

## תוכן עניינים

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

## מטרת התרגיל

מטרת התרגיל הינה הכרות ראשונית עם :

- שפת C#
- סביבת הפיתוח Visual Studio 2019
- מערכת ניהול גרסאות Git
- שירות ענן של הארכת 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.

✓ 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.

✓ 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-github.com. שם ה-Solution והמאגר יהיה זהה והפורמט שלו יהיה : dotNet5781\_ xxxx\_yyyy.

✓ עבור כל תרגיל שם הפרויקט יהיה בפורמט הבא : dotNet5781\_NN\_xxxx\_yyyy. כאשר :

- dotNet5781 – שמו המקוצר של הקורס ומספר השנה העברית בספרות לועזיות
- NN – מספר התרגיל בשתי ספרות בלבד – עם אפס מוביל (מספר התרגיל הזה – 00)
- xxxx ו- yyyy – 4 ספרות אחרונות של תעודת הזהות של שני הסטודנטים

אי הקפדה על פורמטים אלה תמנע קבלת ציון לתרגיל או לכל התרגילים בהתאם.

✓ הגשת התרגילים תתבצע ע"י הזנת הקישור לגרסת המאגר (Release/Tag) ב-github.com שנוצרה 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 :
  - 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

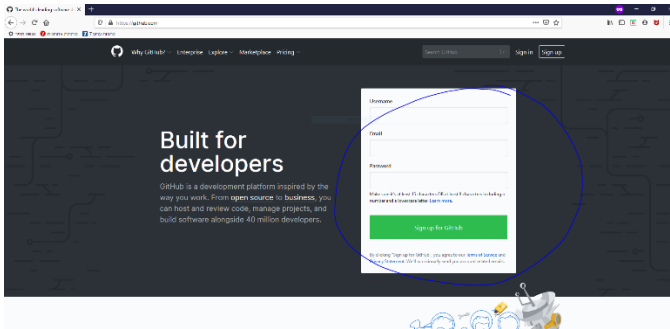
## 1. פתיחת חשבון ב-github.com

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"א):



3. הכניסו שם משתמש, כתובת דוא"ל שלך (אם תשתמשו בכתובת דוא"ל שלכם מהמכון – עם הדומיין g.jct.ac.il, אפשר לקבל הטבות לסטודנטים), ובחרו סיסמא לפי ההוראות.

4. בדף הבא, צריך לפתור פאזל (נכון לתשרי תשפ"א), וללחוץ על כפתור Join a free plan.

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

What kind of work do you do, mainly?

Student  
I go to school

How much programming experience do you have?

A moderate amount  
I'm somewhat experienced

או

A little  
I'm new to programming

What do you plan to use GitHub for?

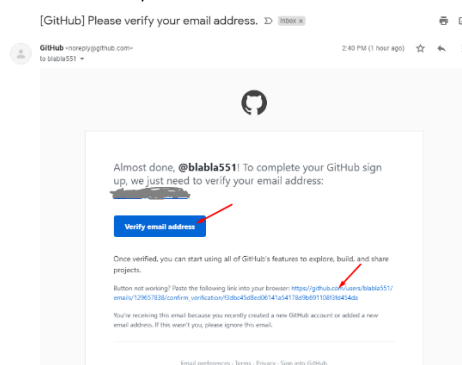
School work and student projects

, ואולי עוד אחת או שתיים מן האפשרויות בסעיף הזה (אפשר לסמן עד 3)

I am interested in: C#, .NET

כמובן תוכלו לרשום עוד, לפי רצונכם.

6. היכנסו לשירות הדוא"ל שלכם, יש שם הודעה מ-github.com עם כפתור לאימות כתובת דוא"ל. לחצו עליו.



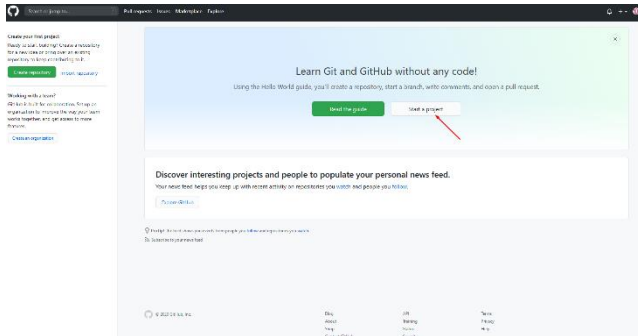
7. מגיעים לדף שבו מציעים ליצור מאגר (Repository) חדש.

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:



3. לחצו על כפתור Start a project. אפשר גם ללחוץ על הכפתור הירוק ולקרוא את המדריך הבסיסי.
4. בחלון הבא תיצרו את המאגר הראשון. לפני היצירה הוא נראה ככה: פרטים שצריך לשים לב אליהם:

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

Owner \*  / Repository name \*

Great repository names are short and memorable. Need inspiration? How about [furry-adventure?](#)

Description (optional)

☐ Public  
Anyone on the internet can see this repository. You choose who can commit.

☒ Private  
You choose who can see and commit to this repository.

Initialize this repository with:  
Skip this step if you're importing an existing repository.

☒ Add a README file  
This is where you can write a long description for your project. [Learn more.](#)

☒ Add .gitignore  
Choose which files not to track from a list of templates. [Learn more.](#)  
.gitignore template:

☐ Choose a license  
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set [master](#) as the default branch. Change the default name in your [settings](#).

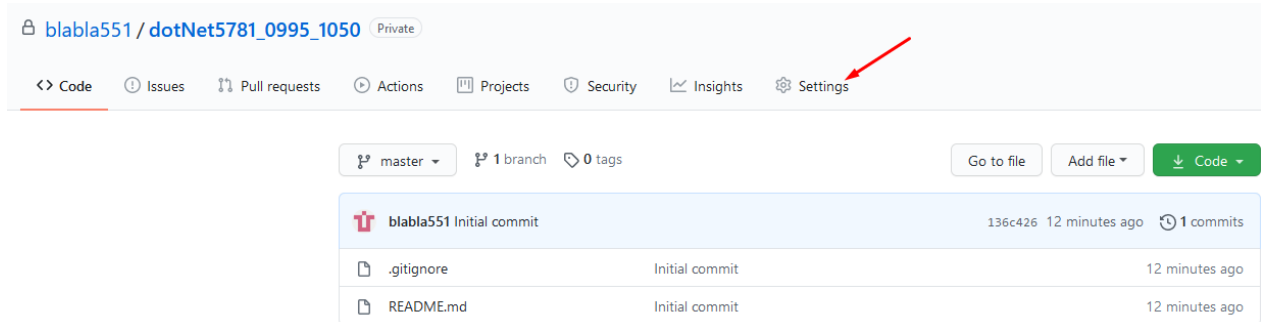
- א. הקלידו את שם המאגר לפי הדרישות כנ"ל.
  - ב. תיאור (Description) – רצוי לרשום, אפשר לדלג.
  - ג. המאגר חייב להיות מוגדר **פרטי (Private)**. במקרה של העתקה ממאגר ציבורי, גם בעל המאגר הציבורי ייחשב כאשם (לא עכברא גנב אלא חורא גנב...).
  - ד. סמנו את התיבה ליד "Add a README file".
  - ה. סמנו את התיבה 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).
5. לחצו על הכפתור הירוק Create repository.

