מדריך למתחילים: סביבת העבודה Visual Studio וכתיבת קוד ב-C#

פרק א': מהי סביבת פיתוח (IDE) ומה תפקיד הקומפיילר?

תוכנת **Visual Studio** היא סביבת פיתוח משולבת (IDE - Integrated Development Environment) מקצועית המרכזת את כל הכלים הנחוצים למתכנת תחת קורת גג אחת:

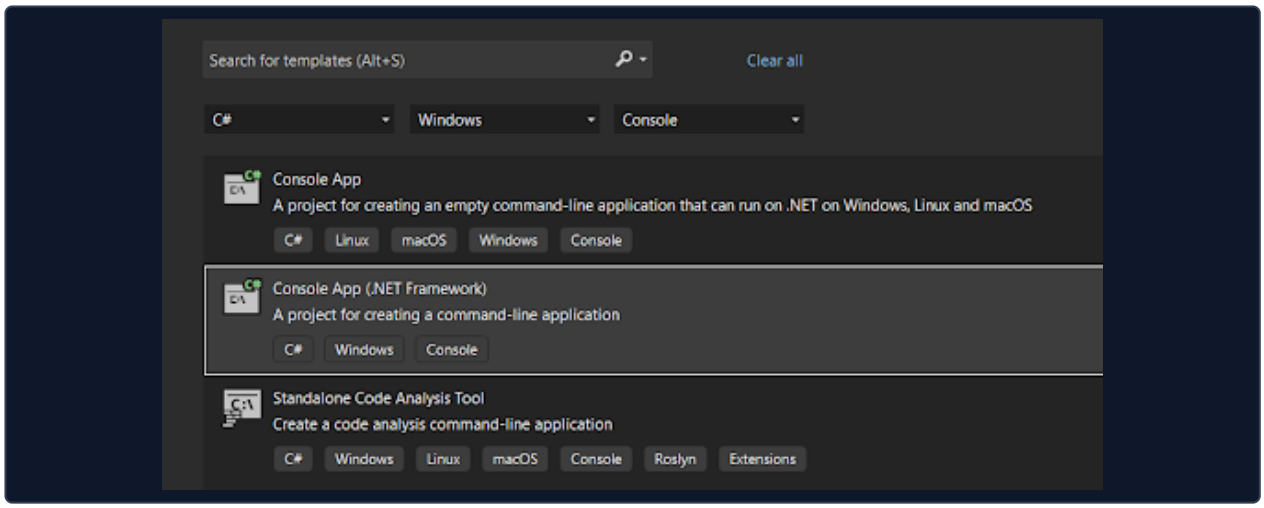
- **עורך הקוד (Code Editor):** חלון חכם לכתיבת הטקסט של התוכנית. הוא צובע מילות מפתח, משלים שמות פקודות ומסמן שגיאות תחביר באופן מיידי בזמן ההקלדה.
- **קומפיילר (Compiler - מהדר):** הרכיב שבודק את תקינות הקוד ומתרגם אותו משפת C# לשפת מכונה שהמעבד מבין ומסוגל להריץ. אם נפלה שגיאה (כגון חוסר בנקודה-פסיק, אות גדולה/קטנה שגויה או סוגר שלא נסגר), הקומפיילר יציג הודעת שגיאה מדויקת ולא יאפשר את הרצת התוכנית עד לתיקון.

פרק ב': פתיחת פרויקט חדש צעד-אחר-צעד

שלב 1 בחירת תבנית פרויקט עם Main מובנה

לאחר הפעלת התוכנה ולחיצה על `Create a new project`, נבחר את תבנית הפרויקט המתאימה ביותר ללימוד יסודות התכנות:

- במסננים העליונים נבחר בשפת **C#** ובסוג פרויקט **Console**.
- נסמן במפורש את התבנית: **Console App (.NET Framework)**.
- **מדוע תבנית זו?** תבנית זו יוצרת באופן אוטומטי ומובנה את שלד התוכנית המלא עם המחלקה `Program` ופונקציית ה-`Main` הקלאסית, ומאפשרת מעקב נוח ומסודר אחר גבולות התוכנית.
- לסיום שלב זה נלחץ על כפתור **Next**.

