

המדריך המקוצר ליצירת Class



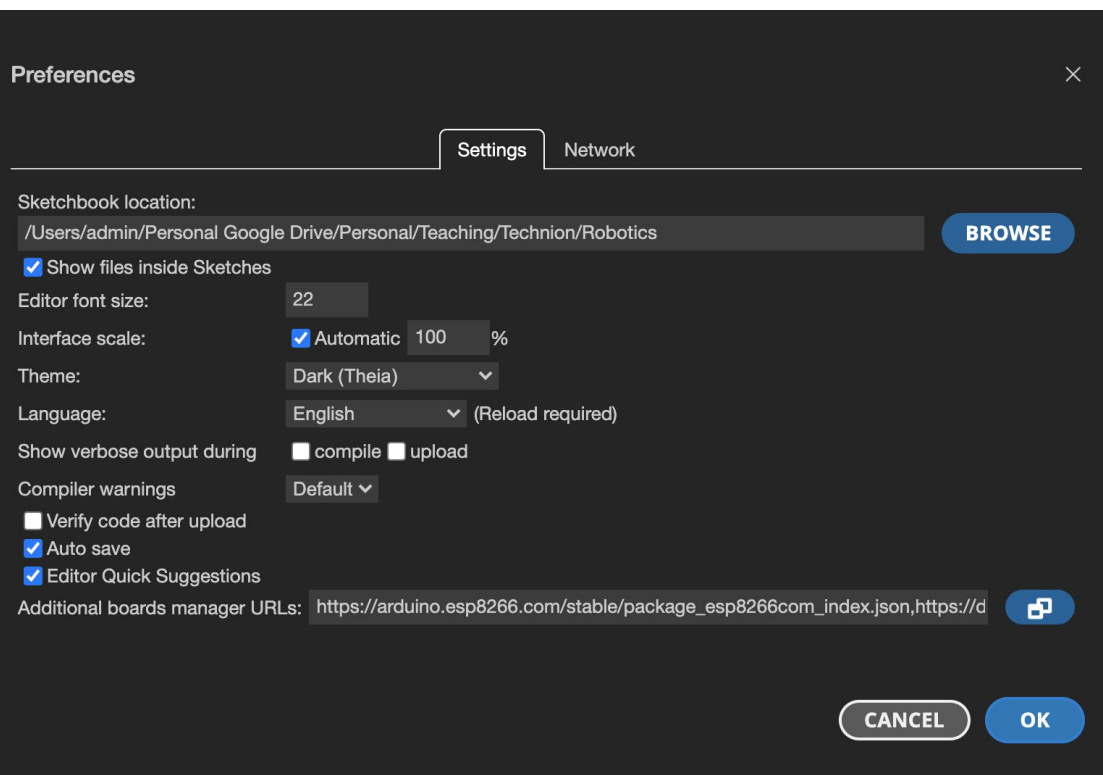
GEEKNINJA

מאת גל ארבל
ספט 2022

שלב 1

צרו תיקיה חדשה
בספריית ארדואינו
בשם `class` שלכם

טיפ! אפשר לראות היכן
ארדואינו שומר את התיקיות
ב `file\preferences`



שלב 2

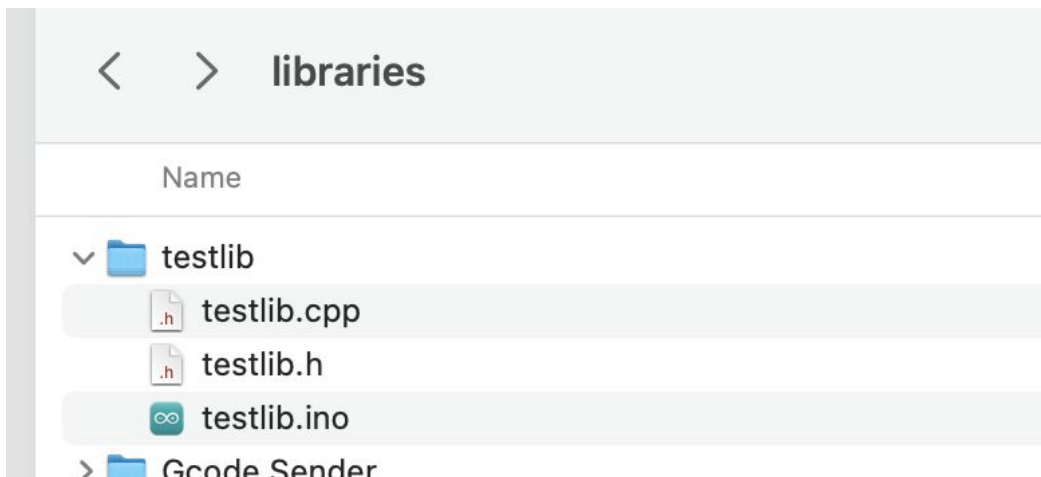
הוסיפו 3 קבצי טקסט בתיקיה
עם אותו שם. בדוגמא:
testlib