### 3. שיתוף בן הזוג במאגר

פרק זה יתבצע על ידי אותו הסטודנט שביצע את פרק 2

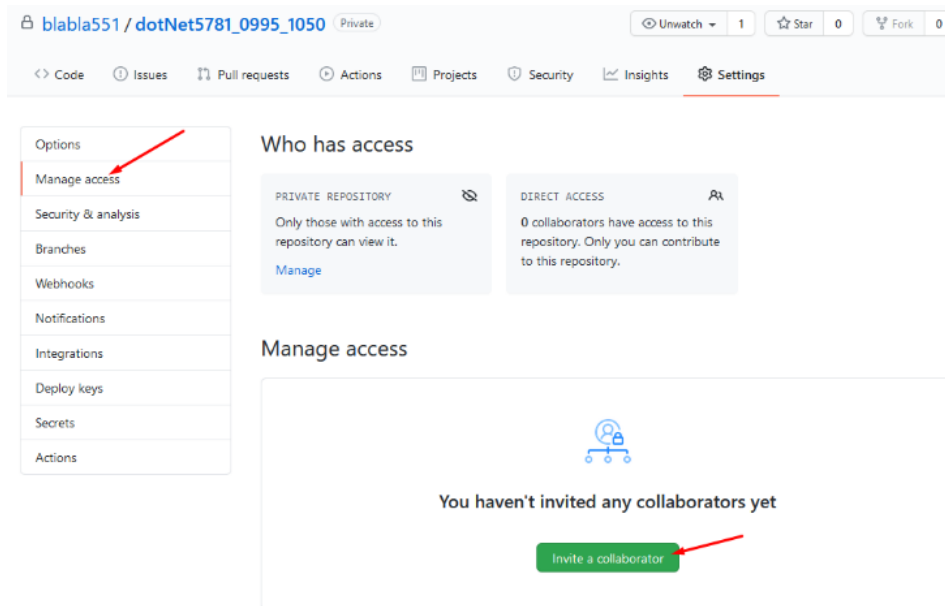
אחרי פרק 2, יש לנו מאגר פרטי, ואנחנו רוצים לעבוד עליו עם עוד מישהו. לצורך העניין נניח שהוא כבר פתח לעצמו חשבון ב-github.com (לפי ההוראות בפרק 1).

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 בברטיסיית Settings בסרגל הכרטיסיות.

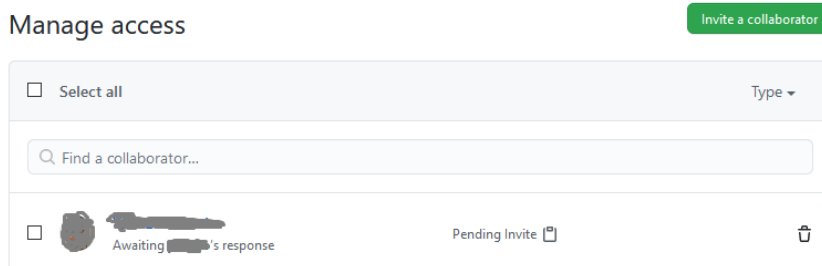
3. בחרו "Manage Access" בצד שמאל, ואז לחצו על הכפתור הירוק Invite a collaborator (הזמן משת"פ):



4. רשמו את שם המשתמש של שותפכם לזוג, ולחצו על <Add <user> to <repository>

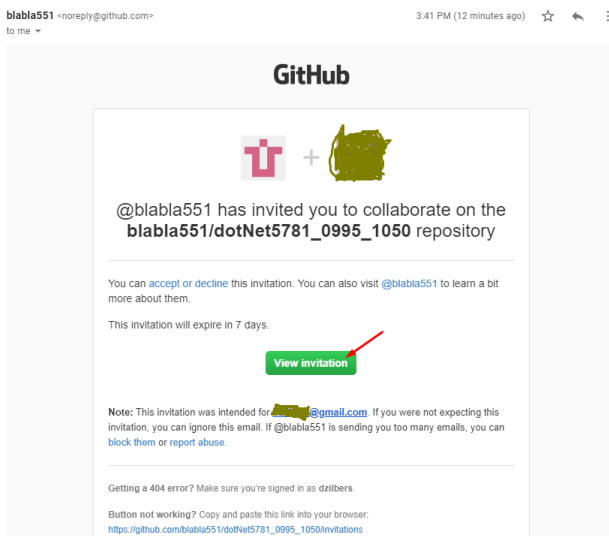
5. בשלב הזה, github.com אמורים לשלוח קישור של ההזמנה לדוא"ל של המשתמש שהוזמן.

#### Manage access



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

blabla551 invited you to blabla551/dotNet5781\_0995\_1050



blabla551 invited you to collaborate

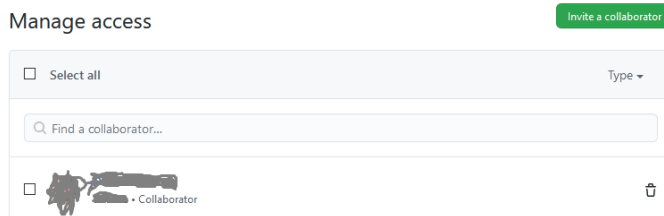
Accept invitation Decline

Owners of dotNet5781\_0995\_1050 will be able to see:

- Your public profile information
- Certain activity within this repository
- Country of request origin
- Your access level for this repository
- Your IP address

Is this user sending spam or malicious content?  
[Block blabla551](#)

לאחר האישור, בדף ניהול הגישה של השותף הראשון יופיע:



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.

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, מאגר בודד עם תוכן בסיסי (קובץ README 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 לשותף גם את מרצה ואת בודק הקבוצה בה אתם לומדים.

#### 4. יצירת עותק מקומי של המאגר

פרק זה יתבצע בנפרד 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.

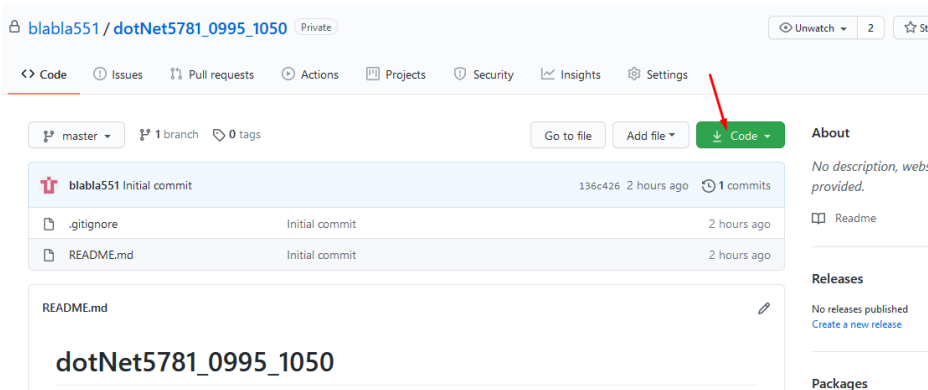
בהמשך, נדאג כל פעם שאנחנו יוצרים פרויקט ליצור אותו באותו הפתרון (Solution) של הקורס שכבר נמצא בתיקיית המאגר, ולכן כשנעדכן את המאגר מול github.com – "נדחוף" (push) ל-github.com את השינויים שעשינו, ונמשוך (pull) מ-github.com את השינויים שהשותף שלנו לזוג עשה.

