

## תוכן עניינים

2	מטרת התרגיל .....	
2	הנחיות כלליות .....	
2	המטלה – קווים כלליים .....	
3	הוראות פרטניות .....	
3	1. פתיחת חשבון ב- <b>github.com</b> .....	
5	2. פתיחת מאגר חדש .....	
6	3. שיתוף בן הזוג במאגר .....	
8	4. יצירת עותק מקומי של המאגר .....	
10	5. יצירת פתרון בתוך המאגר המקומי .....	
10	א. יצירת פרויקט .....	
11	ב. שימוש ב- <b>Snippets</b> .....	
11	ג. כתיבת קוד קצר .....	
12	ד. העברת הקוד לפונקציה באמצעות <b>Refactor</b> .....	
14	6. עדכון וסנכרון המאגרים .....	
14	א. אישור השינויים ( <b>ע"י השותף הראשון</b> ) .....	
15	ב. עדכון המאגר הראשי ( <b>ע"י השותף הראשון</b> ) .....	
16	ג. עדכון המאגר המקומי אצל השותף השני ( <b>ע"י השותף השני</b> ) .....	
17	ד. עדכון וסנכרון המאגר של השותף השני ( <b>ע"י השותף השני</b> ) .....	
17	ה. סנכרון המאגר של השותף הראשון ( <b>ע"י השותף הראשון</b> ) .....	
18	7. פעולות נוספות לתרגול עבודה בזוגות .....	
18	א. עבודה באותו קובץ, בשורות שונות .....	
19	ב. עבודה באותן שורות .....	
21	8. הגשה .....	
21	א. יצירת תג (Tag) .....	

## מטרת התרגיל

מטרת התרגיל הינה הכרות ראשונית עם :

- שפת C#
- סביבת הפיתוח Visual Studio 2019
- מערכת ניהול גרסאות Git
- שירות ענן של הארחת מערכת ניהול גרסאות מבוזרת github.com

רמת התרגיל – קל

## הנחיות כלליות

- ✓ ההגשה בזוגות בלבד (**חשוב**) : הקפידו כבר בתרגיל המבוא לעבוד עם שותף שאיתו תבצעו את יתר התרגילים ואת הפרויקט!!
- ✓ עבודה על התרגיל יכול להיעשות רק לאחר ששני השותפים לזוג ביצעו את הנחיות התקנה/עדכון של Visual Studio 2019, גם הסטודנטים שחושבים שהתכנה הזו כבר מותקנת אצלם – חובה לבצע את העדכון ע"פ קובץ ההנחיות שבאתר הקורס.
- נ.ב. **אירעו כבר מקרים כאשר סטודנטים נתקעו באמצע התרגיל בגלל אי ביצוע העדכון.**
- ✓ הגדרת ההרשאה של המאגר שייפתח ב-github.com בעבור התרגיל (ויתר התרגילים והפרויקט) חייבת להיות כ"פרטית", החזקת המאגר עם הרשאה ציבורית אפילו לפרק זמן קצר תמנע קבלת ציון לתרגיל ובכך גם לקורס.
- ✓ יש לסנכרן את המאגרים המקומיים במחשבים של שני השותפים לזוג עם המאגר ב-github.com.
- ✓ יש להוסיף את המרצה ואת בודק התרגילים (שמות משתמש ב-github.com יינתנו ע"י המרצה) כשותפים לפרויקט ב-github.com כבר בתחילת העבודה.
- ✓ אסור לבצע שינויים בפרויקט של התרגיל ב-github.com לאחר הגשתו. שינויים כאלה ימנעו קבלת ציון לתרגיל.
- ✓ כל התרגילים יבוצעו כפרויקטים באותו פתרון (Solution) ובאותו מאגר ב-Tab. שם ה-Solution והמאגר יהיה זהה והפורמט שלו יהיה : **dotNet5782\_ xxxx\_ yyyy**, כאשר :
- dotNet5782 – שמו המקוצר של הקורס ומספר השנה העברית בספרות לועזיות
- xxxx ו- yyyy – 4 ספרות אחרונות של תעודת הזהות של שני הסטודנטים
- ✓ שם הפרויקט עבור התרגיל יהיה **Targil0**.
- נ.ב. אי הקפדה על פורמטים אלה תמנע קבלת ציון לתרגיל או לכל התרגילים בהתאם.
- ✓ הגשת התרגילים תבצע ע"י הזנת הקישור לתאג במאגר (Tag) ב-github.com שנוצר ע"י הסטודנטים עבור ההגשה.

## המטלה – קווים כלליים

1. יצירת חשבון אישי ב-github.com ע"י שני הסטודנטים
2. יצירת מאגר **פרטי** עם שם לפי הפורמט ב-github.com
3. שיתוף המאגר עם השותף להגשת התרגילים, עם המרצה ועם הבודק
4. יצירת Solution ב-Visual Studio 2019 עם פרויקט עבור תרגיל 0 בשמות לפי הפורמט, וחיבורו למאגר – במחשבים של שני הסטודנטים
5. סינכרון ה-Solution של שני השותפים לזוג עם המאגר ב-github.com
6. עבודה על קובץ **README.md** :
  - a. הקובץ מגיע עם מספר שורות.
  - b. שני השותפים לזוג מוסיפים \ עורכים שורות שונות ומסנכרנים ביניהם.
  - c. שני השותפים לזוג עורכים באותה שורה ואחד ממזג.
  - d. שני השותפים לזוג עורכים באותה שורה, והשני ממזג.
7. הגשה

**שימו לב :** עבור כל שלב, יסומן במרקר צהוב מי מהשותפים לזוג אמור לבצע אותו.

ההוראות כתובות בלשון רבים והן פונות לזכר או נקבה כאחד.

## הוראות פרטניות

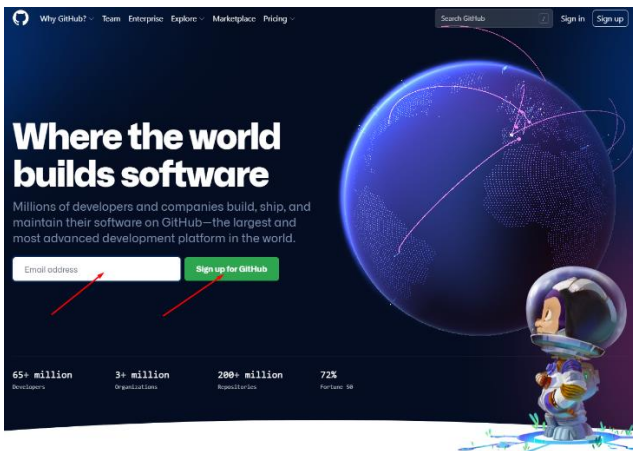
### 1. פתיחת חשבון ב-github.com

פרק זה יתבצע בנפרד על ידי כל אחד משני השותפים לזוג

0. אם יש לכם כבר חשבון ב-github.com, ניתן לדלג לפרק הבא.

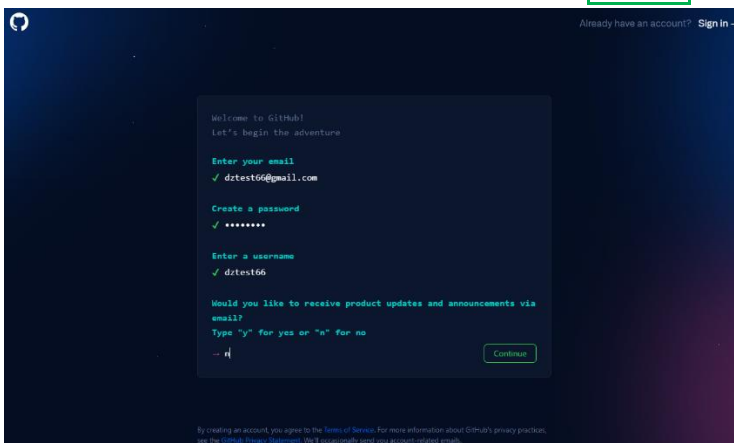
1. היכנסו לאתר בכתובת <https://github.com>

2. דף הבית של האתר נראה כך (נכון לתשרי תשפ"ב):



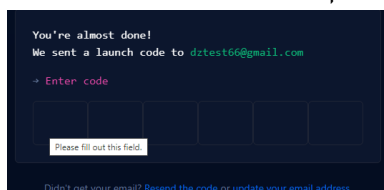
3. הכניסו את כתובת דוא"ל בדף הנ"ל (אם תשתמשו בכתובת דוא"ל שלכם מהמכון – עם הדומיין g.jct.ac.il, אפשר לקבל הטבות לסטודנטים) ותלחצו על הכפתור הירוק **Sign up for GitHub**

במסך הבא לחצו על הכפתור **Continue**, ובחרו סיסמא לפי ההוראות, עוד פעם **Continue**, בחרו שם המשתמש שלכם בשיירות github.com ולחצו עוד פעם **Continue**. בשאלה הבאה תקלידו "y" אם תרצו לקבל עדכונים ההודעות מהאתר לדוא"ל שלכם או "n" אם לאו, ועוד פעם **Continue**.



4. בהמשך, צריך לפתור פאזל (נכון לתשרי תשפ"ב), וללחוץ על כפתור **Create account**.

5. לאחר מכן תקבלו הודעה לדוא"ל עם קוד אמות – תקלידו את הקוד שהתקבל במסך הבא.



6. בשלב הבא, מוסרים קצת מידע על הידע שלכם והמטרות לפתיחת החשבון. בדרך כלל תרצו לבחור לסמן את המשבצות הבאות:

The screenshot shows the GitHub sign-up process. On the left, under 'What specific features are you interested in using?', there are several categories with checkboxes: Collaborative coding, Automation and CI/CD, Security, Client Apps, Project Management, Team Administration, and Community. On the right, under 'How many team members will be working with you?', there are buttons for 'Just me', '2 - 5' (selected), '5 - 10', '10 - 20', '20 - 50', and '50+'. Below that, under 'Are you a student or teacher?', there are buttons for 'Student' (selected) and 'Teacher'. A 'Continue' button is at the bottom right.

ולבסוף תלחצו על הכפתור **Apply for your GitHub student benefits**

The screenshot shows the GitHub Student Developer Pack page. It has a dark blue background with white text. The main heading is 'Learn to ship software like a pro.' Below it, it says 'GitHub gives students free access to the best developer tools so they can learn by doing.' There are two columns of benefits. The left column is titled 'Free' and lists: Unlimited public/private repositories, 2,000 Actions minutes/month, 500MB of Packages storage, and Community support. The right column is titled 'Get additional student benefits' and lists: GitHub Pro (Protect your branches, Draft pull requests, Pages and Wikis, 3,000 CI/CD minutes/month, 2GB of Packages storage, Web-based support), GitHub Student Developer Pack (Free access to the industry's best developer tools), and GitHub Campus Expert training. At the bottom, there is a button that says 'Apply for your GitHub student benefits'.

7. בזמנכם החופשי ולפי רצונכם תוכלו לסייר בדפי **github.com** עבור סטודנטים (**education.github.com**) ולהשתמש בכלים המוצעים שם. לא כל כך רלוונטי לפרויקט שלנו. בינתיים חזרו לאתר הראשי **github.com** אבל הפעם לחצו על **Sign in**:



8. הקלידו את שם המשתמש והסיסמה שבחרתם קודם:

The screenshot shows the GitHub sign-in form. It has a dark blue background with white text. At the top, it says 'Sign in to GitHub'. Below that, there are two input fields: 'Username or email address' and 'Password'. The 'Username or email address' field contains the text 'dztst6'. The 'Password' field contains a series of asterisks. To the right of the password field, there is a link that says 'Forgot password?'. Below the input fields, there is a green button that says 'Sign in'. At the bottom, there is a link that says 'New to GitHub? Create an account.'.

