cpp ייווצר
הקובץ Start.Obj.

המקשר לוקח את כל קבצי ה-Obj שנוצרו ויוצר קובץ אחד, ששמו כשם
הפרויקט, עם סיומת Exe. לדוגמה עבור הפרויקט שלנו יוצר קובץ Debug.Exe.
קובץ זה הוא קובץ ריצה שגם מערכת ההפעלה יכולה להריצו ישירות.

מתי נשתמש בכל פעולה?

| | |
|-------|--|
| הרצה | פעולת ההרצה תתבצע בסוף כתיבת כל קטע לוגי. לאחר שכתבנו את הקטע הלוגי, ברצוננו לבדוק שהוא פועל כשורה. במקרה כזה אנו נעשה את פעולת ההרצה. פעולה זו (כזכור מופעלת ע"י Ctrl+F5) טורחת גם לבנות את הפרויקט אם הוא אינו עדכני. |
| הידור | פעם ב-5 או 10 שורות בריא להעביר את התוכנית הנוכחית הידור. סתם כדי לראות שהתחביר נכון. הפעולה גם דואגת לשמור את התוכנית באותה הזדמנות. ההידור בפני עצמו לא יאמר לנו דבר על נכונותה הלוגית של התוכנית, רק על התחביר שלה. שים לב שהעורך המופלא של ה Visual Studio .NET עוקב אחרי הכתיבה שלך, ומסמן לך שגיאות בזמן הכתיבה, ומפרט לך עליהם במסך התחתון תחת הכותרת Tasks List. ניתן להגיע למסך הזה ע"י לחיצה על Alt+Ctrl+K ולצאת ממנו ע"י לחיצה על הכפתור Esc. |
| Build | פעולת ה-Build שימושית כאשר שינינו כמה קבצים בפרויקט ואנו מעוניינים להעביר את כולם הידור. עדין לא סיימנו את הקטע הלוגי עליו אנו עובדים, ואיננו מעוניינים לבדוק ריצה, אבל אנו מעוניינים לדעת כי התחביר נכון וכי הקשרים בין התוכניות תקינים. |

סיכום

שיעור 2, אותו סיימת ברגע זה, אפשר לך לראשונה לכתוב תוכנית ולהריץ אותה. למדת כיצד לעבוד עם פרויקט (לפתוח, לסגור, ליצור וכן להוסיף לו קבצים) וכן כיצד להזדרז, לקשר ולהריץ.

אותן שתי שורות שרשמת היום, יהפכו לאוסף של מאות רבות של שורות קוד שתכתוב בעוד 5 חודשים. השיעור הזה הוא כמו אותו צעד ראשון בדרך של אלף מילין אותה עליך לעבור. בהצלחה.

עבודה יעילה

מקלדת

העבודה האמיתית של תוכניתן נעשית על מקלדת. מה לעשות, העכבר עדיין לא קורא מחשבות. אי לכך ובהתאם לזאת, כל פעולה שיש לעשות על העכבר פחות נוחה. מה לעשות, אבל להיזיז כל הזמן את הידיים מהמקלדת ולמקלדת - ובכן זה לא נוח.

לכל פעולות ה-Visual Studio .NET יש קיצורי דרך במקלדת. החל מהפשוטים (F5) - להרצה בדיבאג של תוכנית) וכלה בכאלה שאין להם מקבילה ישירה עם העכבר (Ctrl+L גורם למחיקה של שורה).

סימון

אחד השימושים הנוחים ביותר הוא הסימון. פעולת הסימון היא פעולה שעושים פעמים רבות - להעתיק להעביר למחוק. כל הפעולות הללו דורשות סימון של קטע טקסט מסוים. אבל אלא אם כן יש לך תואר ראשון באקרובטיות, סימון הטקסט אינה פעולה כה פשוטה.

ולכן ניתן לסמן טקסט באמצעות המקלדת. פשוט יש ללחוץ על מקש ה-Shift ועל תנועה רגילה במקלדת (אחד מהחצים למשל). הדבר יזיז את הסמן וגם יגרום לסימון הטקסט. נסה זאת על אחת התוכניות שלך, וצפה בפלא בפעולה.

תנועה מהירה

לחיצה על מקש ה-Ctrl ועל אחד ממקשי החצים תביא לפעולות מדהימות. לחיצה על Ctrl ועל החץ השמאלי תזיז את הסמן מילה אחת שמאלה. לחיצה על Ctrl ועל החץ הימני תזיז את הסמן מילה אחת ימינה. לחיצה על מקש ה-Home ועל ה-Ctrl תביא להזזת הסמן לתחילת הקובץ.

קיצורי דרך

את רשימת כל קיצורי המקלדת ניתן למצוא בתפריט ה-Help תחת האופציות: Keyboard Shortcuts, Keyboard mapping schemes, Shortcut Keys.

בטבלה סיכום קצר של הקיצורים היותר שימושיים. לא הכל שימושי לכם עכשיו, חלק יראו לכם ממש מוזרים. מדי פעם תסתכלו בדף הזה, ותגלו שמהו שכל הזמן עשיתם ידנית ניתן לביצוע באמצעות קיצור. הדברים הממש חשובים מסומנים בכוכבית.