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

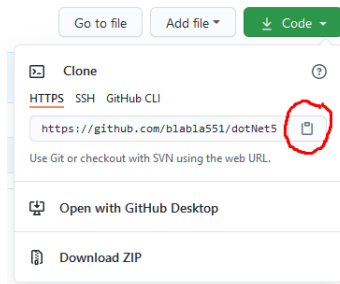
שים לב שיש להשתמש בגרסה האחרונה ביותר (כולל עדכון ל-4.8 framework), וייתכן שלצורך כך תצטרך לעדכן את גרסת VS שיש בידך. להנחיות להתקנה ועדכון, ניתן להיעזר במסמך ההנחיות שנמצא במודל תחת עזרים שונים.

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

2. לחצו על הכפתור הירוק "Code"

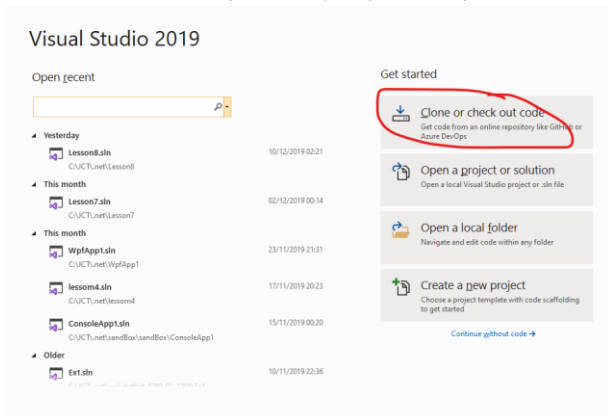


3. תופיע כתובת, אפשר לסמן אותה ולהעתיק, או פשוט לחוץ על הכפתור "העתק":

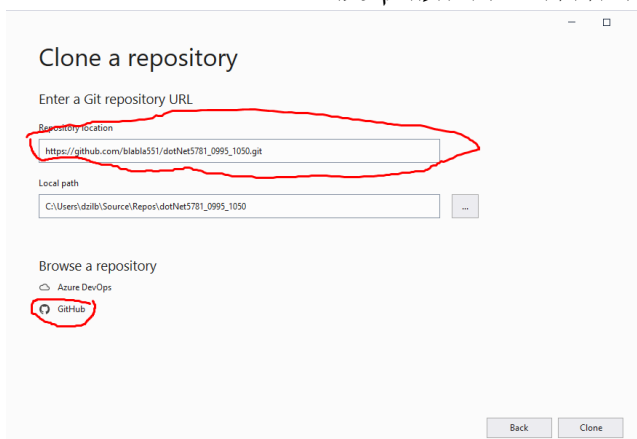


הכתובת בדרך כלל תהיה בפורמט `https://github.com/username/repo-name.git`.

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



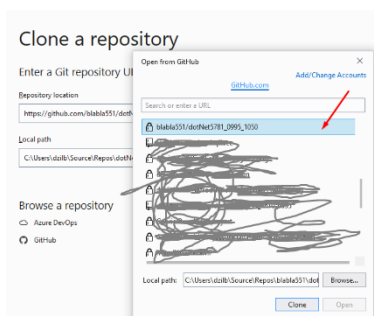
6. בחלון הבא, כתבו את הכתובת המאגר, ואיפה במחשב אנחנו רוצים את העותק שלו.



**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** המלצתנו שתיקיית המאגר תהיה ישירות תחת תת-תיקיה Sources/Repos/ שבתיקיית המשתמש – ללא שם החשבון ב-github.com.

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



7. לבסוף לחצו על כפתור Clone. 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. בחרו File->Close folder כדי לסגור את התיקיה ולחזור למסך הפתיחה של VS.

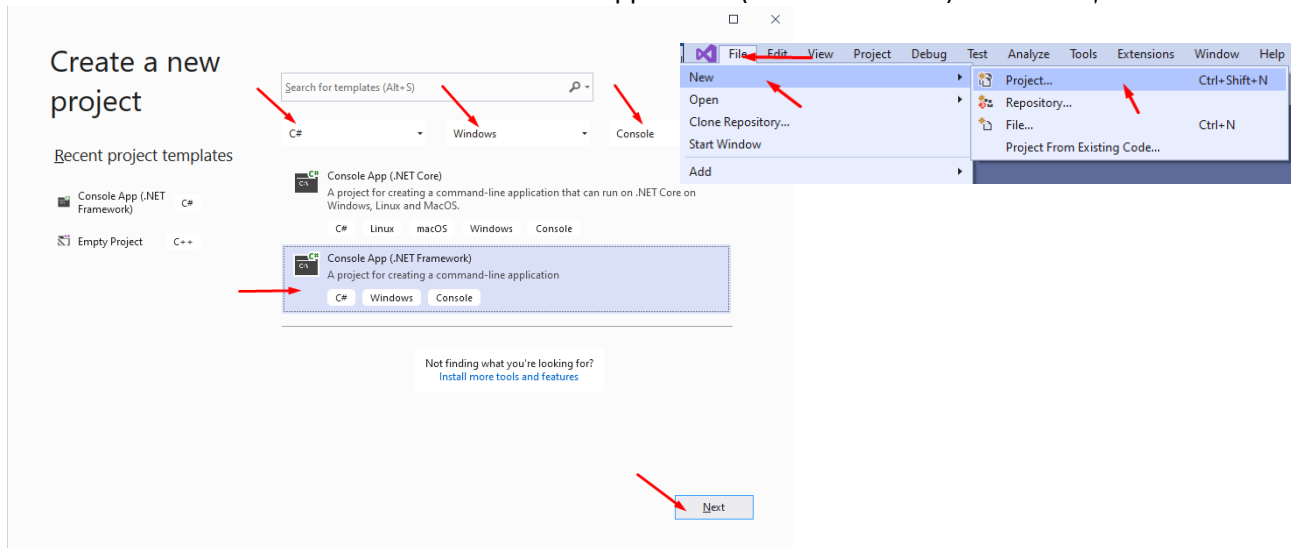


## 5. יצירת פתרון בתוך המאגר המקומי

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 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 מסוג Console Application (.NET Framework):



## Configure your new project

Console App (.NET Framework) C# Windows Console

Project name: dotNet5781\_00\_0995\_1050

Location: C:\Users\dzilb\source\repos

Solution: Create new solution

Solution name: dotNet5781\_0995\_1050

☐ Place solution and project in the same directory

Framework: .NET Framework 4.8

Back Create

שם הפתרון (Solution) יהיה dotNet5781\_00\_0995\_1050. כל אחד מהשותפים לזוג ישים במקום xxxx ו-yyyy את ארבע הספרות האחרונות של תעודת הזהות האישית שלו. השם של הפתרון יהיה זהה לשם של המאגר! (אותו סדר של תעודות זהות). שם הפרויקט יהיה dotNet5781\_00\_0995\_1050. תיקיית המאגר המקומי היא גם תהיה תיקיית הפתרון ופרויקט ימוקם בתוך תיקיית המאגר.