עם סיומות:


.cpp

.h

.ino



שלב 3



The screenshot shows the Arduino IDE interface. At the top, there's a toolbar with a checkmark, a right arrow, and a play button. Below it, a dropdown menu shows 'Arduino Uno'. The main workspace has three tabs: 'testlib.ino' (active), 'testlib.cpp', and 'testlib.h'. On the left, there's a sidebar with icons for file explorer, search, and other IDE functions. The code in 'testlib.ino' is as follows:

```
1  #include "testlib.h"
2
3  testlib mytest;
4
5  void setup() { }
6
7  void loop() { }
```

פתחו את תוכנת
arduino ide, ופתחו
את קובץ ה .ino.

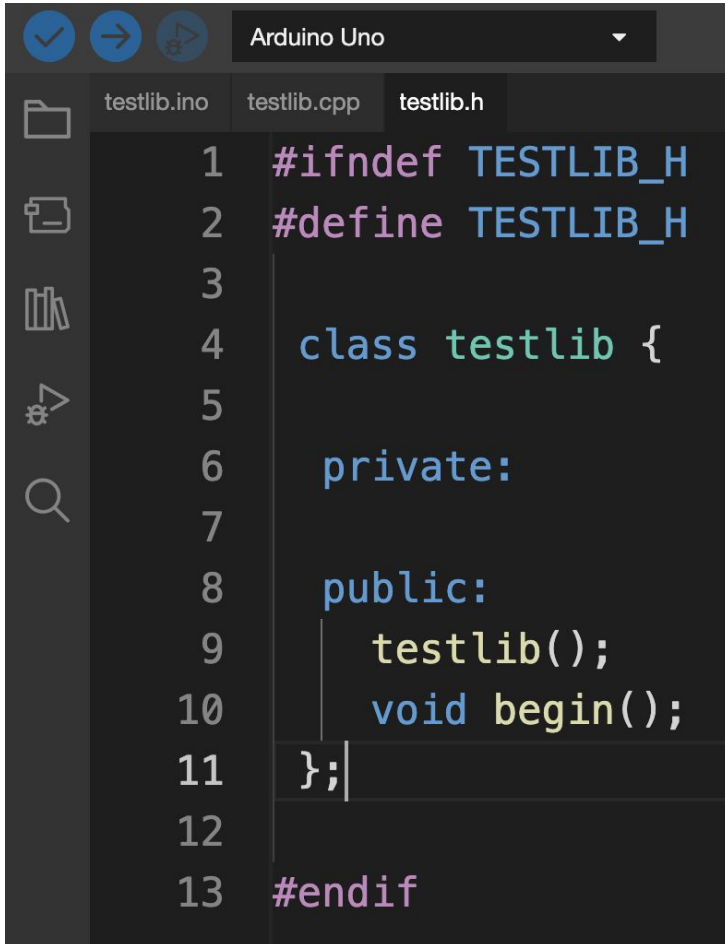
הכניסו הקוד הבא

טיפ! שימו לב ששלושת
הקבצים נפתחו בide

שלב 4

הכניסו הקוד לקובץ h.