עוד משהו: בלחיצה על התפריט Tools->Customize יפתח חלון Customize. בחר ב- Show shortcut keys in ScreenTips (Close) כעת אם תעמוד על כל כפתור בסרגלי הכלים למשך כמה שניות, יופיע טיפ (ToolTip) המציין את קיצור הדרך לפעולה זו.

| פעולה | צרוף מקשים | הסבר |
|-----------------------------|-------------------|---|
| Toggle a bookmark on or off | Ctrl+K, Ctrl+K | Toggle a bookmark to the current line |
| Next Bookmark | Ctrl+K, Ctrl+N | Move to the next bookmark |
| Comment Selection | Ctrl+K, Ctrl+C | Put the selection text in '/' comment |
| Uncomment Selection | Ctrl+K, Ctrl+U | Uncomment the selected '/' commented text |
| Class View | Ctrl+F2 | Move to class view |
| *FindNextWord | Ctrl+F3 | Finds the next occurrence of the selected text |
| LineDelete | Ctrl+Shift+L | Deletes the selected line |
| LineCut | Ctrl+L | Deletes the selected lines and puts them on the Clipboard |
| *WordLeft | Ctrl+Left Arrow | Moves back one word |
| *WordRight | Ctrl+Right Arrow | Moves forward one word |
| *GoToMatchBrace | Ctrl+J | Finds the matching brace |
| LineDownExtend | Shift+Down Arrow | Extends the selection down one line |
| LineEndExtend | Shift+End | Extends the selection to the end of the current line |
| CharLeftExtend | Shift+Left Arrow | Extends the selection one character to the left |
| PageDownExtend | Shift+Page Down | Extends the selection down one page |
| PageUpExtend | Shift+Page Up | Extends the selection up one page |
| CharRightExtend | Shift+Right Arrow | Extends the selection one character to the right |
| LineUpExtend | Shift+Up Arrow | Extends the selection up one line |
| LineEnd | End | Moves to the end of the current line |
| Home | Home | Moves to either the start of the current line or the start of the text on that line |
| PageDown | Page Down | Moves the cursor down one page |
| PageUp | Page Up | Moves the cursor up one page |
| EditToggleOvertyp | Ins | Toggles between inserting and replacing text |
| DeleteBack | Backspace | Deletes the selection or, if there is no selection, the character to the left of the cursor |
| *IndentSelection | Tab | Indents the selected text right one tab stop |
| **Compile | Ctrl+F7 | Compiles the file |
| **Build | F7 | Builds the project |
| *BuildExecute | Ctrl+F5 | Executes the program |
| *Cut | Ctrl+X | Cuts the selection and puts it on the Clipboard When no selection, cuts whole line |
| *SelectAll | Ctrl+A | Selects the entire document |
| *Paste | Ctrl+V | Inserts the Clipboard contents at the insertion point |
| *FileSave | Ctrl+S | Saves the document |
| *FileSaveAll | Ctrl+Shift+SL | Saves all the open files |
| GoToNextErrorTag | F4 | Moves to the line containing the next error or tag |
| FindReplace | Ctrl+H | Replaces the specified text with different text |
| Copy | Ctrl+C | Copies the selection to the Clipboard |

| | | |
|-----------------------|----------------|--|
| | | When no selection, copies whole line |
| Cut | Shift+Del | Cuts the selection and puts it on the Clipboard |
| Delete | Del | Deletes the selection |
| Redo | Ctrl+Y | Redoes the previously undone action |
| Undo | Ctrl+Z | Undoes the last action |
| *Cancel | Esc | Hides windows or cancels modes |
| ToolStop | Ctrl+Break | Stops any running tool or build |
| **DebugQuickWatch | Shift+F9 | Performs immediate evaluation of variables and expressions |
| DebugStopDebugging | Shift+F5 | Stops debugging the program |
| DebugStepOut | Shift+F11 | Steps out of the current function |
| DebugToggleBreakpoint | F9 | Inserts or removes a breakpoint |
| *DebugGo | F5 | Starts or continues the program |
| *DebugStepInto | F11 | Steps into the next statement |
| *DebugStepOver | F10 | Steps over the next statement |
| *DebugRunToCursor | Ctrl+F10 | Runs the program to the line containing the cursor |
| DebugSetNextStatement | Ctrl+Shift+F10 | Sets the instruction pointer to the line containing the cursor |
| *Switch Document/File | Ctrl+TAB | Moves to the last document edited (Like Alt+TAB for windows) |
| *Close Document/File | Ctrl+F4 | Closes the current document (Like Alt+F4 for windows) |
| *Solution Explorer | Alt+Ctrl+L | Highlights the Solution Explorer |
| *Properties Window | F4 | Highlights the Properties Window |
| *File Open | Ctrl+O | Opens a file |
| *Project Open | Ctrl+Shift+O | Opens a project |
| *New File | Ctrl+N | Create a new file |
| *New Project | Ctrl+Shift+N | Create a new project |
| *Add Existing Item | Alt+Shift+A | Add an existing item to the project |
| *Add New Item | Ctrl+Shift+A | Add a new item to the project |

עכבר

למרות האמור לעיל הרבה אנשים דווקא נהנים מהעכבר. מיקרוסופט לא הזניחה את מעריצי העכבר ברחבי העולם ונתנה כמה טיפים ששווה לדעת:

לחיצה ימנית בכל מקום במסך תביא תפריט עם כמה אופציות שימושיות מאוד. מומלץ ביותר!!!