1.ב. מלאו את שם ה-Solution לפני שם הפרויקט – זה ימנע שינוי אוטומטי של המיקום (Location) ע"י Visual Studio תוך כדי הכנסת שם הפרויקט.

2.ב. בכל מקרה וודאו ששם התיקה 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.

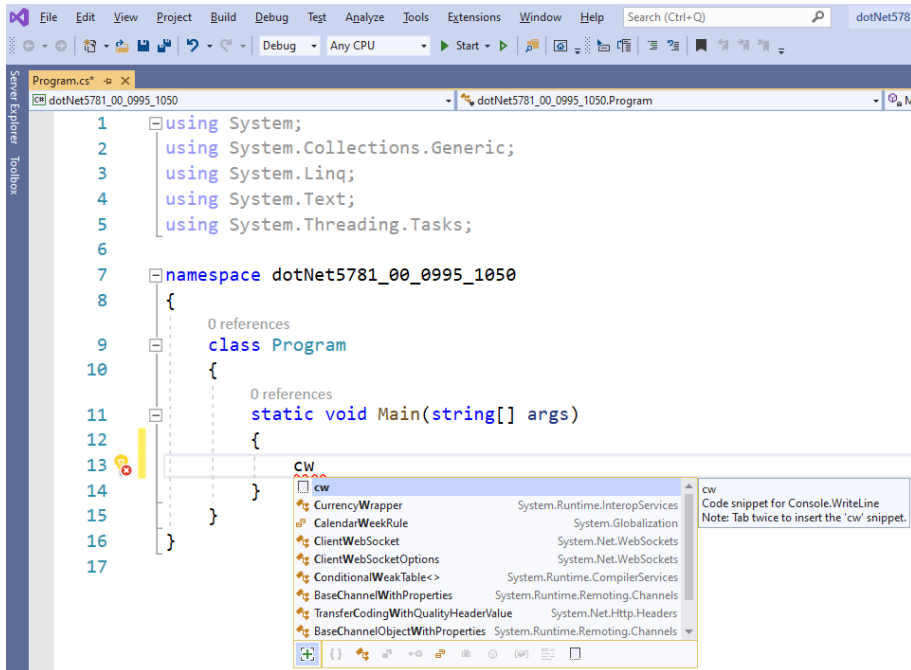
חשוב!! אין לסמן את תיבת הבדיקה Place solution and project in the same directory. כמו כן רצוי לוודא שגרסת ה framework היא הגרסה האחרונה והסופית – 4.8.

לחצו על כפתור Create (צור).

3.ב. 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 בסעיף 5 של פרק 4 לעיל).

## ב. שימוש ב snippet

בתוכנית הראשית (שנפתחה אוטומטית) תכתבו 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: "

נדפוס 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"

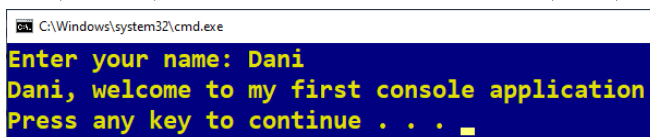
נקלוט מהמשתמש את שמו בעזרת הפעולה Console.ReadLine() ונדפוס את ההודעה הבאה בהתאם:

(כמוכן, עליך להגדיר משתנה, להדפוס 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();

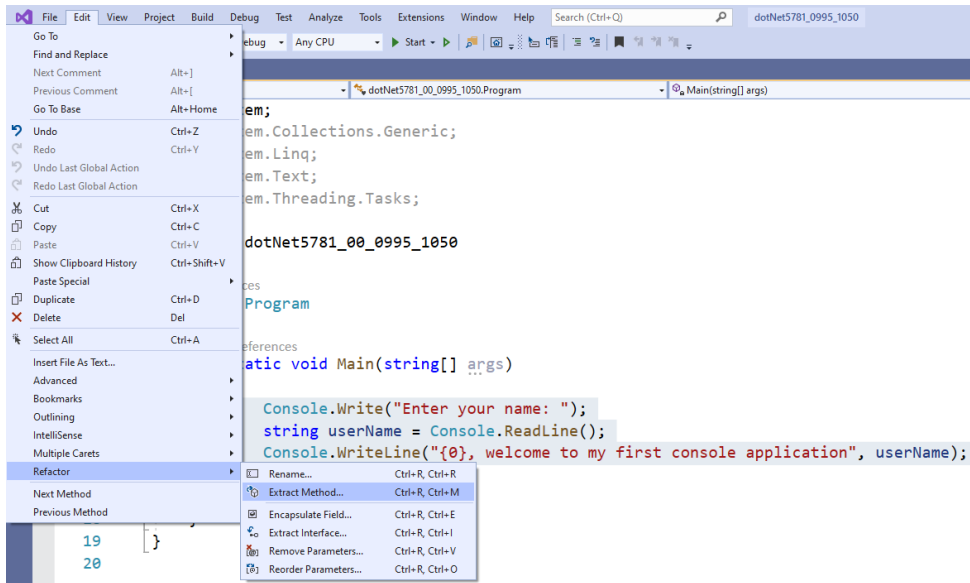
לחילופין, ניתן להריץ ע"י לחיצה למקשים Ctrl-F5 (במקום רק F5). לאחר ההרצה ומילוי השם החלון יראה כך:



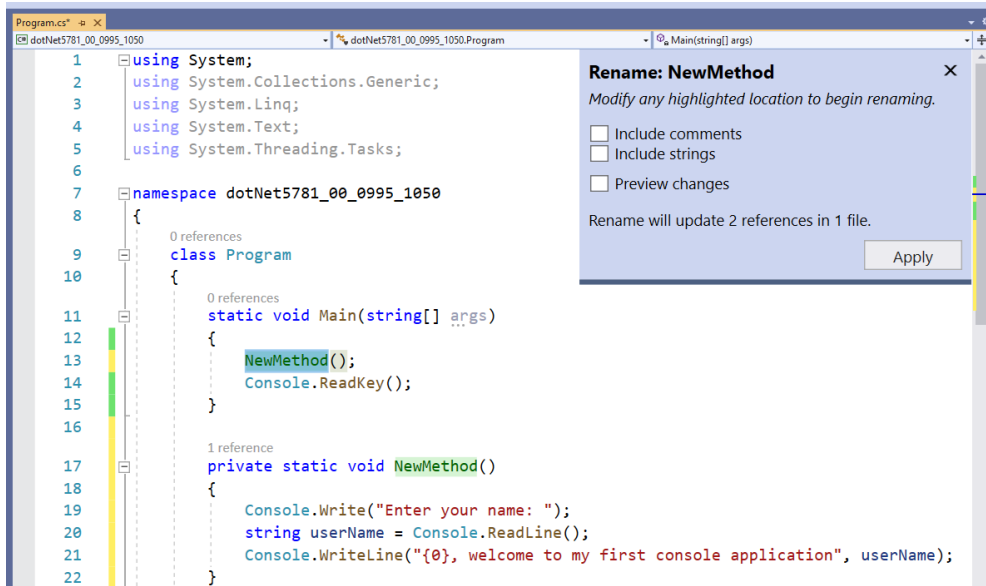
<sup>1</sup> רשימה חלקית של עוד snippet שונים ניתן לראות בקישור הבא (ניתן גם להיכנס לעורך snippet ע"י לחיצה Ctrl-K ואז Ctrl-B): <https://docs.microsoft.com/en-us/visualstudio/ide/visual-csharp-code-snippets?view=vs-2019>

