

# 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 שמותקן 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 בכתובת 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 → Other Project Types → Visual Studio Solutions → 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 הפרוייקטים יהיו בפורמט B20\_Ex01\_X (כאשר X הוא מספר הסעיף).

- **באתר הקורס יש מסמך שמתאר באופן מפורט כיצד יש להכין את הפרוייקטים לטובת התרגיל הזה. יש לעקוב אחר ההוראות בו באדיקות!**

**1. סדרות בינאריות**

כתבו תוכנית הקולטת מהמשתמש 3 מספרים שלמים וחיוביים בני 9 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. בנוסף, התוכנית תדפיס סטטיסטיקה לגבי הקלט:

- מה מספר האפסים / אחדות הממוצע בכל מספר שהוכנס (כלומר צריך לסכום את כל האפסים שהוכנסו מכל הקלטים חלקי מספר הקלטים וכך גם לבצע על האחדות)
- כמה מהמספרים שנקלטו, הם חזקות של 2.
- כמה מהמספרים שנקלטו, הספרות שלהם, בייצוג העשרוני, מהוות סדרה עולה ממש
- מי מהמספרים הוא הגדול ביותר, ומי הקטן ביותר.

יש לספק דוגמאות הרצה על הקלטים הבאים:

- א. 111010100 , 111010010 , 111011000 (המספרים הם: 468, 466, 472 – אף אחד מהם אינו חזקה של 2, ב-1 מהם יש סדרה עולה ממש בספרות, הגדול הוא 472, הקטן הוא 466)
- ב. 001001110 , 010000000 , 001110000
- ג. 101111110 , 110011110 , 011010110

**2. שעון חול למתחילים**

כתבו תוכנית שמדפיסה למסך שעון חול של כוכביות באופן הבא:

```

* * * * *
* * *
*
* * *
* * * * *
```

**יש לממש בעזרת רקורסיה.**

**3. שעון חול למתקדמים**

כתבו תוכנית שתעבוד כמו תוכנית שעון החול למתחילים, אך תקבל את גובה השעון המבוקש כקלט מהמשתמש (יש לטפל בקלט שגוי!).

(נסו לממש זאת ע"י הפנייה ל- assembly שיצרתם בסעיף 2 (שעון חול 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. יש לספק דוגמאות הרצה על הקלטים: 4, 6, 7, 8

**4. ניתוח מחרוזת**

כתבו תוכנית המקבלת כקלט מחרוזת בגודל של 8 תווים (למשל: **babcttcab**) ומספקת את הפרטים הבאים:

1. האם המחרוזת היא **פלינדרום** (יש לממש בעזרת רקורסיה)
  2. אם מדובר במספר – האם הוא מתחלק ב-5 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 2b45bb2r אינן חוקיות. **שימו לב, אותיות גדולות נחשבות שונות מאותיות קטנות.**

**5. סטטיסטיקות מספרים**

כתבו תוכנית המקבלת מהמשתמש מספר שלם וחיובי בן 9 ספרות ומספקת את הפרטים הבאים:

1. מה הספרה הגדולה ביותר במספר
2. מה הספרה הקטנה ביותר במספר
3. כמה מהספרות במספר מתחלקות ב-3 ללא שארית
4. כמה מהספרות גדולות מספרת האחדות במספר (שימו לב, הקלט 0000123 הוא קלט תקין)

**שימו לב:**

1. **חובה** להשתמש במחלקות ובמתודות הבאות:
  - `StringBuilder` – לא הכרנו. דורש מחקר עצמאי.
  - `string.Format`
  - `int.TryParse` – לא הכרנו. יש דוגמא בדוגמת הקוד באתר.
  - `Math` – לא הכרנו בשיעור. דורש מחקר עצמאי.
  - `char` – לא הכרנו בשיעור. דורש מחקר עצמאי.
  - מתודות נוספות של המחלקה `string` שיכולות לעזור.
2. אין צורך להשתמש במערכים.
3. קטעי קוד שכתבנו בכיתה / בסרטוני הוידאו נמצאים באתר תחת קטגוריית "דוגמאות קוד".
4. את הסעיף של בינארי לעשרוני יש לממש בעצמכם, ולא להשתמש במתודה קיימת שמבצעת זאת.
5. **טרם למדנו נושאים הקשורים במחלקות בסי-שארפ לכן בשלב זה, יש לממש את הקוד בעזרת מתודות סטטיות בלבד, וללא חלוקה למחלקות.**
- כל המתודות יוגדרו במחלקה `Program` של כל אחד מהסעיפים.**
- אין לממש את כל התוכנית במתודה `Main` אלא לקרוא מהמתודה `Main` למתודות אחרות.**
- התפקיד של המתודה `Main` הוא להוות נקודת כניסה לתוכנית ותו-לא.**
- יש להפגין חלוקה נכונה למתודות.**

**הגשה**

- יש להגיש עד ליום ד', 28 לאפריל 2020, בשעה 22:00.
- ההגשה תתבצע לתיבת המייל של התרגילים של הקורס כמפורט במסמך הוראות הגשה שניתן למצוא באתר הקורס. נקודות ירדו למי שלא יפעל בדיוק לפי ההוראות.
- אין צורך לתעד בפורמט XML (עוד לא למדנו)
- יש לספק דוגמאות הרצה לכל הסעיפים ע"י הדבקת תצלומי מסך של ה-console למסמך וורד אותו אתם מגישים כפיתרון לחלקו הראשון של התרגיל.
- על קובץ ה-`.doc` להימצא באותה תיקייה של קובץ ה-`.sln`.
- **שמו של הקובץ יהיה: `Ex01_ScreenShots.doc`**
- **יש לעמוד בתקנים לכתיבת קוד כפי שמפורט במסמך הרלוונטי שניתן למצוא באתר הקורס.**
- 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 - <https://www.facebook.com/groups/dn.course.mta.19C>
- **נא להימנע מהעתקות!**

בהצלחה 😊