ניתן לגשת לכמה פעולות שכיחות בעזרת סרגלי הכלים. סרגלי הכלים הם אוספי הכפתורים שנמצאים בראש המסך. בנוסף לסרגלים המופיעים בצורה אוטומטית עם פתיחת ה-Visual Studio .NET ניתן להוסיף עוד כמה. לחץ, בעזרת כפתור העכבר הימני (אלא מה), על שטח סרגלי הכלים, ובחר את סרגלי הכלים שברצונך להציג. כמו כן ניתן לגרור את סרגלי הכלים עם כפתור העכבר השמאלי לכל מקום במסך.

לחיצה כפולה על מילה תגרום לבחירת כל המילה.

ניתן לגרור את הטקסט ממקום למקום כדי להעביר אותו. לחיצה על מקש ה-Ctrl בזמן גרירת הטקסט תגרום להעתקתו.

טיפים כללים

סימניות

כמו סימניות בספר ניתן לשים סימניות בקוד. בעזרת סימניות בקוד ניתן לנוע ממקום למקום במהירות (נניח, איפה עשיתי את לולאת הקלט מהמשתמש בסורס בין ה-2000 שורות). פקודות על Bookmark (סימניה בלעז) ניתן למצוא בסרגל הכלים Edit.

הגדלת המסך

בנוסף להגדלת המסך לגודלו המקסימאלי, ניתן להגדיל את שטח הצפייה בתוכנית. ראשית ניתן להסתיר חלונות שאינם מעוניינים אותנו - לחץ בשטח סרגלי הכלים על כפתור העכבר הימני ובטל אחד השמות המסומנים או את כולם. זה יבטל, בנוסף לסרגלי הכלים, את חלון הפרויקט ואת חלון הפלט. שימושי בעת כתיבת התוכנית.

אם כל זה עדין לא מספק אותך, נסה את הטריק הבא: בתפריט View נסה את האופציה Full Screen. כל המסך, כולל תפריטים ושורת המצב של Windows יפנו את מקומם לשטח התוכנית שלך. כדי לצאת ממצב זה לחץ על מקש ה-Escape. איך עוברים ל Full Screen בלי עכבר?

חיפוש והחלפה

אם לא שמת לב, ניתן להשאיר את תיבת החיפוש ואת תיבת ההחלפה פתוחות. זה מאפשר לנו למצוא טקסט מסוים, לערוך את התוכנית, ואז לחזור ולהמשיך לחפש את הטקסט.

שמירה אוטומטית

ב-Microsoft מכירים תוכניתנים. תוכניתן הוא אדם שיכול לכתוב קוד בן 2000 שורות ולא לשמור אותו, בדיוק כשברדיו הזהירו על הטייפון "גפרי" שמתקרב לתחנת הכוח האזורית.

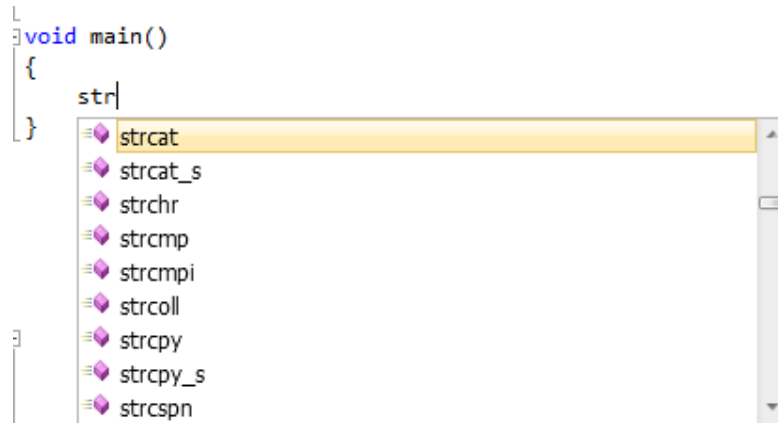
הפתרון? שמירה אוטומטית של קבצים בפרויקט. זה קורה לפני כל קומפילציה שמעבירים, ומאפשר לשמור את הגרסה של הפרויקט גם אם התוכנית (ואיתה כל Windows) עפה בדיוק לפני ששמרתם.

השלמה אוטומטית

בוא נכיר עוד פיצ'ר קטן ומגניב שיש בעבודה עם העורך ב-Visual Studio .NET. אם אתה רוצה להשתמש בפונקציה מחרוזתית שאינך זוכר את שמה, כל שעליך לעשות הוא להקיש את תחילת הפונקציה (STR במקרה שלנו), לבחור את האופציה המדוברת ותופיע רשימה של כל הפונקציות המתאימות. ומה שישמש אותנו הכי הרבה – Ctrl+Space לבקשת השלמה אוטומטית **בכל רגע**.

ו TAB לבחירת הפריט.

זה נראה פחות או יותר כך –



Debug בסיסי

תיקון שגיאות בתכנית

הסיבה ללמה באג נקרא באג היא שפעם המחשבים היו גדולים ותפסו שטח של חדר שלם. היו מסתובבים גוקים בתוך המחשב וגורמים לשגיאות חישוב ושגיאות בתוכנית.

כפי שתגלה במהלך עבודתך, בכל תוכנית ישנן שגיאות, גם לאחר שהעברנו את שלב ההידור (Compilation). שגיאות אלה נקראות באגים, מלשון מקקים בלעז, שכמו שגיאות בתוכנית גם הם מעצבנים ומתחבאים היטב.