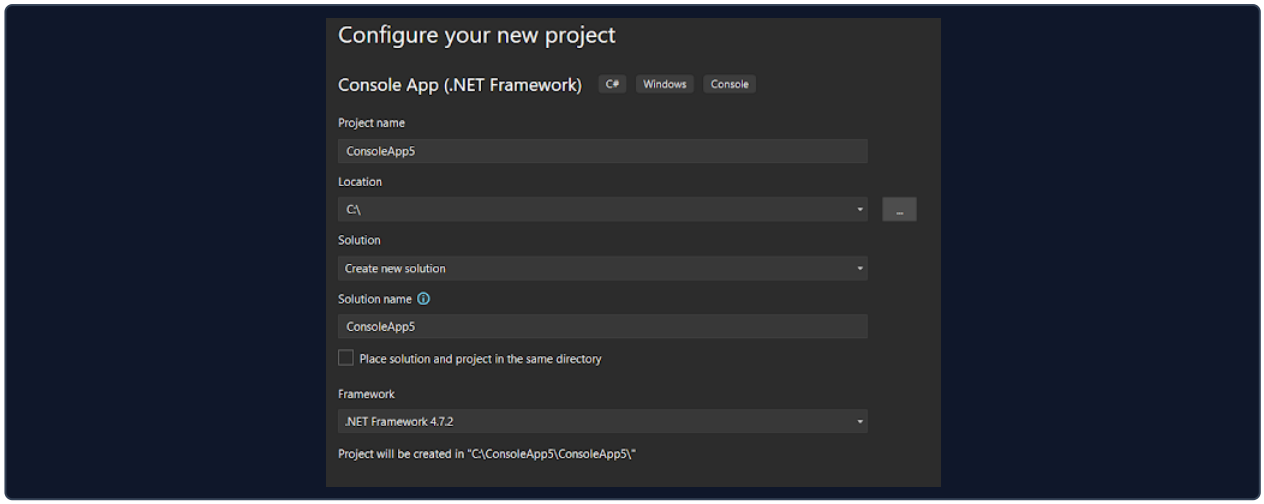


איור 1: סינון לפי C# ו-Console, ובחירה בתבנית Console App (.NET Framework)

שלב 2 הגדרת שם הפרויקט ונתיב השמירה במחשב

בחלון `Configure your new project` נקבע היכן יישמר הפרויקט ובאיזה שם:

- **Project name (שם הפרויקט):** נקליד שם ברור באנגלית, ללא רווחים, המבטא את נושא השיעור (לדו': `Lesson2_Variables` או `ConsoleApp5`).
- **Location (מיקום השמירה):** נתיב התיקייה בדיסק הקשיח. מומלץ לבחור תיקייה מסודרת ונוחה (למשל כונן `C:\` או תיקיית לימודים ייעודית) כדי שנוכל לאתר בקלות את קבצי הפרויקט בהמשך.
- לסיום נלחץ על כפתור **Create** ליצירת הפרויקט ופתיחת סביבת העבודה.



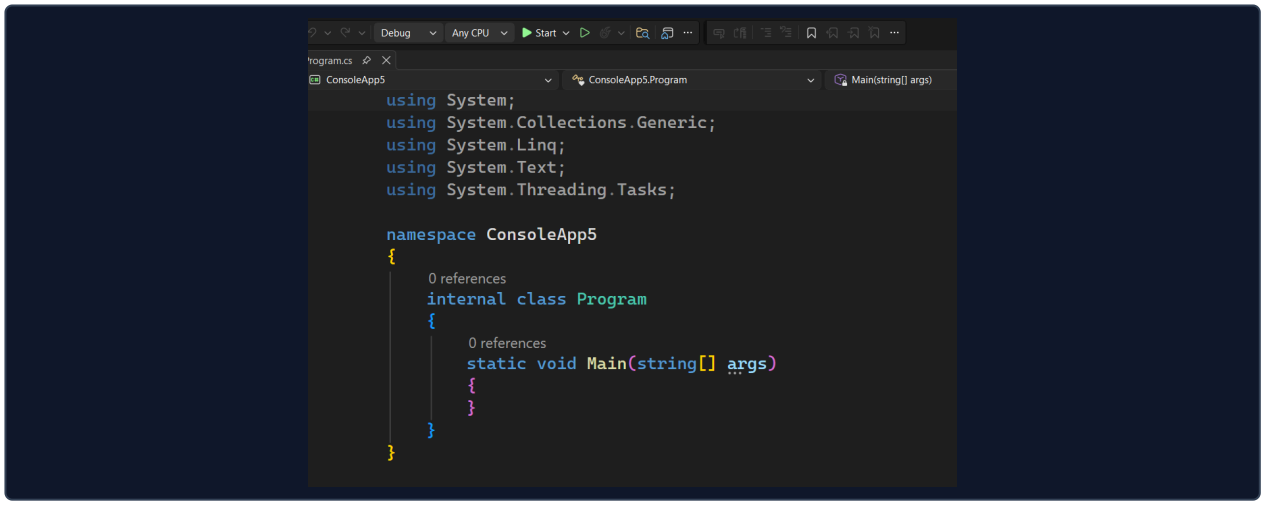
איור 2: קביעת Project name (שם הפרויקט) ו-Location (תיקיית השמירה)