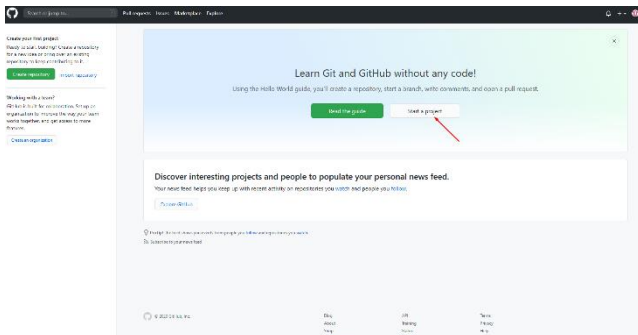
9. מגיעים לדף שבו מציעים ליצור מאגר (**Repository**) חדש. נמשיך לפרק הבא.

פרק זה יתבצע רק על ידי אחד מהשותפים לזוג - בפרק 3 משתפים את המאגר עם השותף השני

**הבהרה :** במסמך זה השתמשנו במילה "מאגר" שמכונה גם ריפו, ריפוזיטורי, **Repo, Repository**.

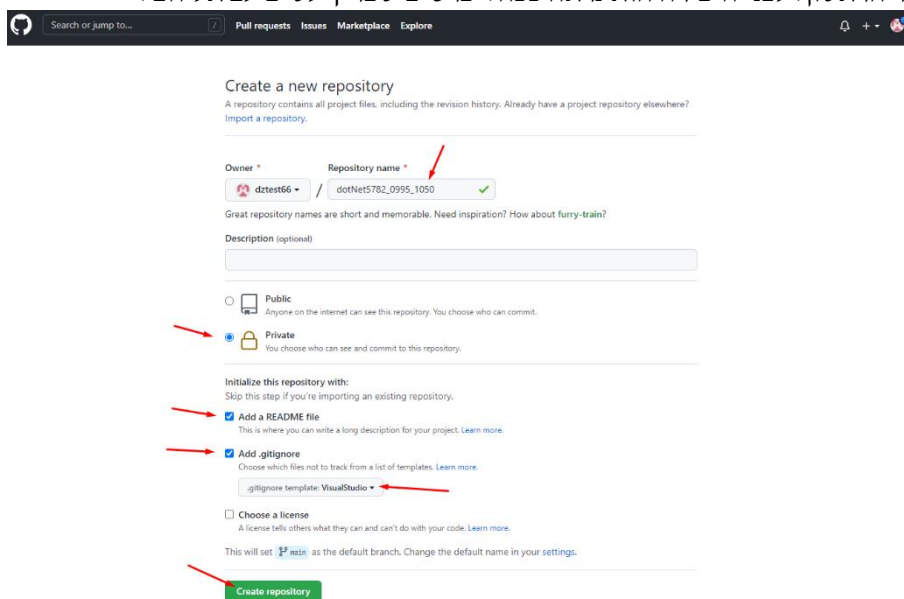
על מנת לחסוך עבודה חוזרת על השלב הזה, המדריך הזה מיועד ליצירת מאגר בודד לכל הקורס, כאשר המאגר יכיל את כל התרגילים ואת כל הפרויקטים. כמובן, **לאחר סיום** הקורס וקבלת ציון, ניתן לפתוח מאגר נוסף ולהעלות אליו רק את הפרויקט הסופי, להפוך אותו לציבורי, ולהוסיף קישור בקו"ח בחלק שמתאר את הפרויקט.

1. תיכנסו לאתר בכתובת <https://github.com> עם החשבון שפתחתם בפרק 1, (במידה ואתם כבר באתר – תוכלו לדלג לסעיף 3)
2. הדף הראשי אחרי שנכנסים נראה כך :



10. לחצו על כפתור **Start a project** או **Create Repository**. אפשר גם לחוץ על הכפתור הירוק ולקרוא את המדריך הבסיסי.

3. בחלון הבא תיצרו את המאגר הראשון. לפני היצירה הוא נראה ככה : פרטים שצריך לשים לב אליהם :



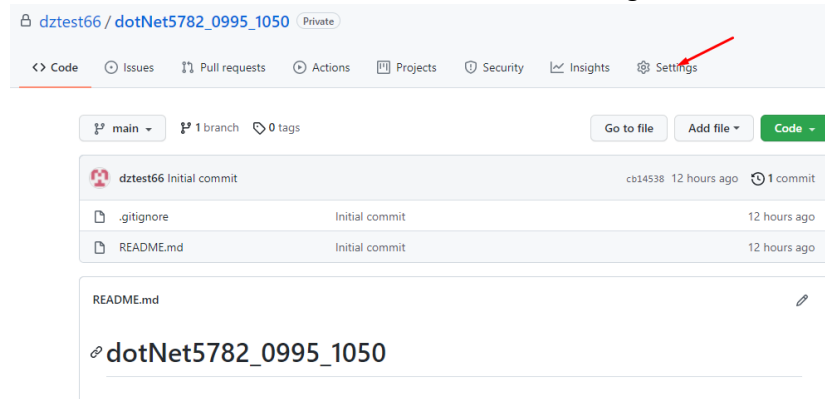
- א. הקלידו את שם המאגר לפי הדרישות כנ"ל.
  - ב. תיאור (**Description**) – רצוי לרשום, אפשר לדלג.
  - ג. המאגר **חייב** להיות מוגדר **פרטי (Private)**. במקרה של העתקה ממאגר ציבורי, גם בעל המאגר הציבורי ייחשב כאשם (לא עכברא גנב אלא חורא גנב...).
  - ד. סמנו את התיבה ליד "Add a README file".
  - ה. סמנו את התיבה ליד "Add .gitignore". בתיבת תפריט שתופיע לאחר הסימון בחרו ב- **Visual Studio**. בדרך הזו, גיט לא ישמור במאגר את כל הקבצים שהפנימיים של VS שהוא מייצר לצורך הניהול הפנימי של הפרויקט, אלא רק את קבצי הקוד הנצרכים.
  - ו. **אל תסמנו** את התיבה ליד **Choose a license** (בחר רישיון).
4. לחצו על הכפתור הירוק **Create repository**.

### 3. שיתוף בן הזוג במאגר

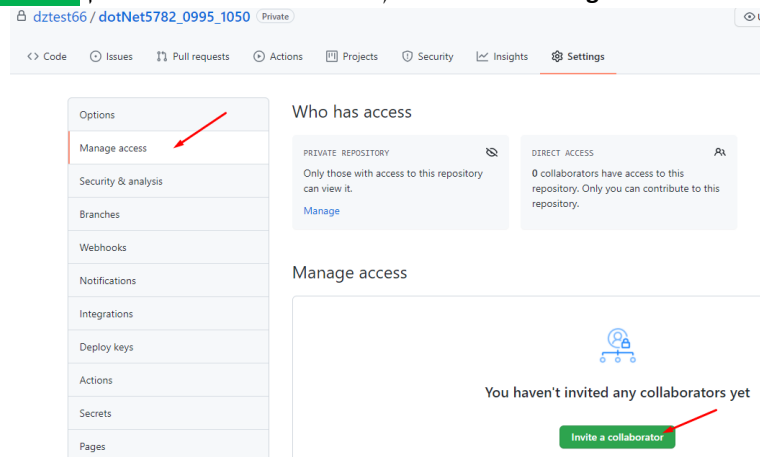
פרק זה יתבצע על ידי אותו הסטודנט שביצע את פרק 2

אחרי פרק 2, יש לנו מאגר פרטי, ואנחנו רוצים לעבוד עליו עם עוד מישהו. לצורך העניין נניח שהוא כבר פתח לעצמו חשבון ב-github.com (לפי ההוראות בפרק 1).

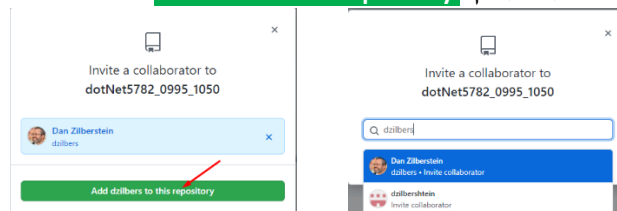
1. כנסו למאגר באחת משתי הדרכים:  
א. כנסו לעמוד הראשי של האתר, בצד שמאל יש רשימת הקישורים למאגרים שלך. לחצו על הקישור המתאים.  
ב. הקלידו בשורת הכתובת של הדפדפן את הכתובת בצורה הבאה:  
`https://github.com/username/repo-name` כאשר נרשם שם חשבונך במקום username ושם המאגר במקום repo-name.  
2. בחרו בכרטיסיית Settings בסרגל הכרטיסיות.



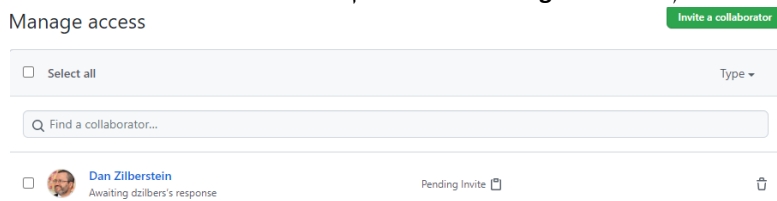
3. בחרו "Manage Access" בצד שמאל, ואז לחצו על הכפתור הירוק "Invite a collaborator" (הזמן משת"פ):



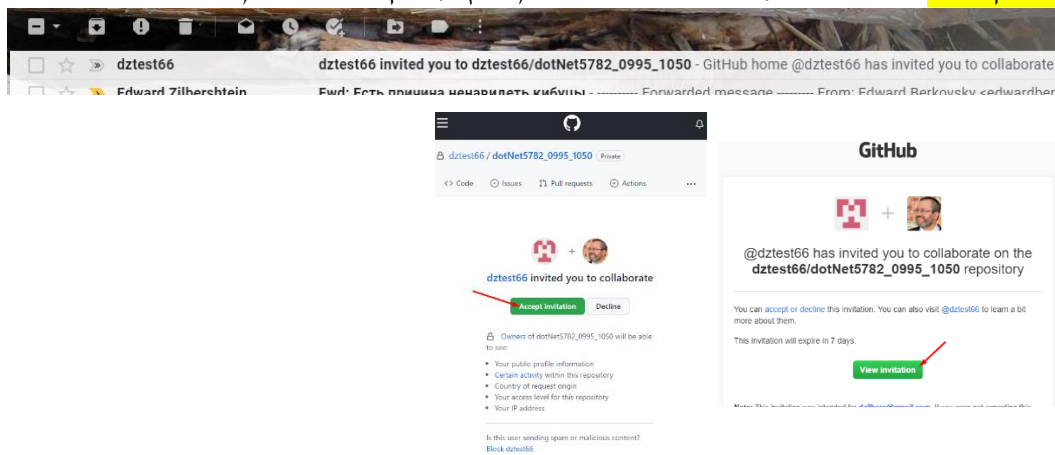
4. רשמו את שם המשתמש או השם או מספר טלפון של שותפכם לזוג (ובהמשך גם של המרצה, ושל הבודק), ולחצו על הכפתור הירוק "Add user to this repository".



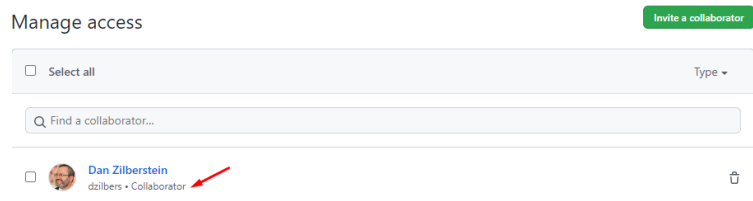
5. בשלב הזה, github.com אמורים לשלוח קישור של ההזמנה לדוא"ל של המשתמש שהוזמן.



6. **השותף השני** יפתח את הודעת ההזמנה בדוא"ל שלו, ילחץ על הקישור בהזמנה, ויאשר את ההזמנה.