#### ד. העברת הקוד לפונקציה באמצעות refactor

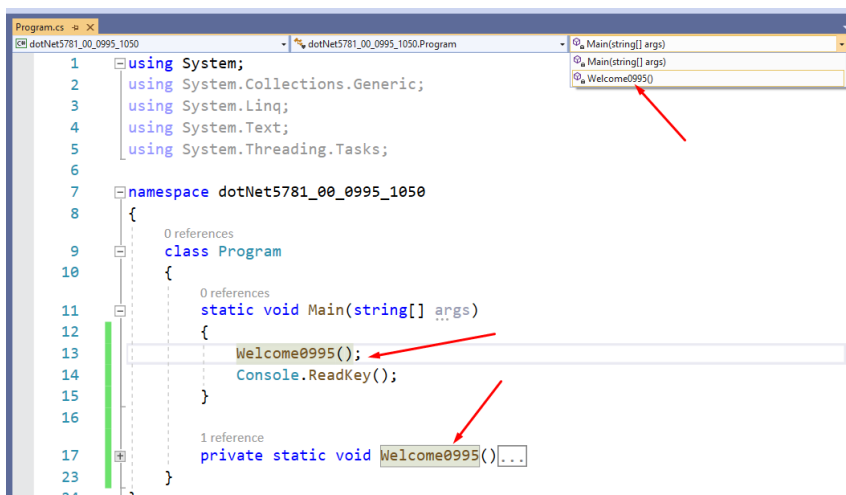
כעת נרצה שהקוד לא יהיה סתם בתוכנית הראשית אלא בתוך פונקציה נפרדת שנקרא לה מהתוכנית הראשית. נוכל להשתמש ב-refactor על מנת להעביר שורות בקוד לפונקציה נפרדת. נסמן את שלושת שורות הקוד שכתבנו ללא (`Console.ReadKey()`) ובתפריט העריכה נבחר תת-תפריט של Refactor ואז Extract Method.



הקוד שסימננו יועבר לפונקציה בשם `NewMethod` – השם יסומן ברקע ויפתח 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 בסינכרון בין שני השותפים, נשנה את הגדרת המחלקה למחלקה חלקית (בהמשך תלמדו עליה יותר לעומק). חשוב! הסינכרון יעבוד היטב גם במחלקה רגילה, ולא רק במחלקה חלקית. כך נעבוד ברוב המקרים בהמשך. אך כרגע לצורך הקלה על הלימוד נשתמש ברעיון החלקיות. לכן:

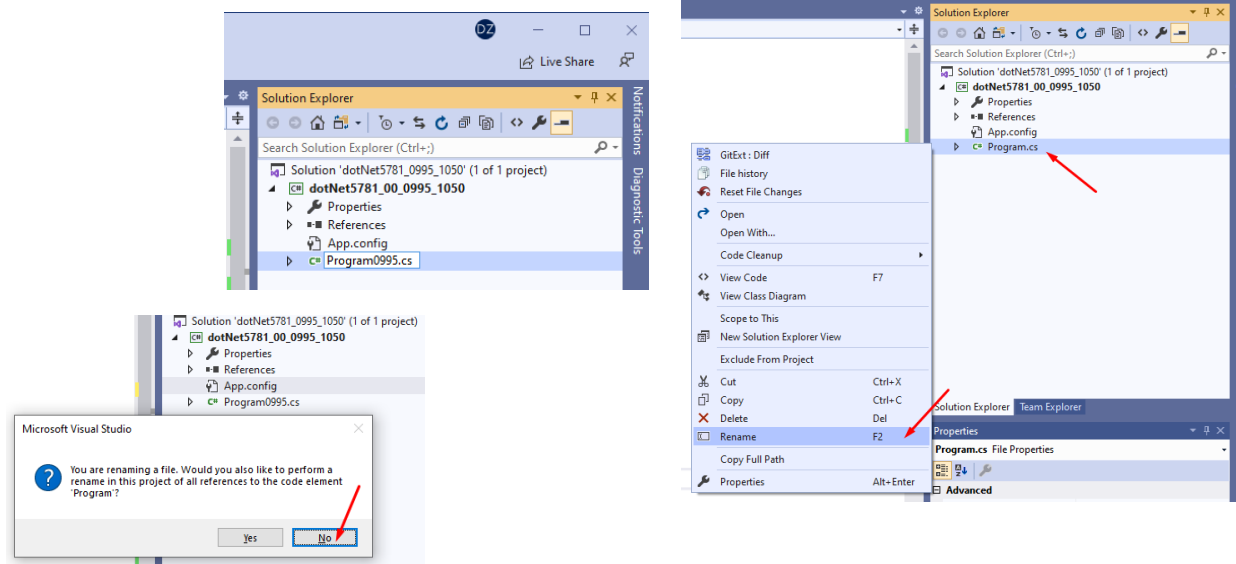
1. **השותף הראשון** יוסיף מילת מפתח **partial** לפני הגדרת המחלקה
2. וכן יוסיף הגדרת פונקציה חלקית ללא מימוש בשם `WelcomeYYYY`, כאשר `YYYY` אלו ארבע הספרות האחרונות של מספר ת"ז של השותף השני
3. וכן יפעיל את הפונקציה בתוכנית הראשית:

```
namespace dotNet5781_00_0995_1050
{
    0 references
    partial class Program
    {
        0 references
        static void Main(string[] args)
        {
            Welcome0995();
            Welcome1050();
            Console.ReadKey();
        }

        1 reference
        static partial void Welcome1050();

        1 reference
        private static void Welcome0995()
    }
}
```

4. בהמשך לכך **השותף הראשון** ישנה את שם המודול (הקובץ) ל- `ProgramXXXX.cs` – ארבע הספרות האחרונות של השותף הראשון
- חשוב! בחלון הקופץ של אישור עדכון רוחבי יש לבחור באפשרות **No** (זה חשוב על מנת לא לשנות את שם המחלקה):



## 6. עדכון וסינכרון 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,**

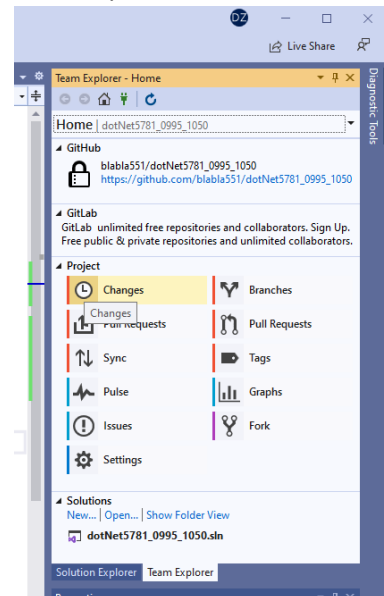
רצוי שבזמן ביצוע הפרק שני השותפים יהיו מחוברים ב-zoom כאשר השותפים משתפים את המסך לסירוגין.  
 סינכרון מורכב משני חלקים עיקריים: אישור השינויים שביצעתי לפרסום (commit), ועדכון מול המאגר הראשי.

