

## שיעור 1

המייל שלי:

- [jon.muehlst@gmail.com](mailto:jon.muehlst@gmail.com)

- להתחיל כל מייל בקידומת [INTROCPP]

מבנה בסיסי של תוכנה

```
// my first program in C++
#include <iostream>
using namespace std;

int main ()
{
    cout << "Hello World! ";
    cout << "I'm a C++ program";
    return 0;
}
```

מהי תוכנה?

input

Get data from the keyboard, a file, or some other device.

output

Display data on the screen or send data to a file or other device.

math

Perform basic mathematical operations like addition and multiplication.

conditional execution

Check for certain conditions and execute the appropriate sequence of statements.

repetition

Perform some action repeatedly, usually with some variation.

## מהו DEBUGGING

לפעמים, התוכנות שנכתוב לא יצליחו לרוץ בגלל תקלות \ באגים. התהליך שבו אנחנו מחפשים את הבאגים ומתקנים אותם נקרא Debugging

יכולות להיות שלוש סיבות לבאג:

- .1 תחביר \ syntax
  - .1.1 נקודה פסיק
  - .1.2 סוגריים מסולסלים
- .2 באג בזמן ריצה \ runtime
  - .2.1 חריגות זיכרון
  - .2.2 פעולות לא חוקיות
- .3 התוכנה שכתבתם לא עושה את מה שתכנתם \ סמנטיקה

מהם קומנטים \ comments (ובאופן יותר כללי, מה זה תיעוד \ documentation)

אחד הדברים החשובים ביותר בכתיבת תוכנית. התיעוד מאפשר לכם (וחשוב לא פחות... לאחרים) לחזור לקוד אחרי שלא נגעתם בו תקופה ולהבין מה כתוב בו

תדאגו לתת לכל פונקציה ומשתנה שמות שמסבירים את עצמם.

## רווחים \ indentation

שימו לב לשימוש ברווחים בכל בלוק קוד

```
// Determine whether the user input is an even or an odd number
#include <iostream>
using namespace std;

int main () {
```

```

int num;

cout << "Enter a number \n";
cin >> num;

if(num % 2 == 0){
    cout << "The number is even \n";
} else {
    cout << "The number is odd \n";
}
return 0;
}

```

### עקביות

עקביות היא אחד העקרונות החשובים בתכנות. היא מאפשרת מבט יותר ברור על הקוד וגורמת לקוד להיות יותר מסודר. למשל, אם החלטתם להשתמש בשיטת camelCase לכתיבת שמות של משתנים, למשל **variableOne**, אל תכתבו משתנים אחרים ככה: **variable\_two**

### משתנים

ב-CPP אנחנו נדרשים לציין מהו סוג המשתנה שבו נשתמש

הסוגים המובנים:

- **char** - 1 byte
- **int** - x bytes
- **float** - x bytes
- **double** - x bytes
- **bool** - theoretically 1 bit. practically 1 byte
- **void**
- **string**

### ASIDE

מהו bit? מה זה byte?

ה-bit הוא התא הקטן ביותר בזכרון. אפשר לשמור בו 0 או 1.  
ה-byte מאחסן 8 bits.  
כלומר, אם בכל bit אפשר לשמור שני מצבים (1\0) אז ב-byte אפשר לשמור  $8^2$  אפשרויות. כלומר 256.

### הצבת משתנים

לדוגמה

```
int x = 4; // x == 4  
x = 5; // now x == 5
```

### **תזכורת:**

האופרטור = משמש להשמה.  
האופרטור == משמש להשוואה.

### פעולות מתמטיות

+ ●  
- ●  
\* ●  
/ ●  
% ●

### השוואות

< ●  
> ●  
<= ●  
>= ●  
== ●  
!= ●

### התניות

אם התנאי התקיים והרצנו את הקוד בבלוק הנוכחי המחשב ימשיך לרוץ ל-if הבא

```
if( x == 4 ){  
    cout << " x equals 4 \n";  
}
```

אם התנאי התקיים והרצנו את הקוד בבלוק הנוכחי, המחשב ידלג על הבלוק ב-else

```
if( x == 5 ){  
    cout << " x equals 5 \n";  
} else {  
    cout << "something else";  
}
```