לאחר האישור, בדף ניהול הגישה של השותף הראשון יופיע (לאחר רענון הדף):



7. במקרה של תקלה בקבלת הדוא"ל - עד שהמשתמש שהוזמן יאשר את ההזמנה, ההזמנה מחכה, וניתן להעתיק את "קישור לאישור ההזמנה", ולשלוח אותה עצמאית בכל אמצעי אחר למשתמש שהוזמן.
8. מהרגע שההזמנה אושרה, שני השותפים לזוג יכולים לעבוד מול המאגר ישירות. תדאגו לוודא ששותפכם מאשר את ההזמנה ואם לאו, זה סימן שיייתכן והוא לא מעניין להשתתף בקורס וכדאי לכם לבדוק אפשרות להחליפו. אם ההזמנה אינה מאושרת תוך זמן סביר, ההזמנה יכולה להפוך ללא תקינה (בלחיצת אישור ההזמנה מקבלים תשובה שההזמנה לא נמצאה). במצב הזה יהיה צורך שהמזמין ימחק את ההזמנה הישנה ויזמין את השותף מחדש.
9. כעת, הזמינו באותה הדרך את המרצה ואת בודק התרגילים להיות שותפים למאגר שלכם.

**סיכום ההוראות עד לשלב זה:** יש לנו שני חשבונות ב-[github.com](https://github.com), מאגר בודד עם תוכן בסיסי (קובץ **README.md** והגדרות שמתאימות לעבודה עם VS), משני החשבונות יש גישה למאגר (לא משנה מי מהשותפים מחזיק את המאגר, מלבד היכולת "לזרוק" את השני על ידי ביטול ההזמנה).

על מנת לאפשר בדיקה של התרגילים, יש לשתף גם את מרצה ואת בודק הקבוצה בה אתם לומדים.

#### 4. יצירת עותק מקומי של המאגר

פרק זה יתבצע בנפרד (ובמקביל) על ידי כל אחד מהשותפים לזוג

המטרה של השלב הזה, היא ליצור בתוך המחשב של הסטודנט מאגר מקומי, שהתוכן שלו יהיה זהה לתוכן המאגר ב-github.com, כאשר מפעם לפעם נדאג לבצע עדכון של התוכן בין המאגרים.

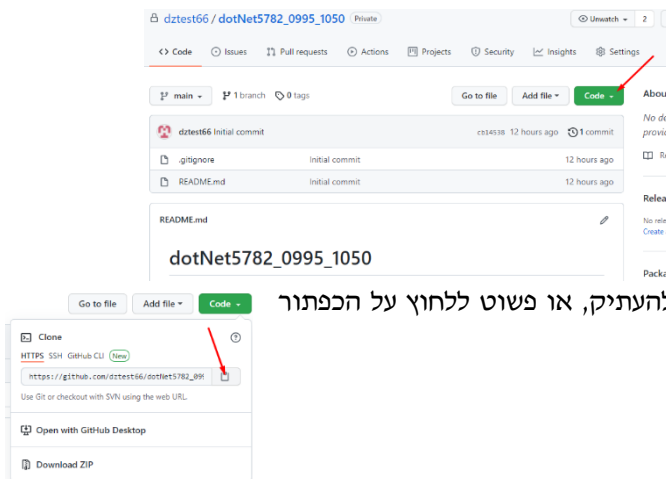
בהמשך עבור כל תרגיל בהתאם, נדאג כל פעם שאנחנו יוצרים פרויקט ליצור אותו באותו הפתרון (Solution) שנפתח כעת ונקשור אותו למאגר שפתחנו קודם. לכן כשנעדכן את המאגר מול github.com – נמשוך (pull) מ-github.com את השינויים שהשותף שלנו לזוג עשה, ונדחוף (push) ל-github.com את השינויים שעשינו.

המדריך כאן נכתב עבור VS2019, אחרי שאותחלו בו הגדרות ברירת מחדל עבור C#.

**שים לב שיש להשתמש בגרסה האחרונה ביותר שהותקנה לפי ההנחיות באתר הקורס**, וייתכן שלצורך כך תצטרך לעדכן את גרסת VS שיש בידך. להנחיות להתקנה ועדכון ראה במסמך ההנחיות שנמצא באתר הקורס.

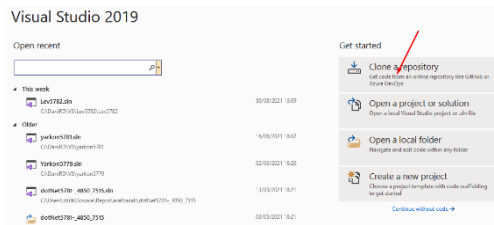
1. היכנסו בדפדפן לאתר github.com ופתחו את המאגר הרצוי.

2. לחצו על הכפתור הירוק **Code**:



3. תופיע כתובת, אפשר לסמן אותה ולהעתיק, או פשוט ללחוץ על הכפתור "העתק":

הכתובת בדרך כלל תהיה בפורמט <https://github.com/username/repo-name.git>.



עכשיו נעבור לויזואל סטודיו.

4. פתחו (הריצו) את תכנת ה-Visual Studio.

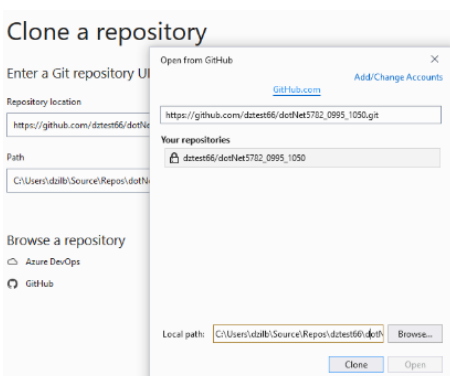
5. בחלון הפתיחה, ישנן מספר אפשרויות. בחרו באפשרות לשבט (להעתיק) מאגר קיים, בצד ימין למעלה:

6. בחלון הבא, כתבו את הכתובת המאגר, ואיפה במחשב אנחנו רוצים את העותק שלו.

**שימו לב:** הכתובת ב-local path צריכה להיות תיקייה שעוד לא קיימת – כדאי שתהיה במקום שאתם מרכזים את כל חומרי הקורס שלכם. כדאי שלא תהיה עברית בנתיב (path) של התיקיה. שם החלק האחרון של תיקיית המאגר יהיה כשם המאגר. את הקישור למאגר נעתיק ע"י הדבק לאחר שעשינו העתק בשלב הקודם.

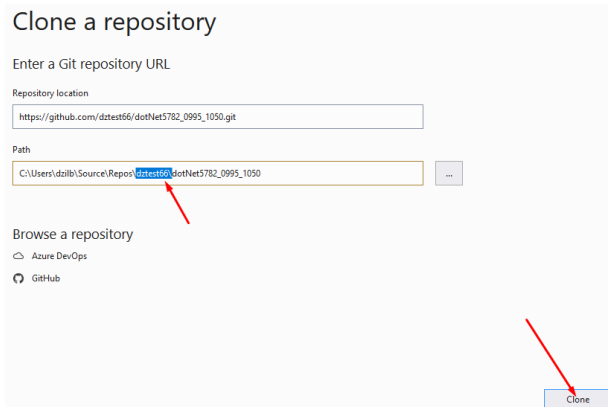
**נ.ב.1** תיקיית המאגר **צריכה להיות** תחת תתי-תיקיה Sources/Repos/ שבתיקיית המשתמש – **כלא יחד עם** שם החשבון ב-github.com. בדרך כלל הנתיב המלא לתיקיה יהיה כדלקמן (שם המאגר רק פעם אחת בנתיב):

**C:\Users\user-name\Source\Repos\account-name\dotNet5782\_xxxx\_yyyy**



**נ.ב.2** ניתן ללחוץ על הכפתור GitHub למטה ולבחור ישירות את מאגר המקור (תצטרכו לבצע כניסה לחשבון מתוך VS), לאחר בחירת המאגר לחצו על הכפתור **Clone**:



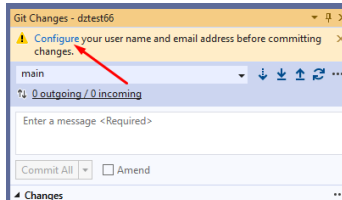
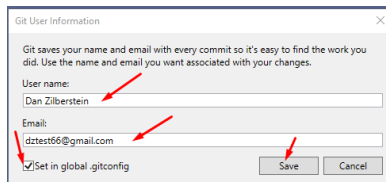


7. אם לא בחרתם לפתוח חלון בחירת מאגרים (מחשבון רשום) בנתיב (Path) תבדקו שמופיע שם החשבון שבו נפתח המאגר ב-github.com והוסיפוהו אם חסר (תוסיפו גם סלאש הפוך "/") ולאחר מכן לחצו על הכפתור **Clone** :

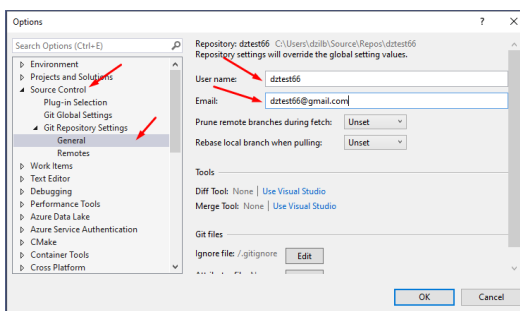
נ.ב. בפעם הראשונה תצטרכו לתת שם משתמש וסיסמא לאתר **github.com**.

8. ייפתח חלון של **VS**, אבל בגלל שהמאגר לא מכיל פתרון של **VS**, הוא יהיה כמעט ריק.

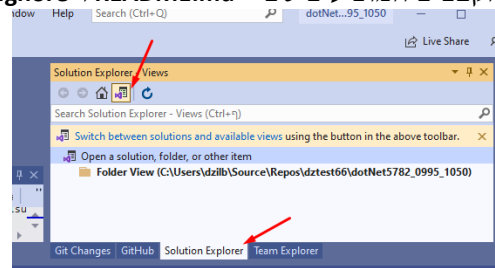
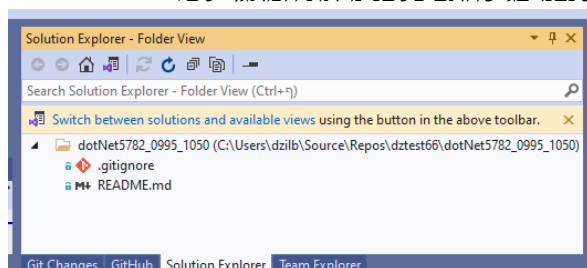
9. בחר לעדכן את השם ואת כתובת הדוא"ל שלכם עבור רישום ב-**commit** בהמשך :



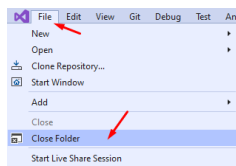
10. בפעם הראשונה גם מומלץ לעבור על הגדרות **git** של **VS** ולעדכן איפה שצריך (תפריט **Tools** ושם **Options**, בחלון שנפתח תרחיבו בצד שמאל את **Source Control**). למשל :



11. תעברו ללשונית **Solution Explorer** ותעברו לתצוגת תיקייה (**Folder View**), תראו את התוכן הבא. שימו לב על שני הקבצים המופיעים שם – **README.md** ו-**gitignore**. שנבנו בגיטהאב כשבנינו את המאגר שם.



12. בחרו בתפריט הראשית **File => Close folder** למסך הפתיחה של **VS**.