טיפ! הפונקציות ב public וב
private נקראות **מתודות**

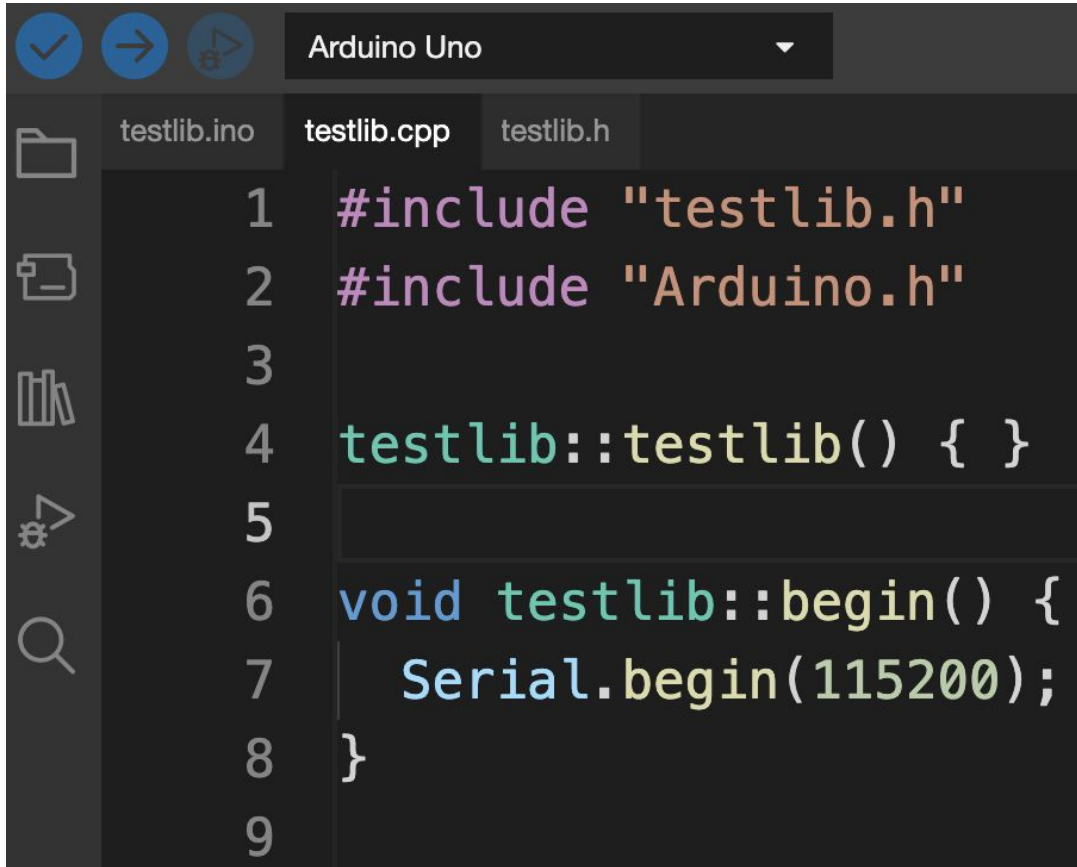


```
1  #ifndef TESTLIB_H
2  #define TESTLIB_H
3
4  class testlib {
5
6  private:
7
8  public:
9      testlib();
10     void begin();
11 };
12
13 #endif
```

שלב 5

הכניסו הקוד לקובץ cpp.

טיפ! כאן אפשר להכניס את
כל ההגדרות הקבועות של
ארדואינו, כמו pinMode



The screenshot shows the Arduino IDE interface. At the top, there's a toolbar with a checkmark, a right arrow, and a magnifying glass. Below it, a dropdown menu shows 'Arduino Uno'. The main workspace has three tabs: 'testlib.ino', 'testlib.cpp', and 'testlib.h'. The 'testlib.cpp' tab is active, displaying the following code:

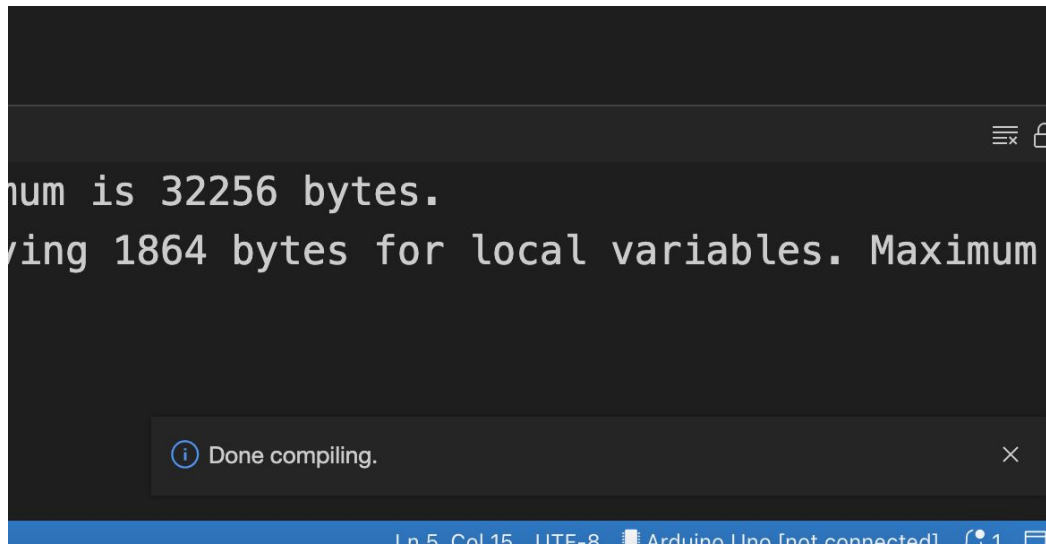
```
1  #include "testlib.h"
2  #include "Arduino.h"
3
4  testlib::testlib() { }
5
6  void testlib::begin() {
7      Serial.begin(115200);
8  }
9
```

