קורס תכנות

שיעור ראשון: מבוא

פרטים

- מרצה: אסף זריצקי- assafzar@post.tau.ac.il

- שעת קבלה: יום ד' 10:00 – 9:00
  - בבניין שנקר חדר 405 א'
  - בתיאום מראש בלבד

הכול באתר הוירטואל

- באתר הקורס תוכלו למצוא את:
  - המצגות שהועברו בשיעורים
  - שיעורי הבית
  - פתרונות לשיעורי הבית
  - הודעות משתנות
  - הדרכות שונות
- באחריותכם להתעדכן!

תרגולים

- התרגולים נערכים בכיתת-מחשבים
- מטרתם תרגול מעשי של החומר שנלמד בהרצאות
- מבנה התרגול:
  - הצגה של דוגמא פשוטה בנושא הנלמד
  - תרגול מעשי של דוגמא פשוטה
  - הצגה של פתרונות לבעיות ברמת שיעורי הבית
  - לפעמים פירוט נוסף על חומר הלימוד מעבר למה שנלמד בהרצאה

שיעורי בית

- כל שבוע יפורסם תרגיל עד יום התרגול
- מועד הגשת התרגיל הוא שבוע לאחר הפרסום
- יש חובת הגשה!
- עליכם להגיש לפחות 11/13 תרגילים, חובת הגשה לתרגילים הראשון והאחרון
- ההכנה וההגשה ביחידים
- מותר לחלוק רעיונות ... אסור לחלוק קוד!
- ההעתקה = כישלון בקורס!

שיעורי בית (המשך)

- שאלות לגבי התרגילים אפשר לשאול את המתרגלים
- בדקו את עמוד שאלות ותשובות לפני ששואלים...

http://www.cs.tau.ac.il/~assafzar/cprog11a/q&a.html

- כל תרגיל יכלול שני תרגילים שרק אחד מהם ייבדק
- הגשת שיעורי הבית דרך אתר הוירטואל

הציון בקורס

- 80% מהציון הוא ציון בחינת הגמר.
- 20% מהציון הוא ממוצע תרגילי-הבית.

- ציון המעבר בקורס הוא 60.
- חייבים לקבל ציון עובר בבחינה כדי לעבור את הקורס.
- הבחינה כוללת את כל החומר הנלמד בקורס:
  - כולל את ההרצאות, התרגולים, ושיעורי-הבית.

מטרות הקורס

- היכרות בסיסית עם מבנה המחשב ויכולותיו
- הכרת שפת תכנות: שפת C
- שימוש בתכנות לפתרון בעיות

סילבוס – רשימת נושאי הקורס

1. הכרת המחשב
2. מבנה תוכנית בשפת C
3. היכרות עם סביבת העבודה התכנותית
4. טיפוסי נתונים בסיסים
5. קליטת נתונים והצגתם
6. פעולות אריתמטיות ב C
7. פעולות לוגיות ב C
8. משפטי תנאי (if / else)
9. לולאות (for / while)
10. פונקציות ושימוש בהן
11. רקורסיה
12. מערכים
13. מיונים
14. מחרוזות
15. מצביעים
16. הגדרת טיפוסי נתונים על ידי המתכנת (structures)
17. הקצאת זיכרון דינאמית
18. רשימות מקושרות

ספרים מומלצים

- **C How to program**
  (Deitel and Deitel)

- **A Book on C**
  (Kelley and Pohl)

- **The C Programming Language**
  (Kernighan and Ritchie)

נושאי השיעור היום

- הכרת המחשב
- מבוא לשפות תכנות
- שפת C – יתרונות וחסרונות
- כלי פיתוח תוכנית
- שלבי תרגום התוכנית
- תוכניות לדוגמא

מה המחשב יודע לעשות?

- לנתח נתונים
- לעשות חישובים מתמטיים
- לתכנן מטוסים
- להציג סרטים
- להשמיע מוזיקה
- לגלוש באינטרנט
- לשלוח דואר-אלקטרוני
- לשחק שחמט/ברידג'/שש-בש/...
- להכין סרטי אנימציה
- לבצע עיבוד-תמלילים
- להכין מצגות
- ועוד ועוד....

היכן יש מחשבים?

- כספומט
- טלפון סלולארי
- ממיר טלוויזיה
- DVD
- מענה קולי
- מכונות-חטיפים
- מכוניות
- מטוסים
- לוויינים
- ועוד ועוד...