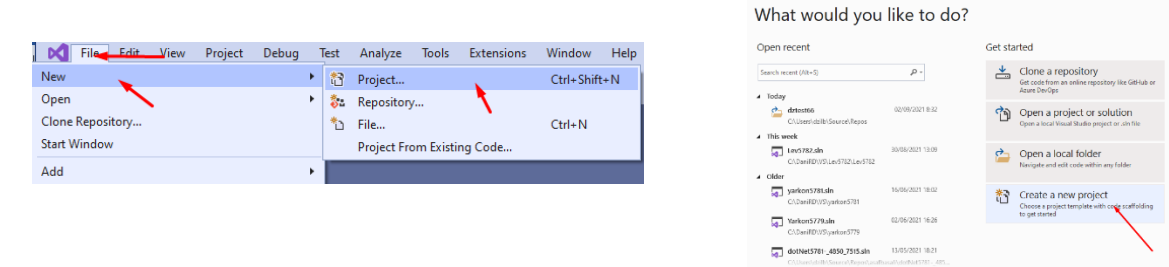


## 5. יצירת פתרון בתוך המאגר המקומי

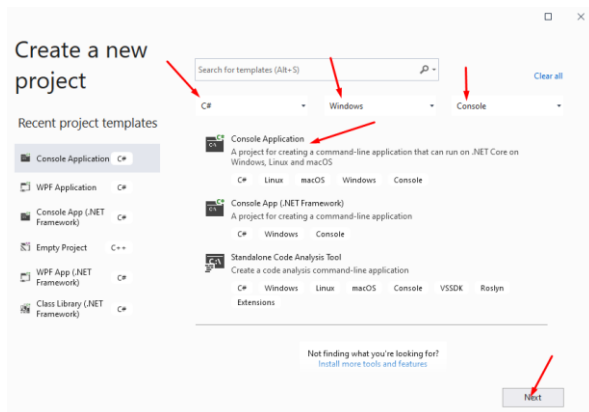
פרק זה יבוצע על ידי אחד (בלבד) מהשותפים לזוג (נקרא לו השותף הראשון) במחשבו האישי, רצוי שבזמן ביצוע הפרק שני השותפים יהיו מחוברים ב-zoom או יישבו ביחד כאשר השותף הראשון משתף את המסך ונותן לשותפו זמן להכיל את הנעשה. חובה שלשני השותפים לזוג תהיה אותה גרסה של Visual Studio.

### א. יצירת פרויקט

צרו פרויקט חדש (Create a new Project או בתפריט <File=>New<=>Project...)

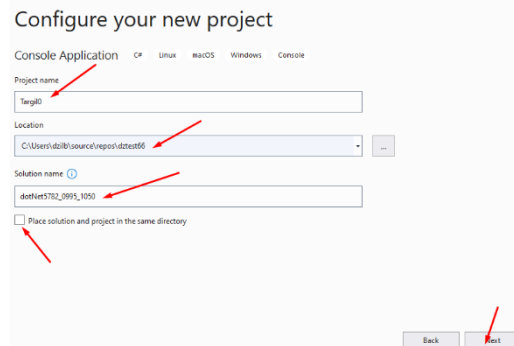


ובחרו בניית פרויקט מסוג מסוג Console Application :



נ.ב. בשום פנים ואופן לא לבחור סוג (.NET Framework) Console Application.

- קנפגו את הפרויקט כדלקמן, לפי הסדר כפי שמופיע כאן – קודם כל שם הפרויקט ורק לאחר מכן שם הפתרון! שם הפתרון (Solution) יהיה **בדיוק** כשם המאגר: **dotNet5782\_XXXX\_YYYY** (אפשר להעתיק משם תיקיית המאגר בסייר הקבצים, למשל)
- שם הפרויקט יהיה **Targil0**
- המיקום (Location) בתיקיית אב של המאגר – לעדכן בשדה! (ניתן לעבור לתיקיה בלחיצה על הכפתור "...").
- חשוב!! אין** לסמן בתיבת סימון **Place solution and project in the same directory**!



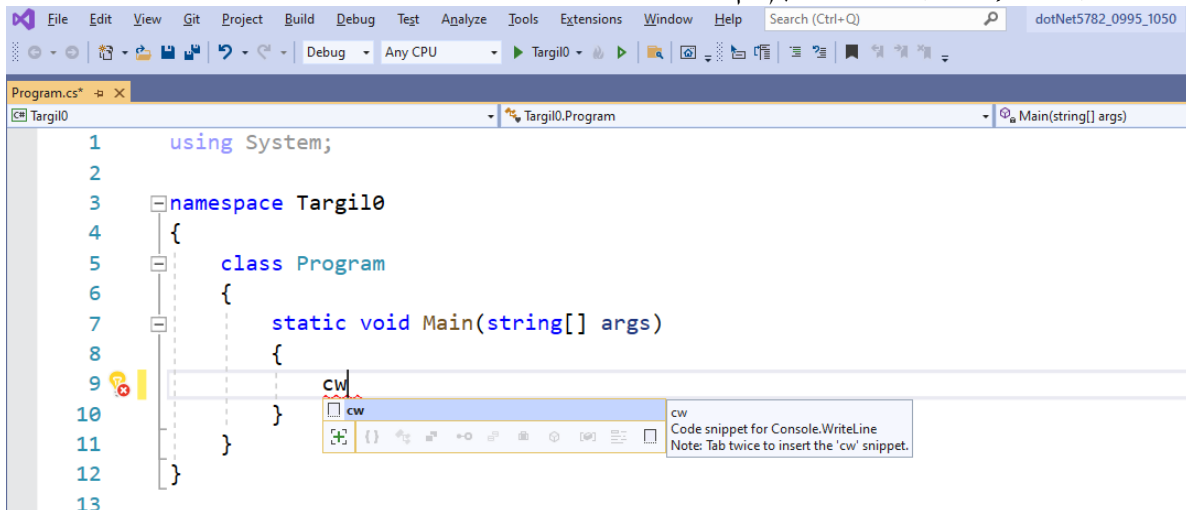
**שימו לב שתקיית המאגר המקומי המקורית היא גם תהיה תיקיית הפתרון ופרויקט ימוקם בתוך תיקיית המאגר!**  
נ.ב.1 בכל מקרה וודאו ששם התיקיה ב-Location יהיה שם תיקיית האב שכוללת בה את תיקיית המאגר המקומי שיצרתם בפרק 4 (הפרק הקודם). אי הקפדה על הכלל הזה תיצור בעיה שתדרוש מחיקה וביצוע הפרק מחדש.

לחצו על כפתור **Next**. בחלון הבא וודאו ש-Target Framework מוגדר כ-**NET 5.0 (Current)**. ולחצו על כפתור **Create**.

נ.ב.2 אם קיבלתם הודעת שגיאה לאחר לחיצה על הכפתור **Create** (כן – גם למיקרוסופט יש באגים) – סגרו את **Visual Studio** והפעילו אותו מחדש תוך שאתם בוחרים באפשרות **Create a new project** בחלון הראשון שנפתח (החלון הזה מודגם בתחילת הפרק הזה).

## ב. שימוש ב-Snippets

בתוכנית הראשית (שנפתחה אוטומטית), הקלידו **cw**



ציור של דף חתוך (משמאל לשורה הראשונה בתפריט שנפתח) אומר לנו שמדובר ב-snippet<sup>1</sup> ויש כאן השלמה אוטומטית לאחר לחיצה על **Tab** (יש לצאת מתפריט הקשר ואז לחוץ **Tab** או פשוט לחוץ פעמיים **Tab**). לאחר לחיצה על **Tab** נקבל את ההשלמה להדפסה למסך.

## ג. כתיבת קוד קצר

"Enter your name: "

נדפיס למסך את המחרוזת (אפשר למחוק את ה-"Line" מפקודת הפלט על מנת לבצע פלט באותה שורה ע"י הפעולה **Console.WriteLine(...)** :

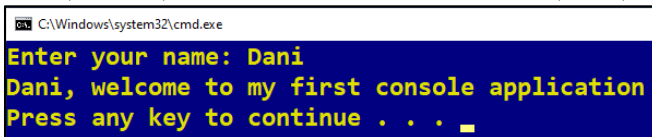
"{0}, welcome to my first console application"

נקלוט מהמשתמש את שמו בעזרת הפעולה **Console.ReadLine()** ונדפיס את ההודעה הבאה בהתאם :

(כמוכן, עליך להגדיר משתנה, להדפיס באמצעות שימוש ב-{0} כך שבהדפסה יופיע השם שהמשתמש הכניס וכו', אנו מניחים שכבר למדת מעט תכנות...) על מנת שהתוכנית לא תיסגר אלא רק לאחר שנלחץ על מקש כלשהו לסיום בסוף הפעולה הראשית אפשר להוסיף:

**Console.ReadKey();**

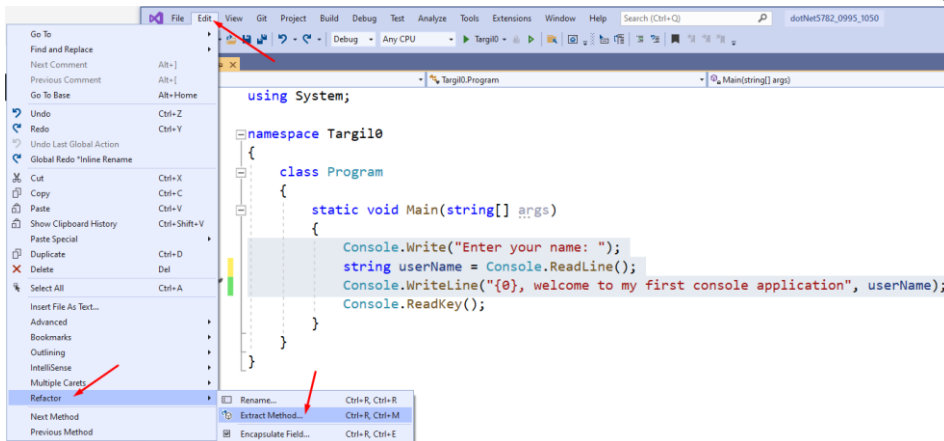
לחילופין, ניתן להריץ ע"י לחיצה למקשים **Ctrl-F5** (במקום רק **F5**). לאחר ההרצה ומילוי השם החלון יראה כך :



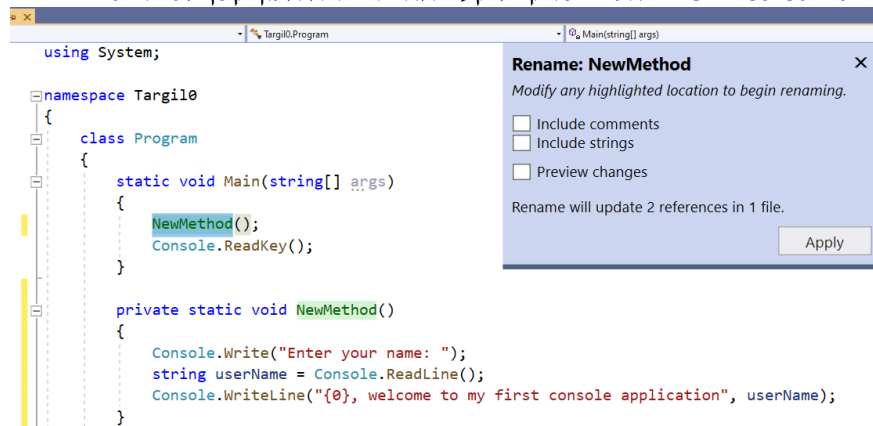
<sup>1</sup> רשימה חלקית של עוד snippet שונים ניתן לראות בקישור הבא (ניתן גם להיכנס לעורך snippet ע"י לחיצה **Ctrl-K** ואז **Ctrl-B**): <https://docs.microsoft.com/en-us/visualstudio/ide/visual-csharp-code-snippets?view=vs-2019>

#### ד. העברת הקוד לפונקציה באמצעות Refactor

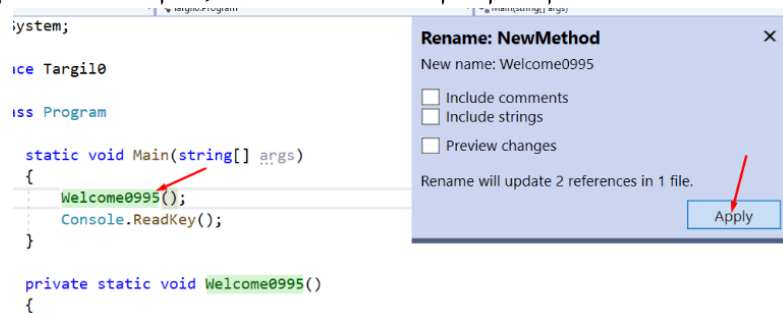
כעת נרצה שהקוד לא יהיה סתם בתוכנית הראשית אלא בתוך פונקציה נפרדת שנקרא לה מהתוכנית הראשית. נוכל להשתמש ב-refactor על מנת להעביר שורות בקוד לפונקציה נפרדת. נסמן את שלושת שורות הקוד שכתבנו ללא (`Console.ReadKey()`) ובתפריט העריכה נבחר תת-תפריט של Refactor ואז Extract Method:



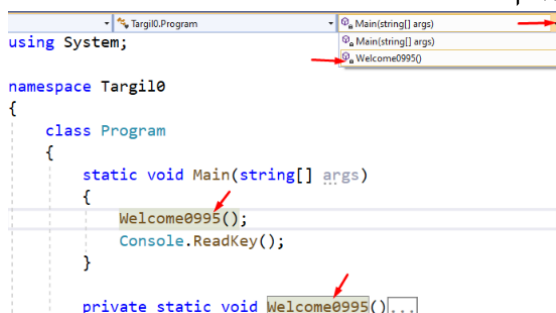
הקוד שסימננו יועבר לפונקציה בשם `NewMethod` – השם יסומן ברקע ייחודי וייפתח חלון קטן לשינוי שם:



נשנה את השם המסומן בתוך הקוד ל-`WelcomeXXXX` (במקום `XXXX` כמובן, ארבע הספרות האחרונות של מספר



ת"ז של השותף הראשון) ונלחץ על `Apply`. כעת הקוד יראה כך:



שימו לב שנוספה לנו פונקציה למחלקה `Program` וקריאה לה מהתוכנית הראשית שבמחלקה `Program`. שימו לב לתיבת הבחירה למעלה שניתן לדלג באמצעותה בקלות לקוד של כל פונקציה במחלקה.

**שאלה: מדוע הפונקציה הוגדרה כ-`static`?** (לא צריך לענות בהגשה אבל חשוב לדעת ולהבין את התשובה)

כעת, על מנת שבשעת שיתוף המאגר לא ניתקל בשלב הזה בבעיות בסיכרון בין שני השותפים, נשנה את הגדרת המחלקה למחלקה חלקית (בהמשך הקורס תלמדו עליה יותר לעומק). חשוב! הסיכרון יעבוד היטב גם במחלקה רגילה, ולא רק במחלקה חלקית. כך נעבוד ברוב המקרים בהמשך. אך כרגע לצורך הקלה על הלימוד נשתמש ברעיון החלקיות. לכן:

1. **השותף הראשון** יוסיף מילת מפתח **partial** לפני הגדרת המחלקה.
2. וכן יוסיף הגדרת פונקציה חלקית ללא מימוש בשם **WellcomeYYYY**, כאשר **YYYY** אלו ארבע הספרות האחרונות של מספר ת"ז של השותף השני.
3. וכן יפעיל את הפונקציה בתוכנית הראשית:

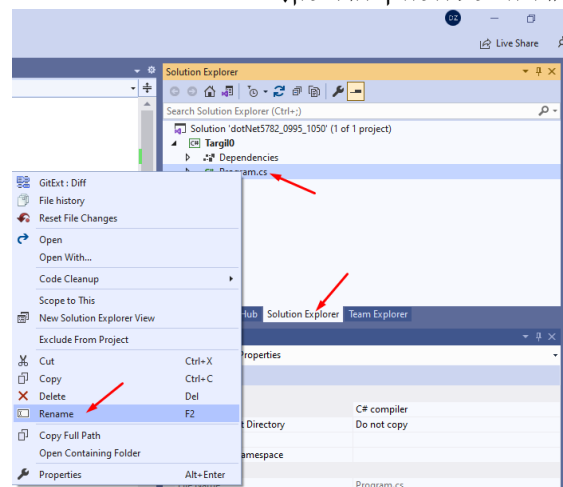
```
using System;

namespace Targil0
{
    partial class Program
    {
        static void Main(string[] args)
        {
            Welcome0995();
            Welcome1050();
            Console.ReadKey();
        }

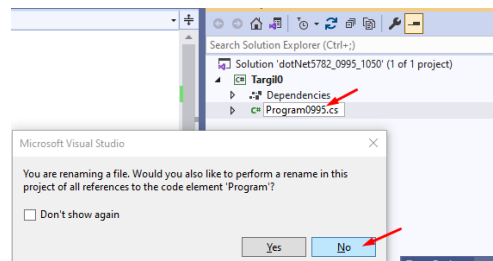
        static partial void Welcome1050();

        private static void Welcome0995() { ... }
    }
}
```

4. בהמשך לכך **השותף הראשון** ישנה את שם המודול (הקובץ) ל-**ProgramXXXX.cs** – ארבע הספרות האחרונות של השותף הראשון):



5. חשוב! בחלון הקופץ של אישור עדכון רוחבי יש לבחור באפשרות **No** (זה חשוב על מנת לא לשנות את שם המחלקה):



## 6. עדכון וסנכרון המאגרים

**אין להתחיל בביצוע הפרק הזה לפני שהשותף הראשון השלים את פרק 5.**  
**שלב א – ב, ה של פרק זה יבוצע על ידי השותף הראשון במחשבו האישי,**  
**שלב ג – ד של פרק זה יבוצע על ידי השותף השני במחשבו האישי,**

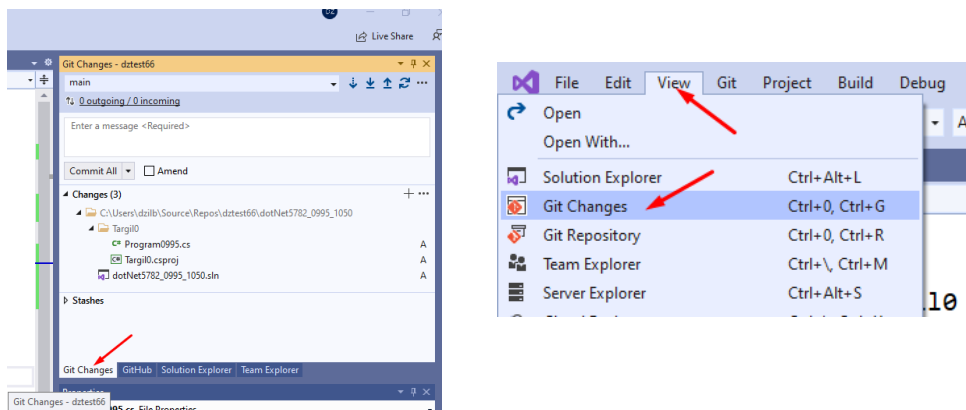
רצוי שבזמן ביצוע הפרק שני השותפים יהיו מחוברים ב-zoom כאשר השותפים משתפים את המסך לסירוגין.  
 סינכרון מורכב משני חלקים עיקריים: אישור השינויים שביצעתי לפרסום (commit), ועדכון מול המאגר הראשי.

סיכום קצר של הפעולות שאנו הולכים לבצע בשלב זה (ובכלל בכל עבודה משותפת) - כל פעם שמבצעים עדכון המאגר יש 3 שלבים בלשונית של **Git Changes**:

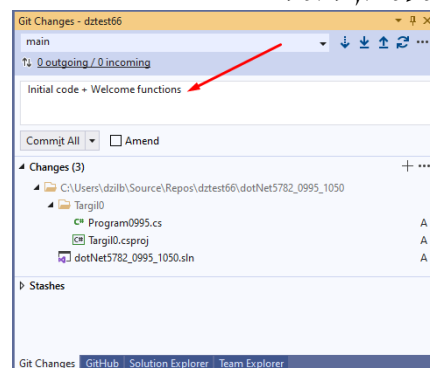
- כתיבת הערה **Commit All** (אישור השינויים שביצעתי במחשב שלי)
- כפתור **pull** (קבלת השינויים שאחרים ביצעו מהמאגר הראשי למחשב האישי ומיזוגם עם השינויים שביצעתי במחשב האישי שלי)
- כפתור **push** (עדכון השינויים במאגר הראשי ב **github.com**)
- נ.ב. במקום שתי הפעולות ב. ו-ג. ניתן לבצע פעולה אחת ב-**VS** ע"י כפתור **Sync**

### א. אישור השינויים (ע"י השותף הראשון)

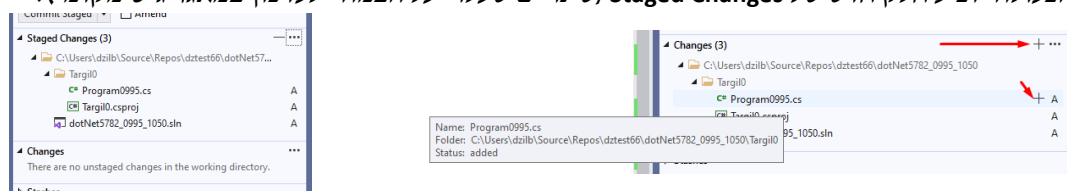
1. אם חסרה לשונית **Git Changes**, בחרו מהתפריט הראשית **View => Git Changes**:



- בפעם הראשונה, אם לא עדכנתם את הפרטים כמו שמופיע **בסעיף 9 בפרק 4**, נצטרך למסור פרטים שיתארו אותנו ויופיעו על כל "**commit**" שנעשה (כלומר, כל שורת קוד או קובץ שנוסח, יהיה רשום שאנחנו הוספנו אותו, עם הפרטים האלו).
- בשלב הזה יופיע פירוט כל השינויים שנעשו בתיקיית העבודה המקומית (לעומת המאגר בשרת שכלל רק את התיקייה הראשית, אנחנו הוספנו את ה-**Solution** ואת הפרויקט עם הקוד כנ"ל) – ראה בתמונה לעיל.
- בחלק של קלט טקסט (בתחילה מופיע שם באפור: \_\_\_\_\_) חובה לרשום הערה שתתאר את השינוי שעשינו, למשל:

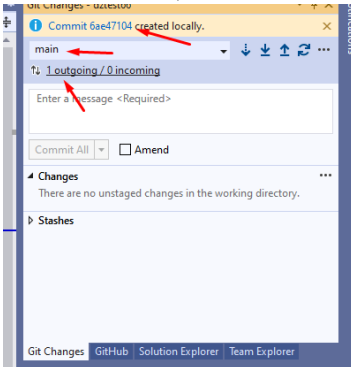


- בחלק עם השינויים (**Changes**) נלחץ על סימן ה"++" בצד ימין. בהמשך, כשתרצו לכלול בעדכון רק חלק מהשינויים שנעשו, תוכלו ללחוץ על סימן ה"--" מימין לקבצים ספציפיים (ראו את החץ הקטן בתמונה). לאחר הפעולה יופיע חלק חדש של **Staged Changes** (שינויים שעלו "על הבמה" לעדכון במאגר גיט מקומי):



בהמשך, אם טעיתם ותרצו "להוריד" קבצים מהבמה – חפשו סימני "--" בחלק של **Staged Changes**.

6. עכשיו על מנת לשלוח את השינויי שעלו על הבמה למאגר גיט המקומי נוכל ללחוץ על הכפתור **Commit Staged** שהחליף את **Commit All** שהיה שם קודם. לידיעתכם, אפשר היה לדלג על סעיף 5 וללחוץ ישירות על **Commit All** – אז כל השינויים היו "עולים על הבמה" והעדכון היה נשלח למאגר בלחיצה אחת. השלימו את הפעולה ע"י הלחיצה כפי שהוסבר. שימו לב שלמעלה הופיעה הודעה שהעדכון (**commit**) נבנה מקומית (בענף **main**) ועוד יופיעה שיש עדכון יוצא אחד (זאת אומרת יש עדכון במאגר מקומי שלא נשלח למאגר המרוחק – בגיטהאב):

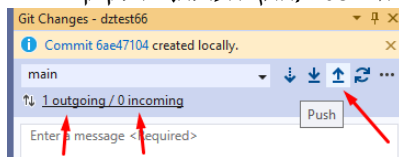


**נ.ב.1:** אם כפתור העדכון "אפור" ולא ניתן ללחוץ עליו – זה אומר שאין שינויים שאפשר לעדכן אתם את המאגר המקומי. זה יכול לקרות גם אם באמת אין שינויים וגם אם בטעות המאגר שלכם לא מקושר לתיקיית העבודה של הפרויקט בגלל שטעיתם או דילגתם על משהו באחד השלבים הקודמים. במצב הזה תרצו למחוק את התיקיה ולחזור לאחד השלבים הקודמים (לפי המצב שבו זיהיתם את הבעיה).

