

# תרגיל מס' 1 – Assemblies ושפת C#

## חלק 1 – גשש בלש

### מטרות

- הטמעת המושגים הבסיסיים בעולם הפיתוח עם ה- .Net Framework. כגון:
  - PE
  - Assembly
  - MSIL
  - Metadata
- הכרת הכלי ildasm שמהווה חלק מסט הכלים שמגיעים עם ה- .NET Framework.
- חשיפה לקוד MSIL
- חשיפה ל- Assembly Manifest
- הבנת היתרונות והחסרונות שבקוד מנוהל

### ידע נדרש

- הכרת המבנה של Assembly והמושגים השונים
- הכרת תחביר בסיסי בשפה הנתמכת ע"י ה- CLR (כדוגמת C#) כגון:
  - מחלקות - Classes
  - מרחב שמות – namespace
  - מתודות סטטיות
  - פרמטרים למתודות
  - משתני ערך (int, bool)
  - מחלקת Console
- עבודה עם הכלי – ildasm (Intermediate Language Disassembler)

### הכינותי מראש

- קובץ ה- zip (שמכיל את המסמך הזה) מכיל גם קובץ בשם Ex01.exe
- מחשב שמוותקן עליו .NET Framework. לצורך הרצת התוכנית והרצת הכלי ildasm.

### התרגיל

נתון לכם קובץ Ex01.exe שהוא אפליקציה דוט-נטית.

בעת הרצת הקובץ תתבקשו להכניס שם משתמש וסיסמא על מנת להמשיך בפעולת התוכנית .

את שם המשתמש והסיסמא תגלו אחרי פעולות בילוש קצרות על הקובץ בעזרת הכלי לניתוח assemblies

שנקרא ildasm

תזכורת:

יש להריץ חלון console בסביבת דוט-נט (Developer Command Prompt for Visual Studio).

כדי להריץ אותו, יש לחפש אותו בעזרת תפריט החיפוש של Windows

(לפתוח את החיפוש ולכתוב Developer Command – הוראות מלאות ניתן למצוא [כאן](#))

ב- Command Prompt יש להגיע לתיקייה בה נמצא הקובץ Ex01.exe, ואז לכתוב את השורה הבאה –

```
[the dir of the exe file]>ildasm Ex01.exe
```

המשך בעמוד הבא..

---

**יש להפעיל ildasm על הקובץ ולענות על השאלות הבאות:**

1. האם הקובץ הזה הוא .NET Assembly? כן / לא (יש לסמן את התשובה הנכונה)

האם קובץ זה מהווה .NET PE? כן / לא (יש לסמן את התשובה הנכונה)  
נימוק:

---



---



---



---

**תארו את ה- Assembly:**

- a. מה שמו? \_\_\_\_\_
- b. מה גרסתו? \_\_\_\_\_
- c. באיזה חלק של ה- assembly ישנן התשובות לסעיפים a, b? \_\_\_\_\_
- d. באילו assemblies אחרים הוא משתמש? (יש לתת שם וגרסה של כל אחד מהם)
- שם: \_\_\_\_\_ גרסא: \_\_\_\_\_
- שם: \_\_\_\_\_ גרסא: \_\_\_\_\_
- שם: \_\_\_\_\_ גרסא: \_\_\_\_\_

**2. ניתוח ה- MSIL**

עבור כל אחד מהטיפוסים בקוד, יש לתת תיאור מלא:

- a. סוג הטיפוס (class, struct, enum)
- b. שם הטיפוס
- c. רשימת הרכיבים (members) המוגדרים בטיפוס ותיאורם. התיאור כולל:
- i. Name
- ii. Type
- iii. Static / Instance
- iv. Access Modifier (public/private/protected)
- בנוסף, אם הרכיב הוא מתודה (פונקציה):
- v. סוג הערך המוחזר (return value type)
- vi. רשימת הפרמטרים שהיא מקבלת ותיאורם

**יש למלא טבלה כזו:**

Type (Struct/Class/Enum)	Name	Members (methods, fields)
Class	SomeClass	<ul style="list-style-type: none"> <li>public static int SomeMethod(float)</li> <li>...</li> </ul>
		<ul style="list-style-type: none"> <li></li> </ul>

3. מהם שם המשתמש והסימא הנדרשים להרצת התוכנית? (כאן נדרשת עבודת בילוש קטנה...)