באגים יכולים להיגרם ממספר רב של סיבות אך לכולן דבר משותף אחד - קשה לעלות עליהן. כמובן שה-Visual Studio .NET נותן לנו את האפשרות לטפל בבעיה זו. הכלי המקובל לגילוי באגים נקרא דיבאגר (Debugger). זהו כלי שנותן אפשרות מעקב אחר ריצת התוכנית (לאן היא מסתעפת, מה היא עושה ומתי) ולבדוק בכל רגע נתון ערך של משתנה. אנו נראה כיצד כלי זה מיושם בסביבת ה-Visual Studio .NET.

היכן נמצא הדיבאגר

במערכות ישנות יותר הדיבאגר היה כלי נפרד, ולא היה חלק מסביבת העבודה. ב-Visual Studio .NET הדיבאגר הוא חלק אינטרגלי מהסביבה ולכן לוקח זמן מועט יחסית כדי ללמוד להשתמש בו.

כדי ללמוד להשתמש בדיבאגר הוסף את הקובץ StartI לפרויקט תדש והרץ את התוכנית רק כדי לדעת מה היא עושה.



שימוש בדיבאגר

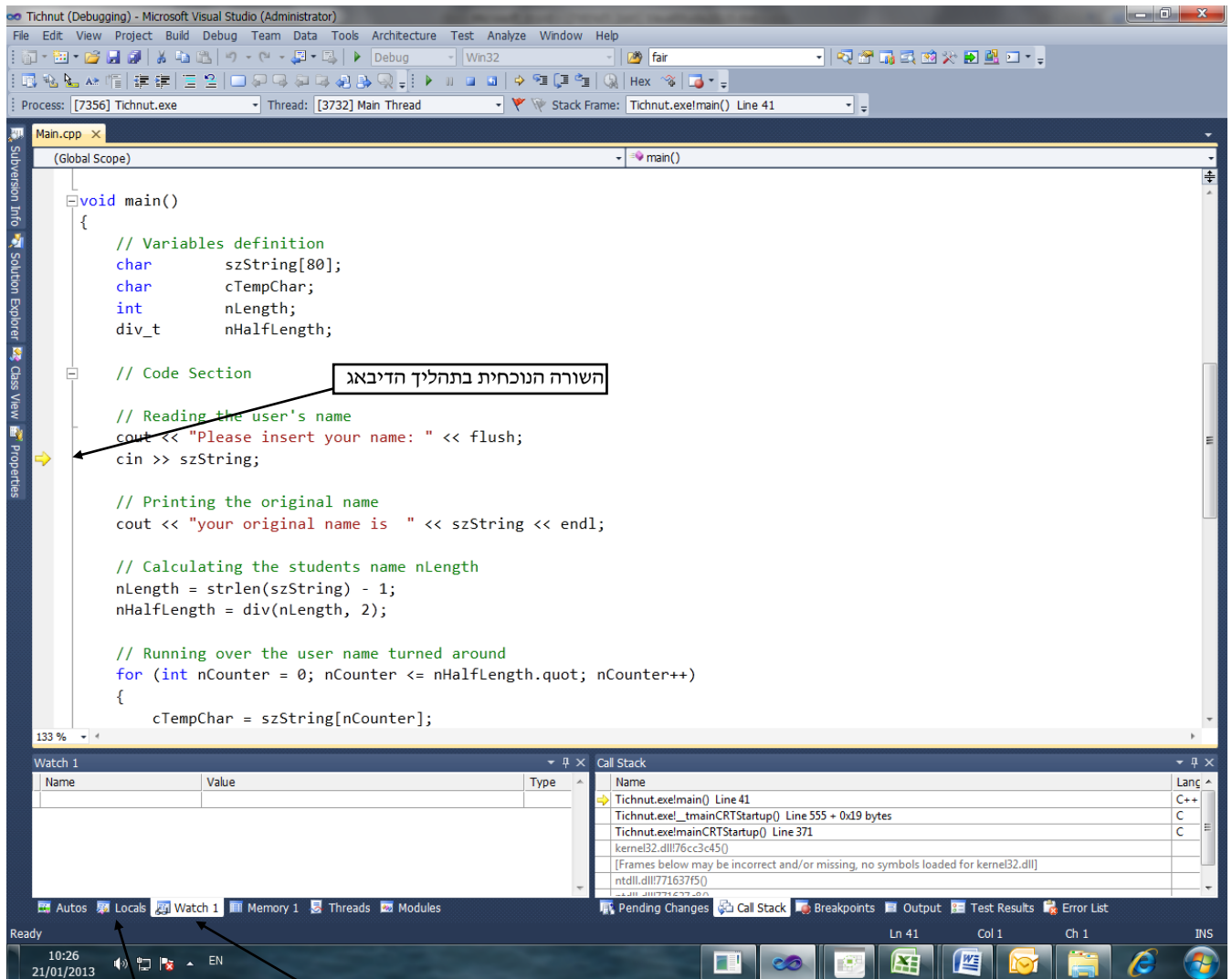
כדי להשתמש בדיבאגר אין צורך להריץ תוכנית נוספת. כל פקודות הדיבאגר חשופות לנו מתוך ה-Visual Studio .NET. כעת נעבור עליהן.

התוכנית MAIN היא תוכנית פשוטה ביותר. היא לוקחת מספר מהמשתמש והופכת את ספרותיו. בעזרת הדיבאגר נראה כיצד היא עושה זאת.

לחץ על המקש F10.



לחיצה על המקש F10 זהה להפעלת פקודה הנקראת Step. פקודה זו גורמת לכך שאנו נעבור שורה אחר שורה על התוכנית שלנו במצב דיבאג.