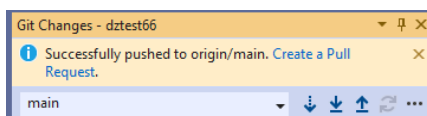
**נ.ב.2:** שימו לב שבכפתור העדכון ניתן לפתוח תפריט ולראות אפשרויות נוספות הכוללות "and Push" ו-"and Sync". מטרת הראשונה לבצע שליחה אוטומטית של העדכון במאגר המקומי למאגר המרוחק (מה שנעשה בנפרד בצעדים הבאים), והשנייה תכלול גם משיכה ומיזוג העדכונים החדשים מהמאגר המרוחק למאגר המקומי לפני ביצוע העדכון המקומי עם השינויים שלכם – נרחיב על זה בהמשך.

## ב. עדכון המאגר הראשי (ע"י השותף הראשון)

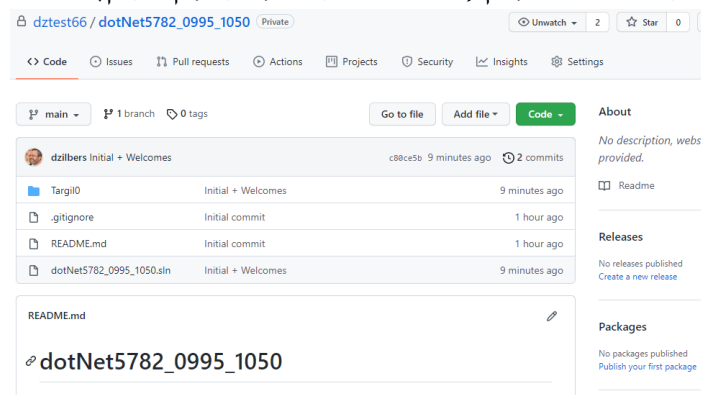
1. ניתן לבצע פעולת **fetch** ע"י כפתור של חץ מקוקו יורד על מנת לעדכן את המאגר המקומי לגבי עדכונים במאגר הרחוק שלא מופיעים במאגר המקומי. לחלופין יש לשאול את השותף האם שלח עדכונים לגיטהאב.
2. פעולת שליחת עדכונים מהמאגר המקומי למאגר המרוחק (בגיטהאב) תתבצע **רק** כאשר בחלון רואים **שיש עדכונים יוצאים ואין עדכונים נכנסים**. הפעולה תתבצע ע"י לחיצה על כפתור **Push** (החץ העולה) כדלקמן:



3. באזור הודעות גיט יופיעו ויתחלפו הפעולות הפנימיות המתבצעות ע"י ויזואל סטודיו בגיט לאחר השלמת פעולת השליחה (ה"דחיפה") תופיע הודעת הצלחה:



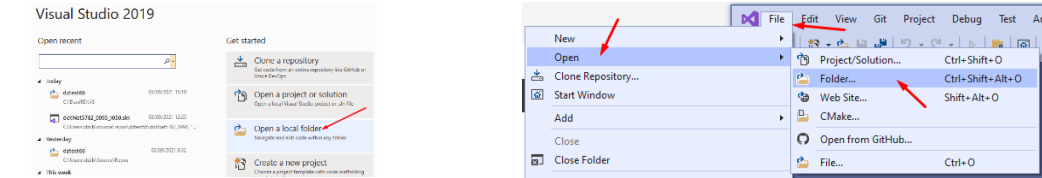
4. אם תקבלו הודעה על תקלה – כנראה שלא הקפדתם על אחד או יותר מהכללים שהזכרנו לעיל.
5. תפתחו את האתר של גיטהאב במאגר שלכם וראו שהוא התעדכן (אם היה פתוח – תעשו ריענון לחלון):



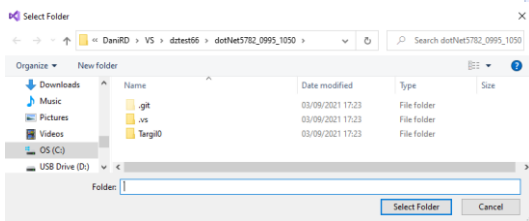
6. בתפריט Git של VS נוכלו למצוא אפשרויות **Manage Branches** ו-**View Branch History**. תפתחו אותן ותעיינו בלשוניות שנפתחות, תנסו להבין מה רואים.

## ג. עדכון המאגר המקומי אצל השותף השני (ע"י השותף השני)

1. בשלב הזה נכנס לתמונה **השותף השני** במחשבו האישי. כמובן **השותף השני** כבר יצר מאגר מקומי ע"י השלמת ההנחיות ב**פרק 4 במלואו ובמדויק**. שימו לב שאם השותף השני ביצע את פרק 4 לאחר שהשותף הראשון השלים את דחיפת השינויים של שלב ב' בפרק הזה (כנ"ל), בתיקיית העבודה של המאגר כבר תופיע תיקיית ה-Solution. נתייחס למצב הזה בהמשך – אך חובה על שני השותפים לקרוא את הנחיות השלב הזה בעיון, כי כאן תיבנה ההבנה של פעולת משיכת העדכונים (pull).
2. השותף השני יריץ את ויזואל סטודיו, אך במקום יצירת פרויקט חדש יבקש לפתוח את תיקיית המאגר לפי אחת משני התמונות הבאות (תלוי באיזה מצב יהיה ה-VS):



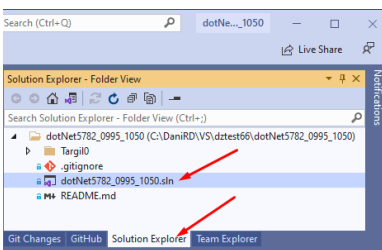
3. בחלון שייפתח סייר לתיקיית המאגר ופתח אותה (כפתור Select Folder):



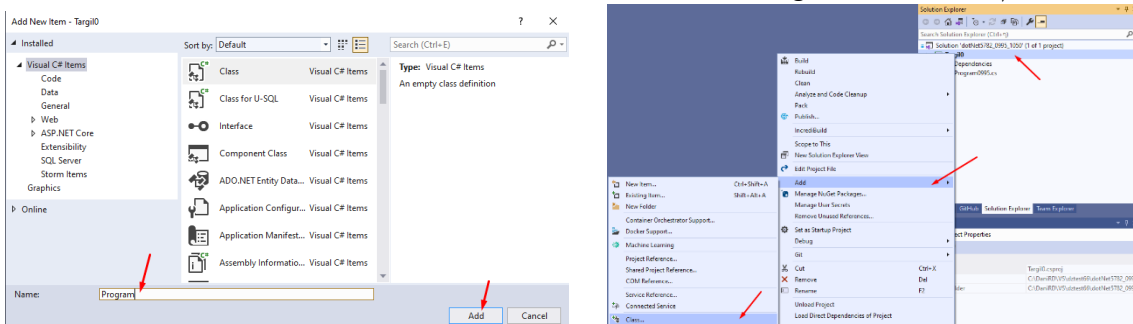
4. אם **פרק 4** בוצע ע"י השותף השני לפני נוספת פרויקט VS ע"י השותף הראשון, על השותף השני לבצע משיכת השינויים ע"י לחיצה על כפתור משיכת עדכונים (pull).



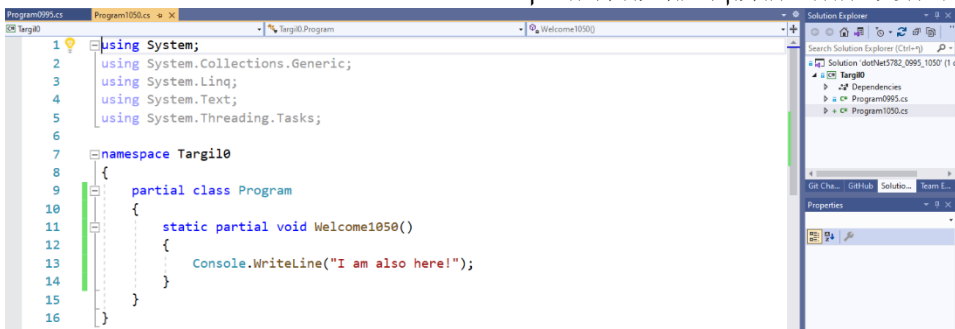
5. עבור ללשונית Solution Explorer ובחר בלחיצה כפולה (double-click) את קובץ הניהול של VS עם סיומת .sln:



6. VS יחליף את הסביבה מהתיקייה של המאגר לפתרון ב-C# הזהה למאגר ב-github.com ולמאגר המקומי של השותף הראשון.
7. כעת, השותף השני יוסיף "קובץ נוסף" למחלקה, שבו הגדרה נוספת של המחלקה החלקית Program – ע"י הוספת מחלקה חדשה בשם Program:



8. לאחר מכן ישנה השותף השני את שם המודול (הקובץ) החדש ע"י הוספת ארבעת הספרות האחרונות של מספר ת"ז שלו (ProgramYYYY.cs) – ללא עדכונים רוחביים (עם לחיצה על No בחלון הקופץ).
9. להגדרת המחלקה יוסיף השותף השני את מילת מפתח partial בדומה למה שעשה השותף הראשון, כמו כן יוסיף בתוך המחלקה את הפעולה WellcomeYYYY, הפעם עם קוד שבו יבצע פלט ו-ואו קלט ברצונו.
10. לאחר שסיימם את השלב הזה – החלון אמור להיראות כך:





ד. עדכון וסנכרון המאגר של השותף השני (ע"י השותף השני)

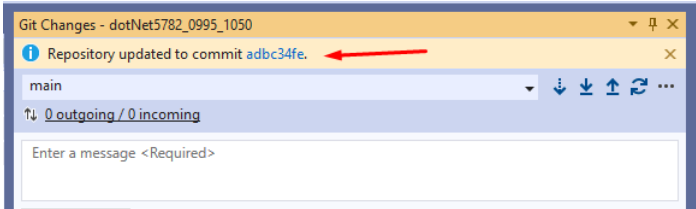
1. **השותף השני** יבצע גם הוא את עדכון השינויים במאגר המקומי ושליחת העדכון למאגר המרוחק כפי שמופיע בשלבים א' ו-ב' של הפרק הזה.

ה. סנכרון המאגר של השותף הראשון (ע"י השותף הראשון)

1. **השותף הראשון** יבצע משיכת העדכון של השותף השני ע"י לחיצה על כפתור **pull** בלשונית **Git Changes**.



2. עקוב אחר ביצוע הפעולה בשורת המצב וראה את הודעת ההצלחה.



**סיכום:** בשני הפרקים האחרונים, הדגמנו איך כל אחד מהשותפים לזוג יכול לעבוד במקביל, במחשב שלו על מודולים/קבצים שונים באותה תיקייה בתוך המאגר, ובסוף, מעבירים את העבודה של כל אחד מהשותפים לזוג למחשב של השני.