סיכום קצר של הפעולות שאנו הולכים לבצע בשלב זה (ובכלל בכל עבודה משותפת) - כל פעם שמבצעים עדכון המאגר יש 3 שלבים בחלון של team explorer:

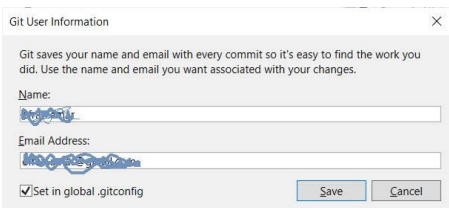
- מסך הבית <- Changes <- כתיבת הערה <- Commit All [אישור השינויים שביצעתי במחשב שלי]
- מסך הבית <- Sync <- pull [קבלת השינויים שאחרים ביצעו מהמאגר הראשי למחשב האישי ומיזוגם עם השינויים שביצעתי במחשב האישי שלי]
- עדיין מתפריט הסינכרון <- push [עדכון השינויים במאגר הראשי ב [github.com](https://github.com)]
- נ.ב. במקום שתי הפעולות ב. ו-ג. ניתן לבצע פעולה אחת ב-VS - מסך הבית <- Sync <- sync

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

1. בחרו View->Team Explorer. יקפוץ בצד ימין למעלה חלון כזה (במקום ה-Solution Explorer הרגיל):



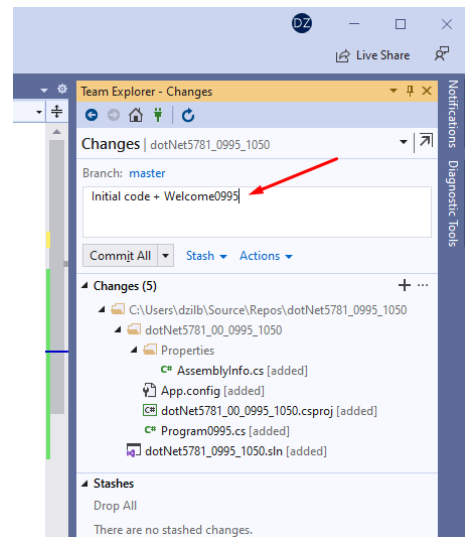
2. בחרו **Changes**. בפעם הראשונה נצטרך למסור פרטים שיתארו אותנו ויופיעו על כל "commit" שנעשה (כלומר, כל שורת קוד או קובץ שנוסיף, יהיה רשום שאנחנו הוספנו אותו, עם הפרטים האלו)



3. אם עובדים במחשב אישי, רצוי לסמן את תיבת "set in global .gitconfig"

4. בשלב הזה פתחו בחלון Team Explorer פירוט של השינויים שנעשו במאגר המקומי (לעומת המאגר בשרת שכלל רק את התיקייה הראשית, אנחנו הוספנו את הפתרון ואת הפרויקט).

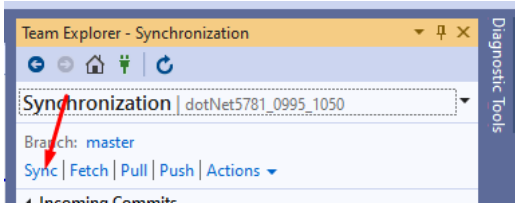
5. בחלק הצהוב חובה לרשום הערה שתאיר את השינוי שעשינו, למשל:



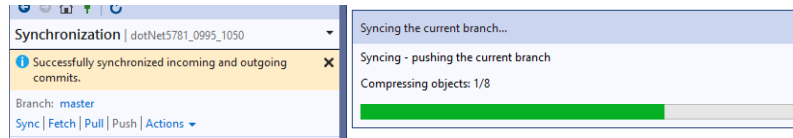
6. אחרי שנרשום הערה, נוכל ללחוץ על **Commit All** (לידיעתכם, יש גם אפשרות לעשות **Commit** רק על חלק מהשינויים, לא במסגרת המדריך הזה). אם בשלב הזה נרצה גם ישר לעדכן במאגר הראשי – ניתן ללחוץ על תפריט ליד **Commit All** ולבחור **Commit All and Sync** (בתרגילים הבאים ובפרויקט – אך לא בתרגיל הזה!)

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

1. לחצו על כפתור הבית בשורה העליונה, ומשם נבחר **Sync**.
2. החלון נראה:



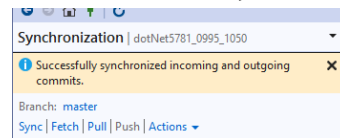
3. לחצו על **Sync**:



לפני נקודת זמן זו, התיקיה במחשב של השותף הראשון שונה מהמאגר המקורי בזה שהוא הוסיף את הפרויקט שלו. ה-sync יוסיף את הפרויקט שהוא יצר למאגר ב-github (אפשר להיכנס ולבדוק דרך האתר).

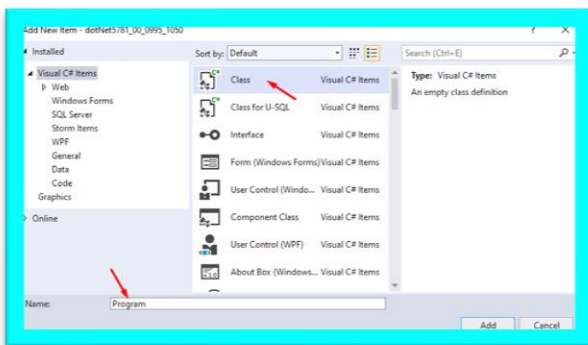
## ג. עדכון המאגר המקומי אצל השותף השני (ע"י השותף השני)

1. השותף השני יבצע את השלבים א-1 (בחר View->Team Explorer), יבחר **Sync** וילחץ על קישור **Sync** כמו בשלב ב לעיל.