שלב 6

קמפלו את הקוד

טיפ! (מקש קיצור:

(ctrl+r



The screenshot shows a terminal window with a dark background. The text in the terminal is white and reads: "num is 32256 bytes." followed by "Using 1864 bytes for local variables. Maximum". Below the terminal text, there is a notification box with a blue 'i' icon and the text "Done compiling." with a close button 'x'. At the bottom of the IDE, a status bar is visible with the text "Ln 5, Col 15 - UTF-8 - Arduino Uno [not connected] - 1".

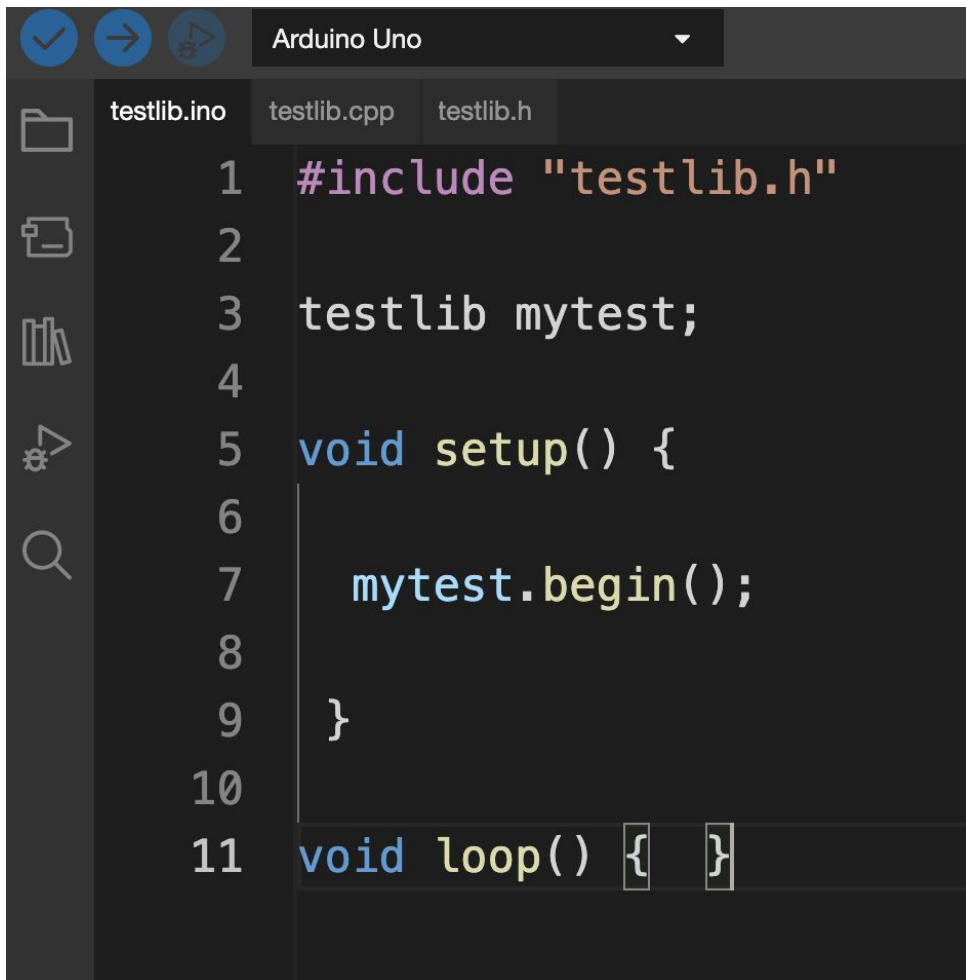
```
num is 32256 bytes.  
Using 1864 bytes for local variables. Maximum
```

Done compiling.

