

דגשים לכתיבת קוד

- (1) שמות משמעותיים לפונקציות ולמשתנים- אין סיבה להתקמצן על תווים במחשב- תנו שם משמעותי לכל פונקציה ולכל משתנה שלכם.
- (2) פונקציות קצרות- מצופה ממכם שלא יהיו פונקציות ארוכות בקוד. במידה ויש לכם פונקציה בקוד שעוברת את ה-50 שורות קוד אתם צריכים לתת סיבה טובה לכך. השתדלו לכתוב פונקציות שלא עוברות את ה-30 שורות קוד.
- (3) תיעוד- מצורף דוגמא לכתיבת פונקציה עם תיעוד:

```
// input arguments:
// arg1 - what arg1 should contain when the function is called. Is the argument
// an input argument or a result argument?
// arg2 - what arg2 should contain when the function is called. Is the argument
// an input argument or a result argument?
//...
// result argument: what is returned? if the return type is void skip this
// line.
// 1-2 lines of summary on what the function does.
int function_name(arg1_Type arg1, arg2_Type arg2) {
    some code;
    an interesting line; // if you think a line is complicated, put a
    comment and explain the line. This shouldn't be done on every line.
    more code;
    return something;
}
```

מצופה ממכם שיהיה תיעוד לכל פונקציה בקוד.

- (4) ראש העמוד- בפרויקט זה אנחנו לא נעבוד עם קבצי h אך עדיין מצופה ממכם לסדר וארגון. מצורף דוגמא למבנה של עמוד:

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define const1 500
#define const2 (1 << 16)

int function_name_1(arguments);
void function_name_2 (arguments);
int function_name_3 (arguments);
void function_name_4 (arguments);
...
int main(int argc, char* argv[]) {
    some code;
}
// comment like showed above
int function_name_1(arguments){
    some code;
}
// comment like showed above
void function_name_2(arguments){
    some code;
}
// comment like showed above
int function_name_3(arguments){
    some code;
}
// comment like showed above
void function_name_4(arguments){
    some code;
}
}
```