אם תשים לב תראה חץ צהוב המצביע על שורה בתוכנית. חץ זה אומר לנו היכן אנו נמצאים בתוכנית.

המשך בלחיצות על F10 והתקדם לשורה בה מתחילה לולאה. שים לב שכאשר הסמן הצהוב נמצא על פקודת ה-cin, לא ניתן להמשיך ולהתקדם בתוכנית. הדבר מתרחש מפני שה-Visual Studio .NET מצפה לקבל קלט מהמשתמש. עבור לחלון התוכנית הרצה, הכנס את המספר 10, והקש Enter. חזור לחלון ה-Visual Studio .NET והמשך ללחוץ על F10.



שים לב, יש ללחוץ על המקש F10 ולא על המקש F11. לחיצה על F11 כאשר הסמן נמצא על פונקציה ספרייה של C++ (כמו cin, cout וכיו') תגרום להצגת קוד האסמבלר שבו הן נכתבו. כדי לצאת ממצב זה יש להמשיך ללחוץ F10 עד שנגיע לקוד הרגיל בשפת C++.

למעשה עברנו כבר חלק מהתוכנית. בשלב זה חלק מהמשתנים כבר מכילים ערכים. כיצד נבדוק אותם?

עמוד עם סמן העכבר על המשתנה nUserNum שנמצא בשורה 38. אמור להופיע חלון שיציג את ערך המשתנה. אם לא מופיע חלון זה, נסה למקם את הסמן טוב יותר וחכה כשנייה. המשך בניסיונותיך עד שיופיע החלון.
* אם אינך רואה את מספרי השורות מצד ימין יש להוסיף ע"י לחיצה על
Tools -> Options -> Text Editor -> All Languages
Line Numbers.



זוהי פעולת הדיבאג הבסיסית. אנו רצים בצורה סדרתית, שורה אחר שורה, ובודקים משתנים המעניינים אותנו.
המשך בלחיצות על F10 עד שהתוכנית תסתיים.

נקודות עצירה (BreakPoints)

כמובן שבשיטת דיבאג זו יש בעיה מסוימת. יש חלקים מהתוכנית שאנו בטוחים שעובדים כשורה וגם לפעמים התוכנית גדולה מאוד ורוצים לבדוק רק חלק ממנה.

לשם כך המציאו נקודות עצירה. נקודות עצירה הן נקודות בהן עוצרת הריצה (מן הסתם). כלומר מתבצעת ריצה רגילה עד הנקודה בה יש נקודת עצירה, וממנה מתבצע תהליך הדיבאג הרגיל.

הנחת נקודת עצירה מתבצעת ע"י לחיצה על המקש F9 בשורה הרצויה.

The screenshot shows the Visual Studio IDE with the following details:

- Code Editor:** Displays the `main()` function in `Main.cpp`. A red dot (breakpoint) is set on the first line of the `for` loop: `for (int nCounter = 0; nCounter <= nHalfLength.quot; nCounter++)`.
- Call Stack:** Shows the current execution stack, with the top frame being `Tichnut.exe!main() Line 41`.
- Breakpoints:** A window at the bottom shows the current breakpoint is active on the selected line.

ע"י לחיצה עם העכבר
באזור זה ניתן
לקבוע/לבטל נקודת
עצירה בשורה זו

וודא כי סיימת את הליך הדיבאג הקודם.
קבע נקודת עצירה בשורה 44.
האם אתה מזהה את הנקודה האדומה שנוספה לצד התוכנית.



ניתן לשים מספר נקודות עצירה בתוכנית. כלומר אנו בודקים את תוכניתנו עד נקודה מסוימת, מניחים שהחלק הבא תקין, ורצים ריצה נורמאלית עד נקודת העצירה הבאה.

קבע נקודת עצירה נוספת, בנקודת ההדפסה שנמצאת לאחר החישוב. (שורה 58)



הרצת דיבאג של התוכנית מתבצעת ע"י המקש F5 (Debug → Start Debugging). הרצה זו עוצרת כאשר מגיעים לנקודת העצירה.

הרץ את התוכנית עד לנקודת העצירה.



כעת העף מבט בחלון ה-Autos. זהו החלון שיושב הפינה השמאלית התחתונה של המסך. הוא מכיל כל פעם (לפי השורה שבה אנו נמצאים) את הערכים של המשתנים הרלוונטיים לפקודה הנוכחית. אפשר להגיע לחלון ה-Autos בעזרת התפריט Debug->Windows->Autos או ע"י לחיצה על Ctrl+Alt+V ואז A.

המשך להתקדם מספר פעמים בלולאה בעזרת המקש F10. ראה כיצד משתנים ערכי המשתנים שבחלון ה-Autos.
עבור לנקודת העצירה הבאה באמצעות המקש F5. אם לא היית קובע נקודת עצירה נוספת היית ממשיך בריצה רגילה עד סוף התוכנית.



איך מסירים נקודת עצירה? עומדים על השורה שבה יש נקודת עצירה ולוחצים על המקש F9. בנוסף, ניתן ללחוץ על נקודת העצירה וכך לבטל אותה.

הסר את נקודת העצירה שהוספת בתוכנית.