4. הריצו את התוכנית ותוכלו להבין חלק מהמצופה בחלק מהתרגילים הבאים.

## חלק 2 – תכנות בסיסי

### מטרות

- פיתוח אפליקציות דוט נט בעזרת ה- Visual Studio
- תרגול בכתיבת תוכנית בתחביר C#
- עבודה עם קלט/פלט בסביבת Console
- היכרות עם המחלקות string, int, float, char, math, StringBuilder

### ידע נדרש

- היכרות עם כלי הפיתוח Microsoft Visual Studio
- היכרות עם אלמנטים בסיסיים ותחביר בסיסי של שפת C#

### הכינותי מראש

- Microsoft Visual Studio מותקן על המחשב.

### התרגיל

יש ליצור Solution חדש **לפי השם שמוגדר בהוראות ההגשה (!!!)**.  
שימו לב, כדי ליצור קודם כל Solution ריק ורק אח"כ להוסיף לו פרויקטים:

File→New→Project→Blank Solution

עבור כל אחד מהסעיפים הבאים, יש ליצור פרוייקט נפרד בתוך ה- Solution.

[Right-Click on the solution node in the tree] → Add → New → Project... → Visual C# → Windows → Empty Project

שמות הפרויקטים יהיו בפורמט Ex01\_0X (כאשר X הוא מספר הסעיף).

- **שימו לב למסמך שמתאר באופן מפורט כיצד יש להכין את הפרויקטים לטובת התרגיל הזה.**  
**יש לעקוב אחר ההוראות בו באדיקות!**

## 1. סדרות בינאריות

כתבו תוכנית הקולטת מהמשתמש 3 מספרים **בפורמט בינארי** בני 8 ספרות כל אחד. **אחרי כל מספר, המשתמש מקיש enter.** (קלט לא חוקי ידווח למשתמש כלא חוקי והתוכנית תמתין לקלט חוקי). התוכנית תמיר את המספרים שהוכנסו למספרים עשרוניים ותדפיס אותם בפורמט עשרוני **בסדר יורד**.

בנוסף, התוכנית תדפיס סטטיסטיקה לגבי הקלט:

- כמה מהמספרים הם חזקות של 2
- מה מספר האפסים / אחדות הממוצע (כלומר סוכמים את כל האפסים/אחדות שהוכנסו מכל הקלטים ומחלקים במספר הקלטים)
- כמה מהמספרים שנקלטו מתחלקים ב-4 ללא שארית
- בכמה מהמספרים, הספרות בייצוג עשרוני מהוות סדרה יורדת ממש.
- כמה מהמספרים שנקלטו, הספרות שלהם, בייצוג העשרוני, מהוות פלינדרום (למשל 121)

**לספק דוגמאות הרצה (צילומי מסך) על 3 סוגי קלטים, וביניהם הקלט הבא:**

01000000, 01111011, 01111001

(המספרים העשרוניים הם: 121, 123, 64 – אחד מהם חזקה של 2, ב-1 מהם יש סדרה יורדת ממש בספרות, אחד מהם הוא פלינדרום, אחד מתחלק ב-4 ללא שארית)

## 2. יהלום למתחילים

כתבו תוכנית שמדפיסה למסך מעוין של כוכביות באופן הבא:

```

*
***
*****
*****
*****
*****
*****
*****
*****
*****

```

**יש לממש בעזרת רקורסיה.**

## 3. יהלום למתקדמים

כתבו תוכנית שתעבוד כמו תוכנית יהלום למתחילים, אך תקבל את גובה המעוין המבוקש כקלט מהמשתמש (יש להעיר למשתמש על קלט לא תקין!).

(נסו לממש זאת ע"י הפנייה ל- assembly שיצרתם בסעיף 2 (יהלום למתחילים) ע"י כך שתוסיפו Reference מהפרויקט של הסעיף הזה לפרויקט של הסעיף הקודם).

במקרה של קלט זוגי, לשיקולכם: להוסיף שורה נוספת באמצע או להוסיף/להוריד 1 מהקלט.