Ln 5, Col 15 - UTF-8 - Arduino Uno [not connected] - 1

שלב 7

הפעילו את
מתודת begin



```
Arduino Uno
testlib.ino testlib.cpp testlib.h
1  #include "testlib.h"
2
3  testlib mytest;
4
5  void setup() {
6
7      mytest.begin();
8
9  }
10
11 void loop() { }
```


שלב 8

הוסיפו מתודות/פרוצדורות כרצונכם

```
void action(int speedR, int speedL);
```

הוסיפו בקובץ h.:

```
void testlib::action(int speedR, int speedL) {
```

ובקובץ cpp.:

```
/* do something */
```

```
}
```

```
void loop() {
```

והפעילו מקובץ הסיומת in.:

```
mytest.action(255,255);
```

```
}
```



כל הכבוד!

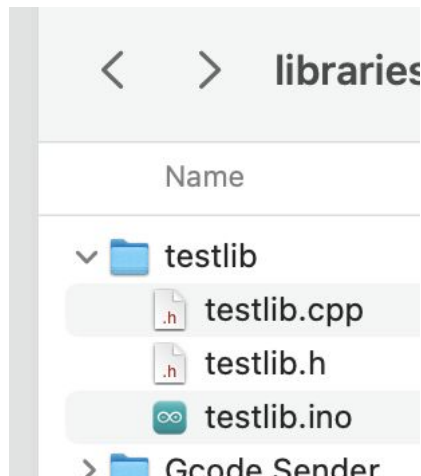
זה כל מה שצריך
בשביל להתחיל!

הסברים על הקוד

קובץ .h הוא הקובץ שבו ה class חושפת את שמה, ואת תכונותיה.

קובץ ה .cpp מגלה כיצד הפונקציות האלה מבוצעות

מקובץ ה .ino מפעילים את הפונקציות מתוך הקוד שלנו



קובץ h.

הגדרת class

מתודות פרטיות (אם יש, של המערכת)

מתודות ציבוריות (שהמשתמש מפעיל)

constructor: זהו המפעיל שמייצר אובייקט

מתודה בשם begin

מתודה בשם action שמקבלת שני int

```
#ifndef TESTLIB_H
```

```
#define TESTLIB_H
```

```
class testlib {
```

```
private:
```

```
public:
```

```
testlib();
```

```
void begin();
```

```
void action(int speedR, int speedL);
```

```
};
```

```
#endif
```

קובץ .cpp

שימו כאן את כל הinclude שלכם

```
#include "testlib.h"
```

```
#include "Arduino.h"
```

```
testlib::testlib() { }
```

```
void testlib::begin() {
```

```
    Serial.begin(115200);
```

```
}
```

```
void testlib::action(int speedR, int speedL) {
```

```
    /* do something */
```

```
}
```

הגדרת פעולות שה**constructor** מבצע

מתודה begin שמפעילה את serial monitor

מתודה action שמקבלת שני משתנים

ועושה "משהו"

קובץ .ino

רק include אחד נחוץ!

הפעלת אובייקט בשם mytest

הפעלת **מתודה begin** (שמפעילה את הserial monitor)

הפעלת **מתודה action** עם 255,255 לדוגמה

```
#include "testlib.h"
```

```
testlib mytest;
```

```
void setup() {
```

```
    mytest.begin();
```

```
}
```

```
void loop() {
```

```
    mytest.action(255,255);
```

```
}
```

שדרוגים

- שה Constructor יקבל ערכים.

- לדוגמה: ערכי פינים בארדואינו

ואז המשתמש יוכל להפעיל את
האובייקט עם מספרי פינים

- הוספת CLI -Command Line Interface



LEVEL
UP!



מוקשים

- אותיות גדולות וקטנות בשמות הקבצים זה class
- הקבצים לא נמצאים באותה ספרייה
- הספרייה לא נמצאת בתיקיית ארדואינו
- שימוש לא נכון בconstructor:
-

אם לא מקבל ערכים: `testlib mytest;`

אם כן מקבל אז צריך לתת אותו: `testlib mytest(xxx);`

דוגמא

<https://github.com/galarb/PenYX>