חלקים עיקריים של מחשב אישי

אמצעי קלט:

מקלדת

עכבר

סורק

אמצעי פלט:

מסך

מדפסת

רמקולים

ה"מוח" של המחשב:

זיכרון

מעבד

אמצעי קלט / פלט:

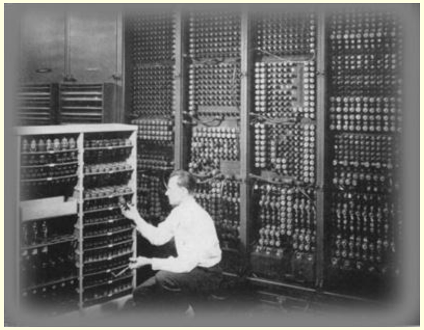
כונן תקליטורים

כונן דיסקטים

דיסק-קשיח

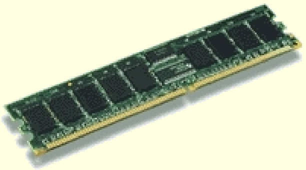
"המוח" של המחשב

- המעבד (CPU – **C**entral **P**rocessing **U**nit):
  - אוסף של רכיבים אלקטרוניים זעירים
  - מסוגל לבצע פעולות בסיסיות על מספרים כגון:
    - חיבור
    - חיסור
    - כפל
    - חילוק
    - שמירת מספר בזיכרון
    - קריאת מספר מהזיכרון
    - השוואת מספרים
- מסוגל לבצע מיליארדי פעולות בשנייה



"המוח" של המחשב - פירוט

- הזיכרון (RAM – Random Access Memory):
  - אוסף של מיליוני או מיליארדי רכיבים אלקטרוניים זעירים
  - כל רכיב מסוגל לשמור מספר (בטווח מסוים)
  - כשמכבים את המחשב תוכן הזיכרון נמחק
    - בשונה ממידע שנשמר על דיסק / דיסקט / תקליטור, שנשאר גם אחרי שמכבים את המחשב
  - תאי הזיכרון ממוספרים (יש להם "כתובת")
  - ניתן לגשת ישירות לכל תא בזיכרון



מה המחשב באמת יודע לעשות ... ומהר

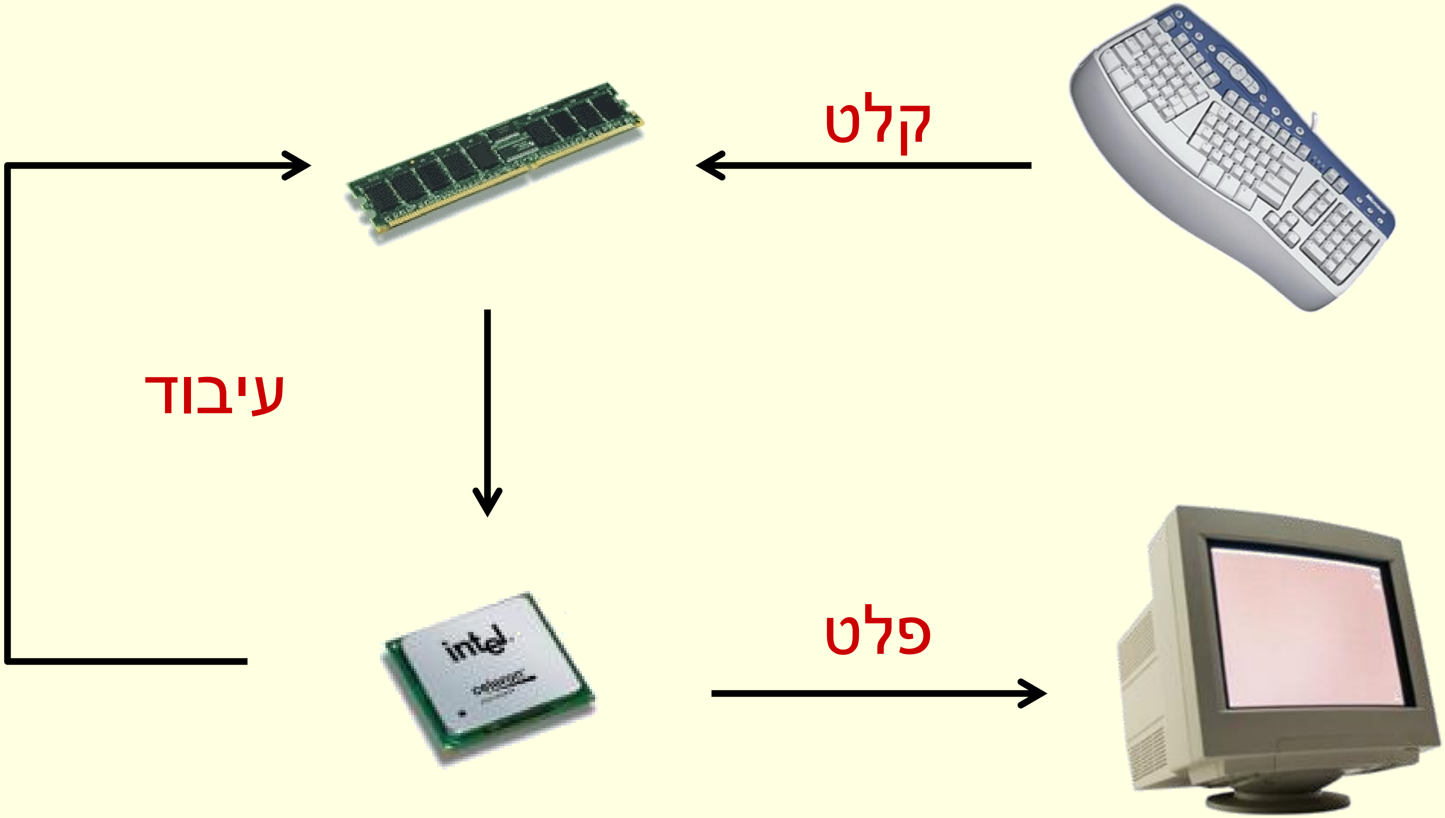
- לעבד מידע והוראות:
  - לבצע פעולות בסיסיות על מספרים (חיבור, חיסור...)
- לקבל מידע והוראות כקלט ולספק מידע כפלט:
  - מידע והוראות מיוצגים במחשב כמספרים
- לבצע פעולות בצורה סדרתית:
  - ביצוע פעולה בתלות בתוצאת פעולה אחרת
  - שמירת תוצאות ביניים לביצוע פעולות נוספות
  - ביצוע פעולות לפי סדר
- לאחסן מידע:
  - מתוך הקלט, או תוצאות עיבוד