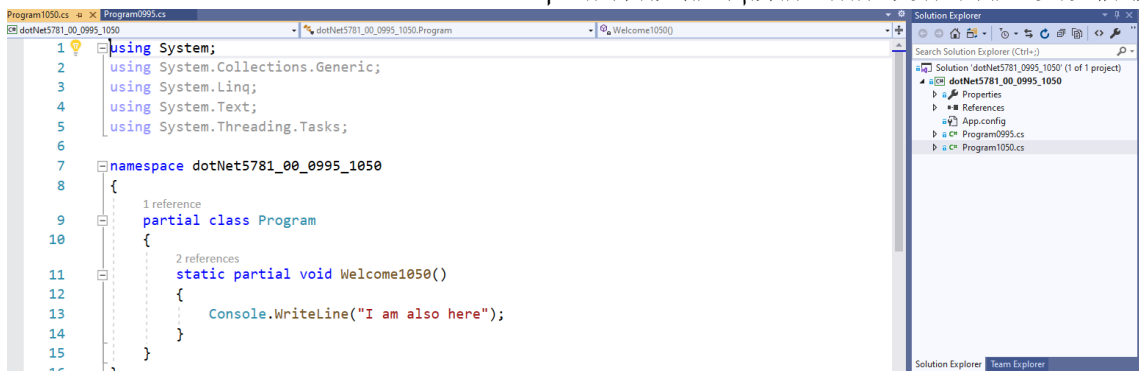


2. לאחר מכן השותף השני עובר ל-Solution explorer – יראה שם תיקיה חדשה שהתווספה (dotNet5781\_00\_xxxx\_yyyy) וקובץ dotNet5781\_xxxx\_yyyy.sln. יעשה לחיצה כפולה (double-click) עם העכבר על הקובץ של הפתרון (.sln) הני"ל. VS יחליף את הסביבה מהתיקיה של המאגר לפתרון ב-C# הזהה למאגר ב-github.com ולמאגר המקומי של השותף הראשון.

3. כעת, השותף השני יוסיף "קובץ נוסף" למחלקה, שבו הגדרה נוספת של המחלקה החלקית Program – ע"י **הוספת מחלקה חדשה בשם Program**:



4. 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).
5. 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.
6. לאחר שתסיימו 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 השלב א-1 (אישור השינויים במחשב שלו), יבחר **Sync** וילחץ על קישור **Sync** כמו בשלב ב לעיל.

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

1. השותף הראשון יבחר **Sync** וילחץ על קישור **Sync** כמו בשלב ב לעיל.

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 מאשר 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 (קולט את השינויים של השותף הראשון ומעביר את השינויים שאושרו על ידיו לאתר).
4. שותף ראשון מסנכרן עוד פעם מול המאגר באתר (קולט את השינויים של השותף השני)
5. וחוזר חלילה סעיף 3 לסירוגין... כל פעולת סנכרון מביאה שינויים מהמאגר באתר (pull) ואז מעבירה את השינויים מהמאגר המקומי לאתר (push). שתי הפעולות push + pull אוגדו ב-VS ל"פעולה" אחת של סנכרון. 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>

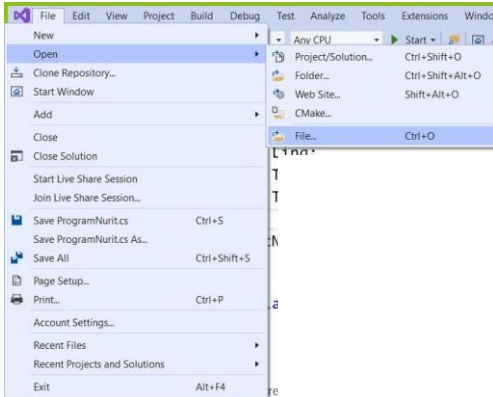
בשלב זה אנחנו מתרגלים ביצוע שינויים במקביל, על ידי שני השותפים לזוג, **באותו הקובץ**. בסיום כל שלב, חובה לבצע סינכרון, ולוודא שכל אחד מהצדדים נשאר עם השינויים ששני הצדדים עשו.

תזכורת: כל פעם שמתרגלים את עדכון המאגר יש 3 שלבים בחלון של team explorer:

- ה. מסך הבית <- Changes <- כתיבת הערה <- Commit All.
- ו. מסך הבית <- Sync <- pull
- ז. עדיין מתפריט הסינכרון <- push
- ח. נ.ב. במקום שתי הפעולות ב. ו-ג. ניתן לבצע פעולה אחת ב-VS - מסך הבית <- Sync <- sync

### א. עבודה באותו קובץ, בשורות שונות

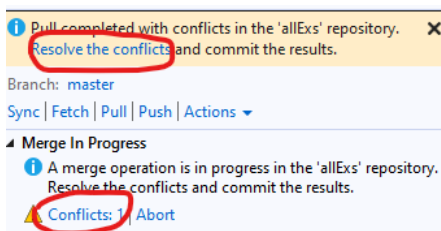
ט. במאגר שלנו, בתיקייה הראשית קיים קובץ שנקרא README.md. הקובץ הזה בפורמט [markdown](#) נוצר על ידי github כשיצרנו את המאגר. אנחנו נתרגל בחלק זה שינויים בקובץ (השינויים יכולים להתבצע גם ישירות ב-github – אך אנו רוצים לעבוד דווקא ב-VS). על מנת לפתוח את הקובץ ב-VS יש לבחור file->open->file:



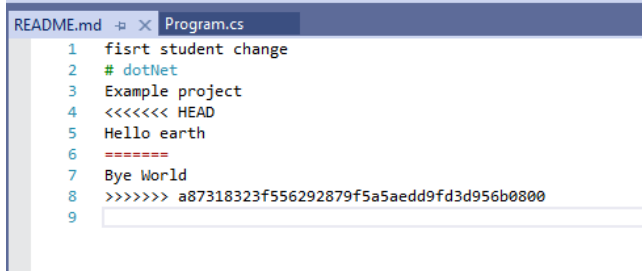
1. אחד מהשותפים לזוג יוסיף שורה לפני השורה הראשונה ויכתוב בה משהו כרצונו.
2. השני מהשותפים לזוג מוסיף שורה אחרי כל השורות וכותב בה Hello World.
3. שני השותפים לזוג מבצעים עדכון של המאגר (לפי השלבים הני"ל).
4. מוודאים ששני השותפים לזוג קיבלו את כל השורות שהיו קודם, בתוספת שתי השורות החדשות.

### ב. עבודה באותו שורה

1. אחד מהשותפים לזוג משנה את המילה Hello למילה אחרת כרצונו, השני משנה את המילה world.
2. בחלון team explorer נכנסים למסך הבית, ומשם להגדרות <- הגדרות גלובליות.
3. בחלק התחתון, תבחרו use visual studio עבור diff tool ו-merge tool.
4. אחד מהשותפים לזוג מבצע סינכרון ראשון.
5. השני מבצע סינכרון, ומגלה שיש תקלה:

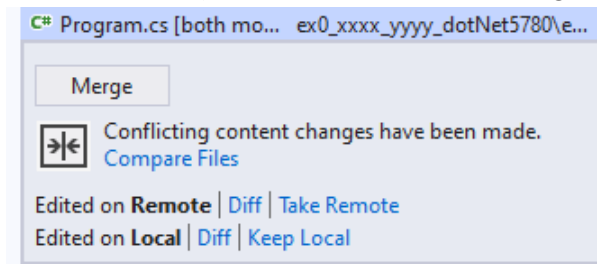


כך נראים חלון הקוד וחלון team explorer. חלון הקוד מציג את הנקודה בה יש סתירה בין שתי הגרסאות (כמה חיצים בתחילת השורה), וחלון ה-team הני"ל, מציג את העובדה שהסינכרון לא הושלם, אלא יש תקלות:



<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.