באופן כללי, כל פעם שתלמיד א' רוצה לעדכן את תלמיד ב' במה שהוא עשה, הסדר הוא:

1. כל אחד משני השותפים מעדכן במאגר המקומי שלו (**Commit**) את השינויים שביצע במחשב שלו.
2. שותף ראשון מבצע סנכרון מול המאגר באתר (מעביר את השינויים שאושרו על ידיו לאתר).
3. שותף שני מסנכרן מול המאגר באתר (מושך את העדכונים של השותף הראשון ודוחף את העדכונים של עצמו).
4. שותף ראשון מסנכרן עוד פעם מול המאגר באתר (מושך את העדכונים של השותף השני).
5. וחוזר חלילה סעיף 3 לסירוגין... כל פעולת סנכרון מושכת עדכונים מהמאגר בגיטהאב (**pull**) ואז דוחפת את השינויים מהמאגר המקומי לגיטהאב (**push**). שתי הפעולות **push + pull** אוגדו ב-VS ל"פעולה" אחת של **sync** שהזכרנו קודם. והכלל הלא כתוב הוא:

**"Always Pull Before Push"**

כעת נעבור להדגמת עבודה במקביל על אותו הקובץ.

## 7. פעולות נוספות לתרגול עבודה בזוגות<sup>2</sup>

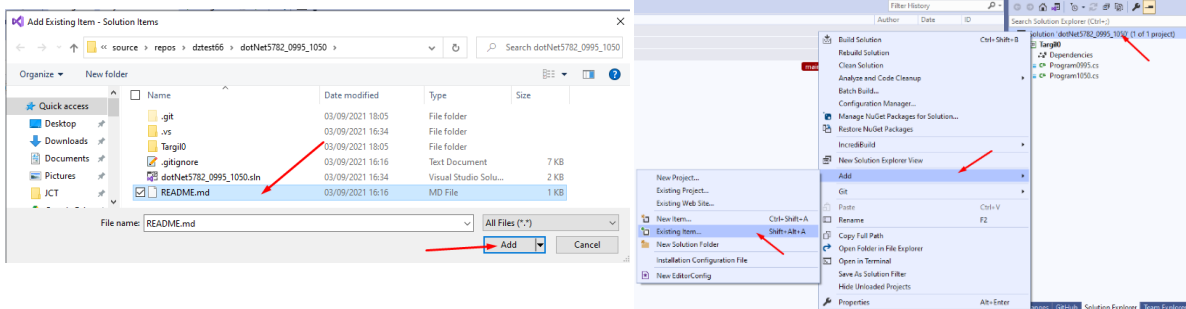
בשלב זה אנחנו מתרגלים ביצוע שינויים במקביל, על ידי שני השותפים לזוג, **באותו הקובץ**. בסיום כל שלב, חובה לבצע סינכרון, ולוודא שכל אחד מהצדדים נשאר עם השינויים ששני הצדדים עשו.

תזכורת של הפעולות שאנו הולכים לבצע בשלב זה (ובכלל בכל עבודה משותפת) - כל פעם שמבצעים עדכון המאגר יש 3 שלבים בלשונית של **Git Changes**:

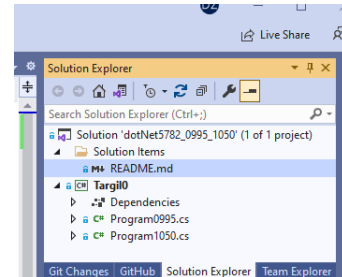
- א. כתיבת הערה **<= Commit All** [אישור השינויים שביצעתי במחשב שלי]
- ב. כפתור **pull** [קבלת השינויים שאחרים ביצעו מהמאגר הראשי למחשב האישי ומיזוגם עם השינויים שביצעתי במחשב האישי שלי]
- ג. כפתור **push** [עדכון השינויים במאגר הראשי ב **github.com**]
- ד. נ.ב. במקום שתי הפעולות ב. ו-ג. ניתן לבצע פעולה אחת ב-**VS** ע"י כפתור **Sync**

## א. עבודה באותו קובץ, בשורות שונות

במאגר שלנו, בתיקייה הראשית קיים קובץ שנקרא **README.md**. הקובץ הזה בפורמט **markdown** נוצר על ידי github כשיצרנו את המאגר. אנחנו נתרגל בחלק זה שינויים בקובץ (השינויים יכולים להתבצע גם ישירות ב-github – אך אנו רוצים לעבוד דווקא ב-VS). על מנת לפתוח את הקובץ ב-VS יש להוסיף קובץ קיים:



לאחר מכן בלשונית של **Solution Explorer** תופיע תיקיה וירטואלית (מדומה) בשם **Solution items** עם הקובץ:

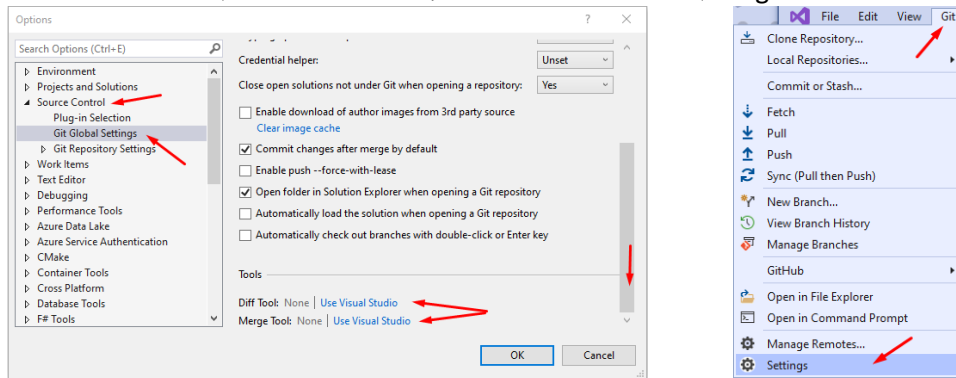


1. אחד מהשותפים לזוג יוסיף שורה לפני השורה הראשונה ויכתוב בה משהו כרצונו. לאחר מכן ישלח את השינוי לעדכון (Commit) במאגר המקומי של גיט וידחוף את העדכון (push) למאגר המרוחק בגיטהאב
2. השני מהשותפים לזוג ימשוך (pull) את העדכון מגיטהאב ויוסיף שורה אחרי כל השורות וכתוב בה **Hello World**. לאחר מכן ישלח את השינוי לעדכון במאגר המקומי של גיט וידחוף את העדכון למאגר המרוחק בגיטהאב
3. השותף הראשון ימשוך את העדכון האחרון למאגר המקומי שלו.
4. וודאו ששני השותפים לזוג קיבלו את כל השורות שהיו קודם, בתוספת שתי השורות החדשות.

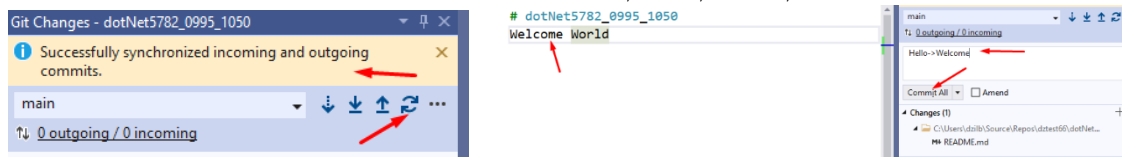
<sup>2</sup> ניתן לבצע את רוב החלק הזה על ידי אדם בודד. פשוט מבצעים צד אחד במחשב בתוכנת VS, וצד שני באתר של **github**. מתרגלים את המשיכה ודחיפה מתוך **VS**. החיסרון: המדריך לא כולל הדרכה של "איך לבצע שינויים בקוד באתר של **github**". מצד שני, זה די אינטואיטיבי.

## ב. עבודה באותן שורות

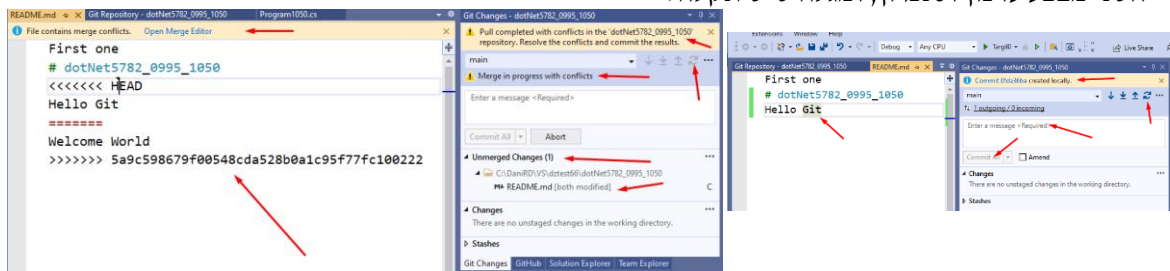
- שני השותפים נכנסים לתפריט Git <= Settings ובחלון שנפתח בוחרים בצד שמאל Source Control <= Git Global Settings. בחלון הזה גוללים בצד ימין עד למטה ומוודאים שבכלים מוגדר עבור שני הכלים (Merge Tool ו-Diff Tool) שימוש בויזואל סטודיו (Use Visual Studio).



- אחד מהשותפים לוג משנה את המילה Hello למילה אחרת כרצונו, השני משנה את המילה world.
- אחד מהשותפים לוג מבצע עדכון וסנכרון ראשוני.

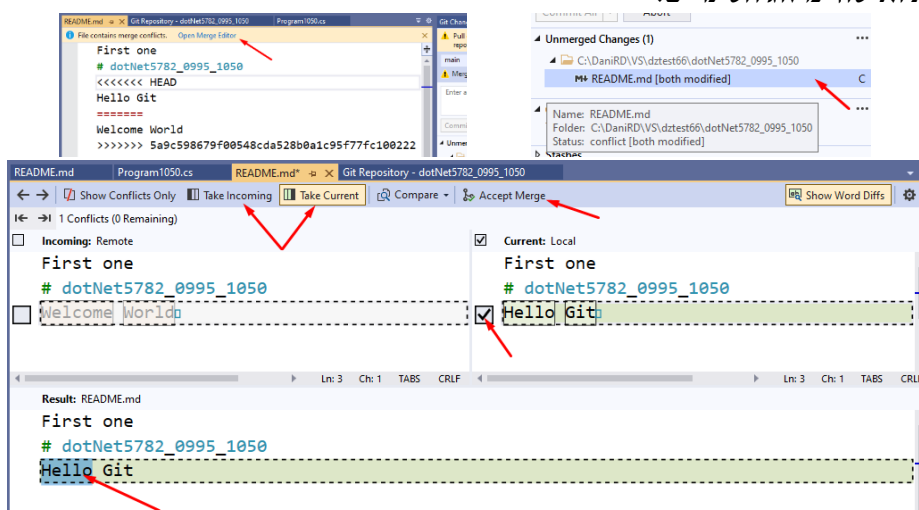


- השני מבצע עדכון וסנכרון, ומגלה שיש תקלה:



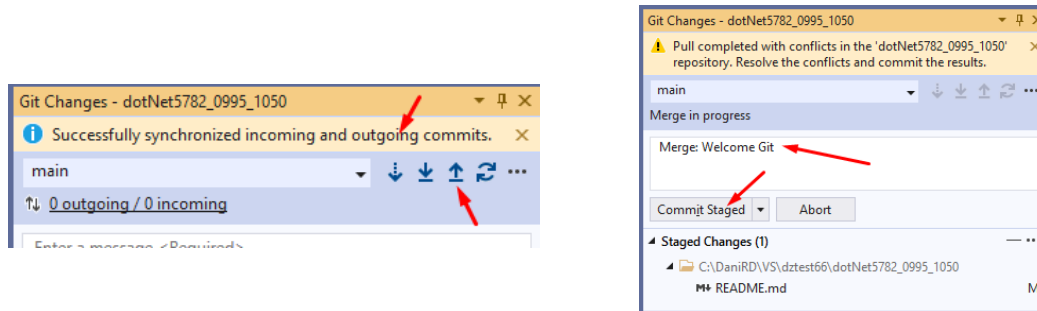
כך נראים חלון הקוד וחלון ה-Git Changes. חלון הקוד מציג את הנקודה בה יש סתירה בין שתי הגרסאות (כמה חיצים בתחילת השורה), וחלון ה-Git Changes הני"ל, מציג את העובדה שהסנכרון לא הושלם, אלא יש תקלות.

- עשו לחיצה כפולה על הקובץ בחלק Unmerged Changes בחלון Git Changes או על הקישור Open Merge Editor בחלון המציג את ההתנגשות בקודבטקסט. ייפתח חלון ההשוואה עם כפתורי בחירת החלטות ואישור מיזוג השינויים.