פרק ג': היכרות עם מסך העבודה והרצת התוכנית

שלב 3 גבולות ה-Main וכפתור ההרצה Start

לאחר פתיחת הפרויקט נגיע אל מסך העבודה הראשי בקובץ `Program.cs`:

- **גבולות ה-Main:** הפונקציה `static void Main(string[] args)` היא נקודת ההתחלה של התוכנית. כל שורות הפקודה שאנו כותבים חייבות להיכתב אך ורק בין הסוגריים המסולסלים `{ }` שלה!
- **לחצן ההרצה (Start):** בסרגל הכלים העליון מופיע חץ ירוק עם הכיתוב `Start`. לחיצה עליו מפעילה את הקומפיילר, בונה את התוכנית ומריצה אותה.
- **הרצה מומלצת במקלדת:** מומלץ מאוד להריץ באמצעות המקשים `Ctrl + F5` – הרצה זו שומרת את חלון השחור (הקונסולה) פתוח בסיום ומאפשרת לצפות בפלט בנוחות.



איור 3: סרגל הכלים העליון (חץ ירוק Start) ופונקציית ה-Main בתוך המחלקה Program

פרק ד': חשיבות ההזחה (Indentation) ומבנה הקוד התקני

בתכנות, **הזחה (Indentation)** היא הרווח (Tab) שאנו משאירים מתחילת השורה לפני כתיבת הפקודה. ההזחה אינה משנה לקומפיילר, אך היא **קריטית עבור המתכנת**: היא מציגה בצורה ויזואלית ברורה אילו פקודות שייכות לתוך ה-`Main`, מונעת טעויות של סגירת סוגריים במקום הלא נכון, והופכת את הקוד לנקי ומקצועי.

```
namespace ConsoleApp5
{
    internal class Program
    {
        static void Main(string[] args)
        {
            // Write your program code strictly inside Main:
            int age = 16;
            string name = "Dana";

            Console.WriteLine("Name: " + name);
            Console.WriteLine("Age: " + age);
        }
    }
}
```

מפת רמות ההזחה בקוד:

רמת הזחה	כמות רווח / Tab	מרכיב הקוד
רמה 1 (חיצונית)	ללא רווח (צמוד לשמאל)	`namespace ConsoleApp5` והסוגריים שלו `{ }`
רמה 2 (פנימית 1)	Tab אחד (4 רווחים)	הגדרת המחלקה: `internal class Program`
רמה 3 (פנימית 2)	שני Tabs (8 רווחים)	הפונקציה הראשית: `static void Main(...)`
רמה 4 (גוף התוכנית)	**שלושה Tabs (12 רווחים)**	**פקודות התוכנית והמשתנים שלנו (נכתבות אך ורק כאן!)**

פרק ה': כללי זהב לעבודה נכונה והימנעות משגיאות

- **שמירה על זוגות סוגריים:** לכל סוגר פותח `{` חייב להיות סוגר סוגר `}` מקביל באותה רמת הזחה.
- **סיום שורה בנקודה-פסיק:** כל פקודת ביצוע או הגדרת משתנה ב-C# חייבת להסתיים בסימן `;`.
- **רגישות לאותיות גדולות וקטנות:** C# מבדילה בין אותיות גדולות לקטנות (למשל: `Main` ולא `main`, `Console` ולא `console`).
- **יישור אוטומטי בתוכנה:** אם הקוד התבלגן, לחיצה על המקשים `Ctrl + K` ולאחריהם `Ctrl + D` תסדר ותיישר באופן אוטומטי את כל ההזחות בקובץ!