6. תלחצו על resolve the conflicts או על Conflicts: 1, (כמובן יש לבחור בקובץ עליו בו הקונפליקטים – במקרה שלנו README) ונגיע לחלון שבו מוצגים המקומות שgit לא הצליח למזג בעצמו. אם נלחץ עליו, ייפתח חלון



כזה :

7. בחלון הזה אפשר להשתמש ב-Compare files כדי לראות את ההבדלים. ואז להחליט איזו גירסה לוקחים, בהתאם לצרכים. (עבור כל שינוי, יש לבחור אם מעדיפים את הגרסה המרוחקת או הגרסה המקומית)

**אין כאן כלל ברור, לפעמים הגרסה הסופית תהיה שילוב של שני השינויים שיתווספו ידנית על ידי הממזג.**

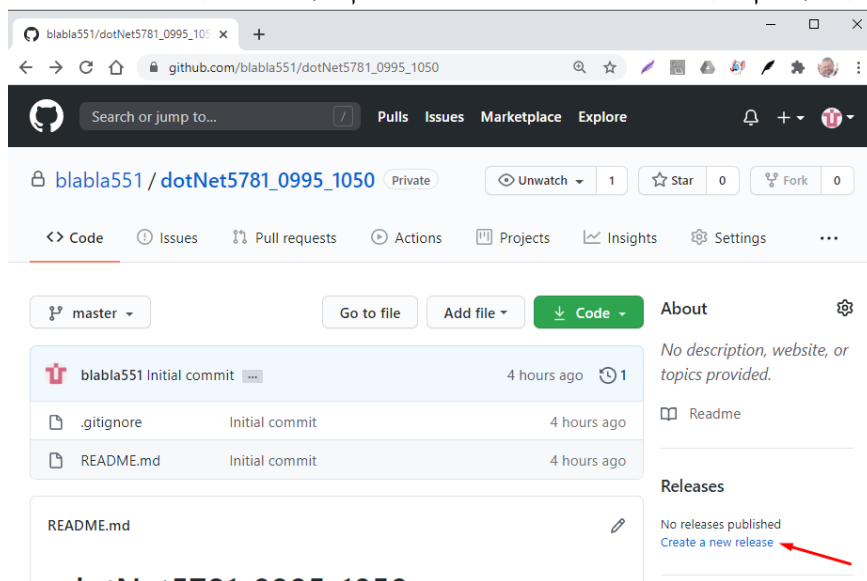
8. כשמסיימים את המיזוג – צריך לבחור Commit merge, לרשום הערה, לבצע Commit, ואז אפשר להמשיך בסנכרון.

9. בשלב הזה, בחרו בשורה נוספת, וחזרו על כל התהליך, אבל התחלפו בתפקיד של "מי דוחף ראשון" ו"מי ממזג את השינויים".

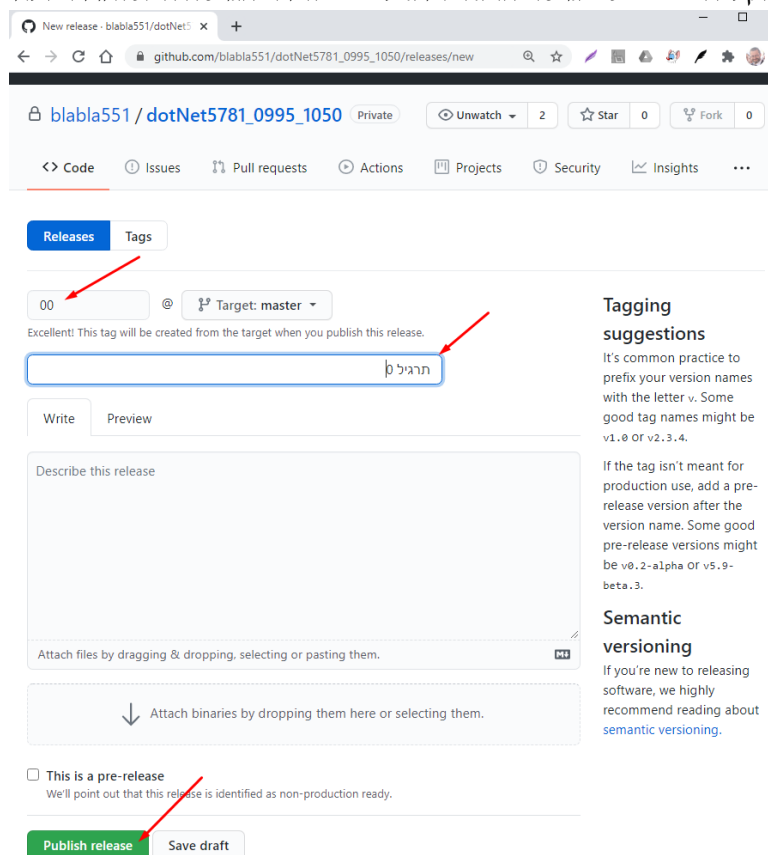
## א. יצירת גרסת שחרור (Release) עם תג (Tag)

לפני הגשה יש ליצור גרסת שחרור במאגר באתר:

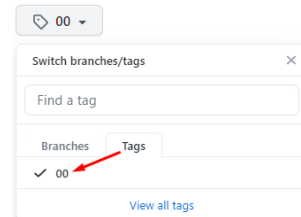
1. בצעו את השינוי האחרון ועדכנו אותו עם כיתוב "Ex0 Final commit", בצעו סנכרון העדכון לאתר.
2. לחצו על הקישור Create a new release בצד ימין בעמוד המאגר:



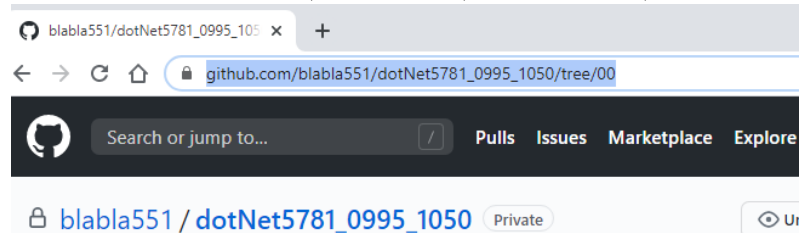
3. הקלידו "00" בשם גרסת התג ו"תרגיל 0" בכותרת הגרסה המשוחררת ולחצו על כפתור ירוק Publish release:



4. בעמוד הבא לחצו על **Code** > בסרגל הכלים.
5. כעת, במקום master מופיע סימן תגית עם כיתוב "00".
6. נפתח תפריט יורד ובו לשוניות Branches ו-Tags. לחצו על לשונית Tags (כך תמיד תוכלו להציג את הגרסה המשוחררת המיוחלת).
7. בחרו בתגית 00.



8. בעמוד שמוצג בכתובת הדף יופיע משהו דמוי: <https://github.com/account-name/repo-name/tree/00>. עליכם להעתיק את הכתובת (הקישור) הזו לטקסט בתיבת ההגשה.



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