מעקב

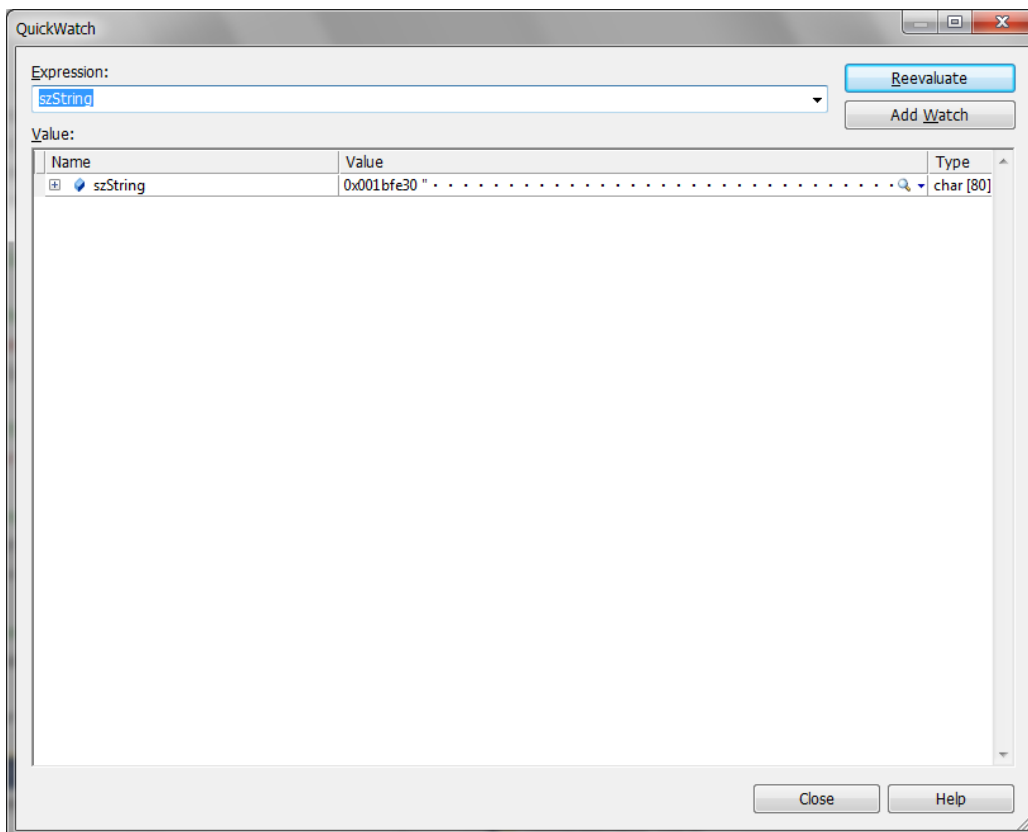
לפעמים מטריד למדי להעביר את סמן העכבר על משתנה כדי לראות את ערכו. נכון, ניתן להשתמש בחלון ה-Autos, אבל מה אם אנחנו רוצים לעקוב אחרי משתנה ספציפי לאורך כל התוכנית? לשם כך בדיוק יש את חלון ה-Watch. הוא מאפשר לעקוב אחר משתנים (וביטויים המכילים משתנים) לאורך כל התוכנית.

מעקב אחר משתנה מתבצע בדרך הבאה :

סמן את המשתנה, ולחץ על כפתור העכבר הימני. בחר את האופציה QuickWatch.



מופיע חלון הנקרא QuickWatch. שימושיות חלון זה מועטה יחסית.



סמן משתנה כלשהו ולחץ על הכפתור Add Watch.



פעולה זו תוסיף את המשתנה שלנו לחלון ה-Watch. ערכם של המשתנים בחלון זה מוצג לאורך כל התוכנית, ומתעדכן לאחר כל פקודה (ניתן לפתוח את חלון ה-Watch בעזרת לחיצה על Ctrl+D+W).

המשך בתהליך הדיבאג.
הוסף את המשתנה nCurrentDigit לחלון ה-Watch. המשך לרוץ בתוכנית עד סופה. שים לב לערכו של nInverted.



טיפים

- לחיצה על כפתור העכבר השמאלי במסך ה-Watch מאפשרת להוסיף משתנה למעקב פשוט ע"י כתיבת שמו בשדה. כמו כן ניתן לסמן שם של משתנה ולגרור אותו לחלון ה-Watch. נסה זאת!
- לחץ על ערכו של משתנה בחלון ה-Autos או בחלון ה-Watch. ניתן לשנות ערך זה כדי "להתערב" בביצוע הרגיל של התוכנית.
- נמאס לך מלדבג? אתה רוצה להפסיק את תהליך הדיבאג? בחר באופציה Debug → Stop Debugging או SHIFT + F5.
- אתה רוצה להפריד בין המסכים? גרור את חלון ה-Watch לצד הימני בו נמצאים חלונות ה-Call Stack, Breakpoints וכו'. הרחב והקטן את החלונות, מתח אותם לגובה, וצמצם אותם. העסק גמיש.
- לנוחיותך, קיימים 4 חלונות Watch. כדי לגשת אליהם לחץ מס' Ctrl+Alt+W+.

כיצד לבנות תהליך דיבאג

"...מיומנות הדיבאג היא אחת המיומנויות החשובות של תוכניתן טוב..." (ביל גייטס)

ללמוד כיצד לדבג נכון תוכנית הוא דבר שלוקח זמן ודורש ניסיון. בתור התחלה יש לזהות את קטע הקוד השגוי. כדי לעשות זאת יש לבנות את התוכנית בחלקים - לכתוב קטע קוד קטן להריץ ולבדוק. לכתוב עוד קטע קוד - להריץ ולבדוק. ככה ניתן מראש לצמצם את קטע התוכנית שאותו יש לדבג בבת אחת.