- בחלון ניתן לסמן שינויים מסוימים בצד שמאל (השינויים שהגיעו מהמאגר המרוחק) או בצד שמאל (השינויים בעדכון במאגר המקומי שעוד לא נשלח למאגר הרחוק) או לסמן בכפתורים למעלה לקחת את כל השינויים מהמאגר המרוחק (Take Incoming) או מהמאגר המקומי (Take Current) ולאחר מכן לבצע עריכה ישירה ידנית למטה בתוצאה של המיזוג. לבסוף יש לחצות על כפתור אישור המיזוג.

7. בחלון ה-Git Changes השינויים יעברו ל-Staged Changes. השלימו את תיאור העדכון (מומלץ להתחיל ממילה Merge על מנת לציין שמדובר בעדכון-מיזוג. לאחר שכל הקבצים עם התנגשויות (כרגע יש רק אחד כזה) יעברו על הבמה, לחצו על כפתור Commit Staged ולאחר מכן דחפו את העדכון-מיזוג למאגר המרוחק.

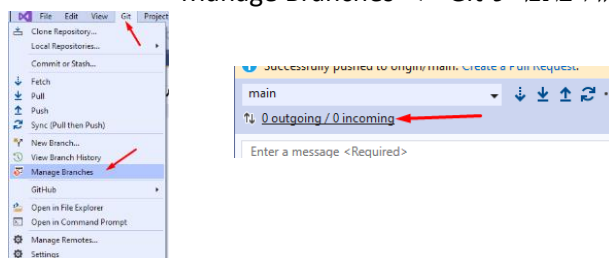


8. עשו עוד ניסיונות של שינויים "לא מסונכרנים" באותו הקובץ (לא חייב להיות באותה שורה – העיקר שהשינויים באותו הקובץ) שיגרמו להתנגשויות (תתחלפו ב"תפקידים" של הראשון והשני) ובדרך שחקו עם האפשרויות השונות בחלון המיזוג. הניסיון שתרכשו יעזור לכם בהתמודדות עם בעיות של חוסר תאום פעולה בין בני הזוג במהלך העבודה על תרגילי ופרויקט הקורס. שימו לב שבעבודה על פתרון ההתנגשויות אין כלל ברור, לפעמים הגרסה הסופית תהיה שילוב של שני השינויים שיתווספו ידנית על ידי הממוזג.

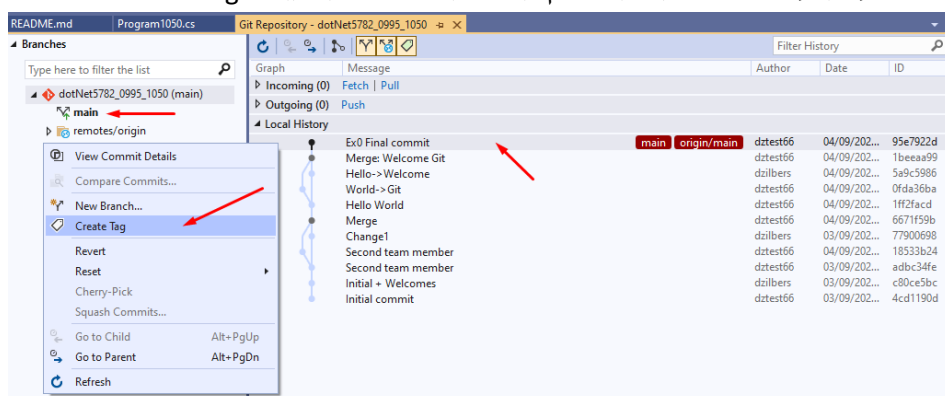
## א. יצירת תג (Tag)

לפני הגשה יש ליצור טאג במאגר הגיטהאב.

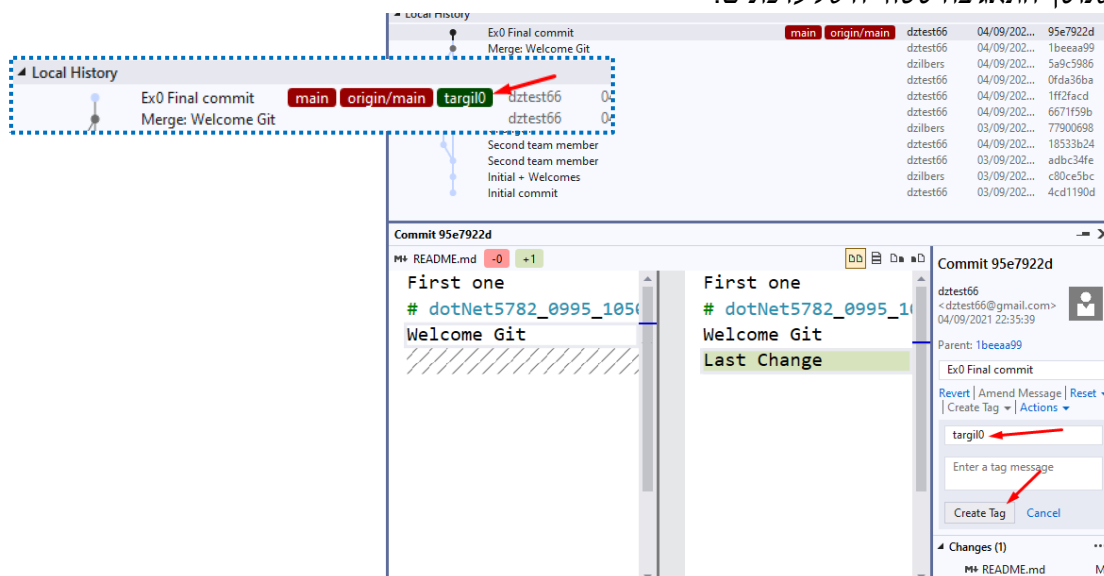
1. בצעו את השינוי האחרון ועדכנו אותו עם כיתוב "Ex0 Final commit", בצעו סנכרון העדכון לאתר.
2. בויזואל סטודיו של השותף שביצע את העדכון האחרון הנ"ל לחצו על הקישור 0 outgoing / 0 incoming או Manage Branches <= Git בתפריט



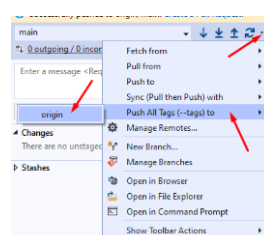
3. בחלון שיפתח וודאו שבצד שמאל מודגש הענף המקומי הראשי. לאחר מכן עם הסמן מעל העדכון האחרון לחצו על כפתור הימני בעכבר על מנת לפתוח תפריט הקשר. בתפריט שיפתח בחרו ב-Create Tag.



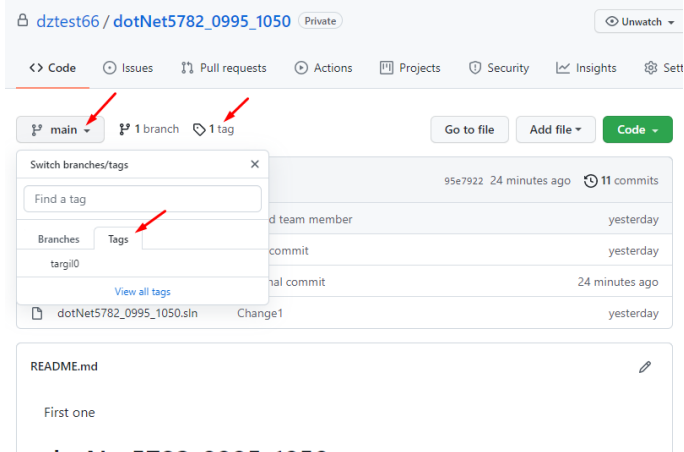
4. ייפתח חלון עם תצוגת השינויים של העדכון ובצד ימין חלק של פעולות על העדכון, כולל חוספת תאג. בשדה שכתוב בו בהתחלה Enter a tag name כתבו את שם התאג עבור ההגשה **targil0**. אין צורך להוסיף הודעת הסבר – כי העברתו לא בהכרח נתמכת בדחיפת התאג למאגר המרוחק. לחצו על הכפתור Create Tag. תראו שנוסף התאג בהיסטוריה של עדכונים.



5. בחלון Git Changes לחצו על כפתור ובתפריט שיפתח בחרו בדחיפת כל התאגים ובחרו בחיבור המרוחק .origin

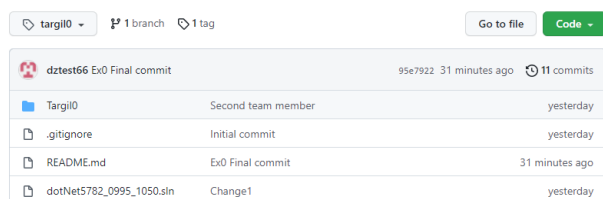


6. כנסו למאגר באתר גיטהאב (או רענונו את החלון אם כבר הייתם שם). שימו לב שבנוסף לכיתוב branch 1 נוסף גם כיתוב tag 1. ב תפריט הענפים (עם הענף הראשי) בחרו בלשונית Tags. תראו שם את התאג שייצרתם.

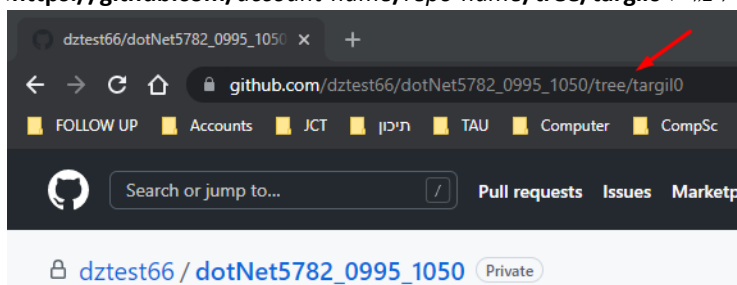


7. אם תלחצו על הכיתוב tag 1 ייפתח חלון עם כל התאגים (כרגע רק אחד), לחצו בחלון הזה על שם התאג שייצרתם (tag10) ותראו יותר פרטים עליו. שימו לב שבגיטהאב מייצרים גרסת שחרור (release) עבור התאג – משהו ייחודי לאתר גיטהאב ולא קשור לגיט עצמו.

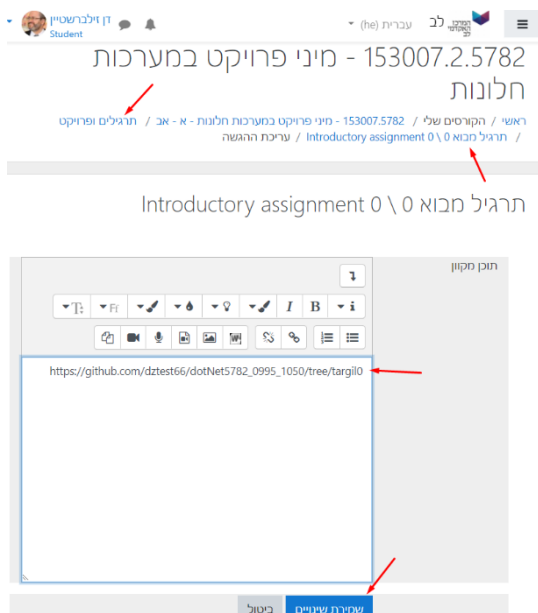
8. בתפריט שפתחתם בסעיף 6 לעיל לחצו על שם התאג. הכיתוב של הענף הראשי יתחלף לתאג.



9. בעמוד שמוצג בכתובת הדף יופיע משהו דמוי: <https://github.com/account-name/repo-name/tree/tag10>



10. עליכם להעתיק את הכתובת (הקישור) הזו לטקסט בתיבת ההגשה.



ב. שימו לב – כל אחד מהשותפים לזוג חייב לבצע בנפרד את ההגשה לתיבת ההגשה של התרגיל!