Matam Code Conventions

1 מבוא

Code Conventions (מוסכמות קוד) מוגדרים כאוסף כללים המנחים את סגנון כתיבת הקוד. כללים אילו אינם באים לשפר את איכות הקוד בצורה ישירה אך מקלים על הבנתו וקריאותו, בעיקר ע"י אנשים נוספים מלבד הכותבים המקוריים.

שימוש במוסכמות לכתיבת הקוד הוא בין השאר בעל היתרונות הבאים:

- מוסכמות בכתיבת הקוד מקלות על הבנת הקוד ע"י מתכנתים (המקוריים ומתכנתים נוספים). תחזוקת הקוד לוקחת עד 80% מכל העבודה על התוכנה , ולכן קריאות הקוד היא חשובה ביותר.
- מוסכמות כתיבת קוד מקלות על מציאת משתנים ופונקציות וחיפושן.

שימו לב! העמידה במוסכמות המובאות כאן היא חובה בתרגילי הבית בקורס. אי עמידה בכללים אלו תגרור הורדה של נקודות מהציון.

2 שמות מזהים (משתנים, פונקציות, קבועים, טיפוסים)

2.1 שמות משתנים + שמות פונקציות

צריכים להכיל אותיות קטנות בלבד ואות גדולה בתחילת כל מילה שאינה ראשונה (סגנון זה מכונה "camelCase"):

```
int doStuff(int number)
{
    int myNumberVariable = 0
    return myNumberVariable + number
}
```

השמות צריכים להיות ברורים ולעמוד בכללים הבאים:

- **ללא קיצורים** – למשל movie (טוב) ולא mov (רע).
- **ללא השמטת תנועות** (vowels) – למשל controller (טוב) ולא cntrllr (רע).
- מותר להשתמש בקיצורים הבאים **בלבד** (גם כחלק משם משתנה):

שם	משמעות	שם	משמעות
k,j,i	עבור משתני אינדקס (זה בסדר גם לשכפל kk,jj,ii ע"מ למנוע בלבול עם קבועים מרוכבים)	*dest, src*	destination, source
ptr	עבור מצביעים	*fd*	עבור file descriptor
n	עבור מס' שלם (אם אין שם מתאים בעל יותר משמעות)	*str*	עבור string
		len	עבור length
num, val	עבור number, value בהתאמה	*arg*	עבור argument
in/out	עבור input/output למשל: val_out, num_in	*tmp, temp*	עבור temporary (בהיעדר שם מתאים).
max, min	עבור מינימום ומקסימום (כדאי – כחלק משם כדי למנוע התנגשות עם הפונק' המתמטיות)	*func, diff*	function, difference

2.2 שמות קבועים

שמות עבור macros וקבועים המוגדרים ב-#define יהיו באותיות גדולות בלבד ועם underscore בין מילים.
למשל:

```
#define MAX_LEN 120
```

או

```
const int MAX_LEN = 120
```

2.3 שמות טיפוסים

שמות טיפוסים המוגדרים בעזרת typedef, מבנה או מחלקה:

- יוגדרו ללא קיצורים ויתחילו באות גדולה (ואות גדולה בתחילת כל מילה בשם הטיפוס)
- שמות הטיפוסים יהיו ביחיד ולא ברבים (למשל Movie אבל לא Movies).
- דוגמאות:

```
typedef struct {
...
} MovieReview;

typedef int Length;

typedef enum { ... } ErrorCode;
```

2.4 שמות שדות מחלקה

שמות שדות מחלקה צריכים להתחיל ב"_m" ולאחר מכן camelCase כמו שמות משתנים

```
class Point {
      int m_x;
      int m_y;
...
};
```

3 שפה

- את כל ההערות ושמות המזהים יש לכתוב באנגלית בלבד!
  אין לכתוב הערות בעברית.
  אין לקרוא למשתנים בעזרת תעתיק, למשל "histabrut" במקום "probability".
- הקוד צריך לתעד את עצמו. המטרה היא שיידרש מינימום תיעוד לקוד:
  Comment Only what the Code Cannot Say – Chapter 17 in 97 Things Every Programmer Should Know, by Kevlin Henney

4 מבנה הקוד

- חובה להשתמש בהזחות (indentation) לכל בלוק (קטע קוד מוקף או "אמור להיות" מוקף סוגריים מסולסלים), כפי שנלמד בקורס מבוא:

```
int doStuff(int n)
{
    if (n > 5)
    {
        return 0;
    }
    return 1;
}
```

- בכל שורת קוד **תופיע רק פקודה (statement) אחת**.
  אין להוסיף פקודות נוספות באותה שורה לאחר ה-";".
  - פרט לכותרת לולאת for, כמובן.
- אורך שורת קוד לא יעלה על 120 תווים.
- אורך פונקציה לא יעלה על 50 שורות בתרגילי ה-C **ומעל 30 שורות בתרגיל ה-++C**.
  מס' זה אינו כולל שורות ריקות, הערות וסוגריים.
- חובה להשתמש בסוגריים מסולסלים לאחר כל if, else, while, for, do while (גם אם יש רק פקודה אחת). סוגר פותח יכול להופיע בשורת הכותרת (כמו בדוגמה מטה) או בשורה נפרדת, לאחריה (כמו בדוגמה למעלה). סוגר סוגר יופיע בשורה נפרדת משל עצמו (ובדוגמא מעלה צריך להיות מיושר עם הסוגר הפותח).

```
int doSomething(int n)
{
    if (n > 0) {
        return n;
    } else {
        return 0;
    }
}
```

- הסוגר הפותח והסוגר הסוגר של פונקציה תמיד יופיעו כל אחד בשורה נפרדת.