הדבר הכי חשוב הוא תכנון מראש. דע בדיוק מה קטע הקוד אמור לעשות. תכנן מראש מה אמור להיות ערכם של משתנים מסוימים עוד לפני שאתה מריץ את התוכנית.

לאחר שהגעת אל הקטע הבעייתי החל לעקוב אחרי המשתנים שבו. אולי הם לא מכילים ערכים נכונים? אולי הם לא משתנים בזמן המתאים? אולי התנאי בלולאה שגוי?

בד"כ לאחר שענית על שאלות אלה תפתור את הבאג. אם עדיין לא, סימן שהאלגוריתם עצמו שגוי. נסה לחשוב עליו שוב ולבדוק אותו מול העיצוב.

הדיבאגר אינו פותר בעיות בדרך קסם. שימוש בדיבאגר ללא תכנון מראש הוא חסר ערך.



שים לב

סיכום

בפרק זה למדת את מיומנות הדיבאג - השלב הרביעי בכתיבת תוכנית - החל מהמעבר הראשוני על כל שורת קוד שכתבת, וכלה בנקודות עצירה וצפייה סלקטיבית במשתנים.

למרות הקלות היחסית של השימוש בדיבאגר, עליך עדיין להתמקצע עם התפיסה המחשבתית שהוא מציג. אל תתבייש, אם יש לך בעיה בתוכנית, גש למדריך התורן ושאל אותו כיצד להתחיל לדבג. בהצלחה.

Debug מתקדם

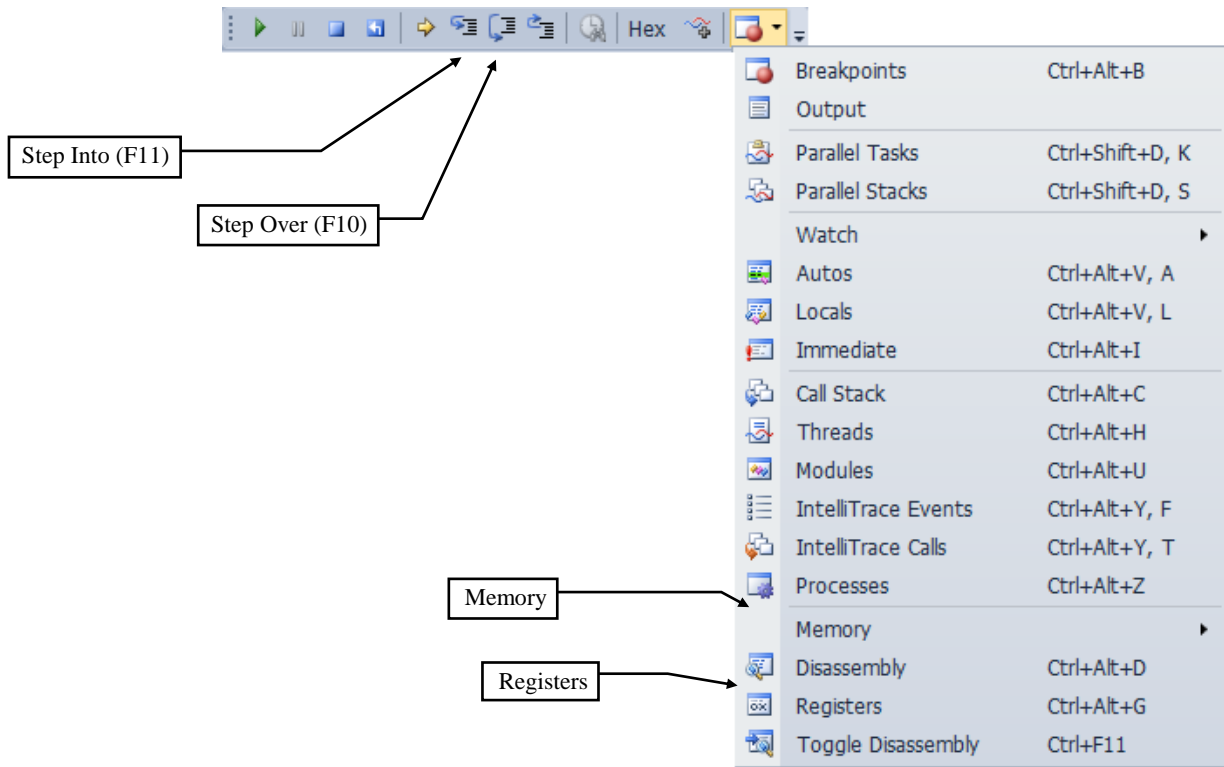
בסרגל כלים זה תמצא גם כפתורים שמאפשרים את כל הפעולות שהוסברו בסעיפים הקודמים.

על-מנת להציג את סרגל הכלים, יש לעמוד עם העכבר על איזור סרגלי הכלים וללחוץ על הכפתור הימני.

מהתפריט המוצג יש לבחור Debug

לאחר שלמדתם את עיקרי הדיבאג הגיע הזמן ללמוד כיצד לדבג שגרות, וכיצד להיכנס לעומק הזיכרון. הדבר הראשון הוא הכרת סרגל הכלים הבא. ודאו כי הוא מופיע באזור סרגלי הכלים. ניתן לגרור את הסרגל מאזור סרגלי הכלים אל שולחן העבודה.