יש לספק דוגמאות הרצה (צילומי מסך) על הקלטים: 4, 5, 7, 8

## 4. ניתוח מחרוזת

כתבו תוכנית המקבלת כקלט מחרוזת בגודל של 6 תווים (למשל: **abccba**) ומספקת את הפרטים הבאים:

1. האם המחרוזת היא פלינדרום (למימוש בעזרת רקורסיה)
  2. אם מדובר במספר - האם הוא מתחלק ב-3 ללא שארית..
  3. אם מדובר במחרוזת באנגלית – את מספר האותיות הגדולות שיש בה (uppercase)
- שימו לב, המחרוזת חייבת להיות מורכבת מאותיות באנגלית בלבד (גדולות או קטנות), או ממספרים בלבד. מחרוזות כגון 4er5er אינן חוקיות. **שימו לב, אותיות גדולות נחשבות שונות מאותיות קטנות.**

**5. סטטיסטיקות מספרים**

כתבו תוכנית המקבלת מהמשתמש מספר שלם בן 6 ספרות ומספקת את הפרטים הבאים:

1. כמה מהספרות גדולות מספרת האחדות במספר
  2. מה הספרה הקטנה ביותר במספר
  3. כמה מהספרות במספר מתחלקות ב-3 ללא שארית
  4. מה ממוצע הספרות
- (שימו לב, הקלט 000123 הוא קלט תקין, סיפרת האחדות היא 3 והיא גם הגדולה ביותר, 0 ספרות במספר קטנות ממנה, 4 ספרות מתחלקות ב-3 ללא שארית, ממוצע הספרות הוא 1) יש להעיר למשתמש על קלט לא תקין ולבקש קלט חדש.

**שימו לב:**

1. חובה להשתמש במחלקות ובמתודות הבאות:
  - `StringBuilder` – לא הכרנו. דורש מחקר עצמאי.
  - `Math` – לא הכרנו. דורש מחקר עצמאי.
  - `char` – לא הכרנו. דורש מחקר עצמאי.
  - `string.Format` (הוסבר בסרטון ההדרכה)
  - `int.TryParse` – יש דוגמא בדוגמת הקוד באתר.
  - מתודות נוספות של המחלקה `string` שיכולות לעזור.
2. אין חובה להשתמש במערכים ועדיף להימנע משימוש במערכים כי טרם למדנו להשתמש במערכים.
3. קישורים לקטעי הקוד שהוצגו בהרצאות ובסרטוני ההדרכה נמצאים במסמך הקורס תחת קטגוריית "דוגמאות קוד".
4. את ההמרה של בינארי לעשרוני יש לממש בעצמכם, ולא להשתמש במתודה קיימת שמבצעת זאת.
5. טרם למדנו נושאים הקשורים במחלקות, לכן בשלב זה יש לממש את הקוד בעזרת מתודות סטטיות בלבד, וללא חלוקה למחלקות. כל המתודות יוגדרו במחלקה `Program` של כל אחד מהסעיפים.
6. אין לממש את כל התוכנית במתודה `Main` אלא לקרוא מהמתודה `Main` למתודות אחרות. התפקיד של המתודה `Main` הוא להוות נקודת כניסה לתוכנית ותו-לא.
7. יש להפגין חלוקה נכונה למתודות.

**הגשה**

- ההגשה תתבצע לתיבת המייל של התרגילים של הקורס כמפורט במסמך הוראות הגשה שניתן למצוא במסמך התכנים של הקורס. נקודות ירדו למי שלא יפעל בדיוק לפי ההוראות.
- אין צורך לתעד בפורמט XML.
- יש לספק דוגמאות הרצה לכל הסעיפים ע"י הדבקת תצלומי מסך של ה-console למסמך הוורד שאותו אתם מגישים כפיתרון לחלקו הראשון של התרגיל.
- על קובץ הוורד להימצא באותה תיקייה של קובץ ה-`sln`.
- שמו של הקובץ יהיה: `Ex01_ScreenShots.doc`
- יש לעמוד בתקנים לכתיבת קוד כפי שמפורט במסמך הרלוונטי.
- אי עמידה בתקנים גוררת הורדה בניקוד.
- כל יום איחור יוריד 4 נקודות מציון התרגיל.
- ניתן להיעזר בקבוצת הפייסבוק של הקורס כדי לשאול שאלות בנוגע לתרגיל.
- נא להימנע מהעתקות!