לחיצה על הכפתור הימני של העבר תפתח את התפריט שהוזכר ב"השלמה אוטומטית" בעמוד 39. כעת ניתן לראות כמה מכפתורי הדיבאג גם בתפריט הזה (מרכז התפריט).



דיבוג שגרות

כמובן שחלוקה נכונה לשגרות היא דבר חשוב ביותר בעת כתיבת קוד. היא מאפשרת שימוש ביחידות תוכנה סגורות, בדוקות, וכך ניתן לחסוך זמן בכתיבה ובדיבאג. אולם מה קורה אם יש באג בשגרה?

כמובן שמצב זה אינו אמור לקרות, אבל גם באגים לא אמורים לקרות ולכן...

בכל מקרה הדבר הראשון שניתן לעשות הוא להיכנס לתוך השגרה בתהליך הדיבאג. כמו שעקבנו שורה אחר שורה בתוכנית בעזרת המקש F10 ניתן להיכנס לתוך כל שגרה ושגרה באמצעות המקש F11. מקש זה פועל כמו מקש ה-F10, אולם אם הפקודה הנוכחית הינה קריאה לשגרה, אנו נכנס לתוכה. פעולה זו נקראת בלעז Step-Into.

שים לב: אם תעמוד על שגרה שלא אתה כתבת (כמו strcpy), לא תמיד תכנס לסורס שלה. במקרה ונכנסת (שהרי איננו מעוניינים לדבג את strcpy), לחץ על Shift+F11. זה גורם ליציאה מהשגרה.

אם Source התוכנית כלל לא קיים במערכת, אזי יוצג קוד האסמבלר של השגרה. אם זה מעניין מישהו, תבדקו איך נראה הקוד של strcpy.

מפת התייחסויות

אם נלחץ על כפתור העכבר הימני מעל משתנה נקבל תפריט קיצור עם פעולות שכיחות עליו. תפריט זה כולל הצגה של ההגדרה שלו (Definition) ושל ההתייחסויות אליו (Reference). דבר זה אפשרי גם לשגרות.

לצורך ביצוע הפעולות הנ"ל, יש צורך בבנייה מיוחדת של הפרויקט הנעשית באופן אוטומטי. זכור, ללא ביצוע בנייה מחדש לא תוכל להשתמש באופציות אלה.

קריאות

בסרגל ה-Debug קיים כפתור המאפשר להציג בזמן הרצת התוכנית את ה-Call Stack (בד"כ מופיע כבר בזמן דיבאג בחלק התחתון של המסך, בצד ימין).

ה-Call Stack הינו חלון בו מוצגת רשימת של כל הקריאות לשגרות, עד לשגרה המורצת (למשל אם שגרה A קראה לשגרה B שקראה לשגרה C אותה בה אנו נמצאים כרגע, יופיעו ב-Call Stack שגרות A ו-B).

ה-Call Stack מאפשר לקבל תמונה על אופן זרימת התוכנית (שימושי בעיקר כאשר מדבגים שגרה הנקראת ממספר מקומות שונים), וכן לגשת במהירות לשגרות הקוראות (ע"י לחיצה כפולה על שמותיהן).

נספחים

Help

בנוסף לסביבת עבודה כללית ונוחה מספקת Microsoft גם עזרה מקיפה על C++ באמצעות ה MSDN . עזרה זו בנויה בצורת ספרים. במקום ספרים שתופסים מקום רב על המדף (ובנוסף לכך פוגעים באיכות הסביבה והם יקרים להפצה) יש קבצים בגודל אלפי מגה המכילים את הטקסט של ספרי העזרה.

עד גרסאות קודמות של ה- Visual Studio העזרה הייתה חלק נפרד מסביבת הפיתוח, שנפתחה בחלון משלה, ועבדה לפי דרישה (חיפוש לפי נושא או מילה).

ב Visual Studio .NET חלון העזרה העוצמתי (MSDN) קיים גם כן. אולם, הסביבה מאפשרת תמיכה ומערכת עזרה הרבה יותר חיה ודינאמית. כלי העזרה מתחלקים ל 2 חלקים עיקריים:

- ❖ חלון ה MSDN
- ❖ העזרה המשולבת בתוך סביבת העבודה
 - עזרה משולבת
 - עזרה דינמית (Dynamic Help)

חלון ה MSDN

כדי להשתמש בחלון ה MSDN הכנס לתיקיית icons ולחץ על הקישור לMSDN

כאן מוצגים כל הערכים שנמצאו התואמים את המילה המבוקשת. במידה ויש יותר ממקום אחד בו מופיעה המילה שחיפשת, יופיע מסך בו ניתן לבחור את הנושא המתאים. בחר את הערך המתאים לך והעזרה תופיע.

- במקרה וערך זה לא סייע לך בהבנת הפונקציה, בדוק את See Also (המופיע בסוף הדף). שם יש רשימה של נושאים רלוונטיים היכולים לסייע לך בהבנה מעמיקה יותר של החומר.
- טקסט אפור, או טקסט שמשומן בקו כחול מתחתיו, מאפשר הצגת נושאים נוספים הקשורים בחומר המבוקש.
- הגעת לנושא ואתה מעוניין לחזור לנושא הקודם בו היית לחץ על כפתור 'אחורה' בסרגל הכלים, או Backspace במקלדת.
- ניתן לדפדף בין עמודי הספר ע"י הכפתורים אחורה, קדימה, הבא וקודם. ניתן להעתיק מה-Help דוגמאות והגדרות שונות.