אופן הפעולה הבסיסי של מחשב

- המחשב מקבל מידע והוראות כקלט ומאחסן אותם בזיכרון
- המעבד מבצע הוראות שנמצאות בזיכרון יחד עם הנתונים שמאוחסנים בו, ושומר את התוצאות בזיכרון
  - ההוראות מבוצעות לפי הסדר שהתקבלו, אלא אם כן מתקבלת הוראה אחרת
- התוצאות מוחזרות כפלט למשתמש במחשב

- מבוסס על "ארכיטקטורת פון-נוימן" (ארה"ב, 1946)

אופן הפעולה הבסיסי של מחשב



מחשב לעומת בן אדם

- המחשב מייעל את העבודה
  - מהר יותר מאדם
  - לא מתעייף (בלאי נמוך)
  - טועה פחות
  - זול יותר

- בן אדם יודע לחשוב ... מחשב לא
  - חייבים להגיד למחשב בדיוק מה ואיך לעשות
  - אחרת הוא לא ידע לעשות זאת

חומרה ותוכנה

- חומרה - חלקים הפיזיים של המחשב
- תוכנה - סדרת הפעולות הנדרשת לביצוע
  - נקרא גם תכנית מחשב, או תוכנת מחשב
  - סדרת הפעולות מקודדת בשפת מחשב
- מערכת הפעלה - תכנה בסיסית המשמשת כממשק בין המשתמש למחשב
  - מופעלת עם הדלקת המחשב
  - מאוחסנת על הדיסק הקשיח
  - מתווכת בין חלקי חומרה
  - מתווכת בין חומרה לתכנות אחרות

קורס תוכנה

- ליצור תוכנות מחשב
- לתת למחשב הוראות בעזרת שפת C

איך אומרים למחשב מה לעשות?

מבוא לשפות תכנות

איך מתקשרים עם המחשב
כדי ליצור תוכנה חדשה

שפות תכנות: שפת מכונה

- השפה הבסיסית של המחשב:
  - מיוצגת על ידי זרמים חשמליים
  - מכילה פעולות פשוטות (כגון פעולות חשבון בסיסיות)
- הוראות למעבד מיוצגות ע"י רצפים של 0/1 המתארים את עוצמת הזרם בכל חיבור של המעבד
  - לדוגמא: 011101010010
- כל מעבד מכיר אך ורק את שפת המכונה שלו
  - יתכנו שפות מכונה שונות למעבדים שונים

שפות תכנות: שפת מכונה

- במחשבים הראשונים כל פעולה בסיסית הצריכה הזזת מתגים חשמליים רבים כדי להעביר זרמים מתאימים
  - כמו שרואים בסרטים ישנים

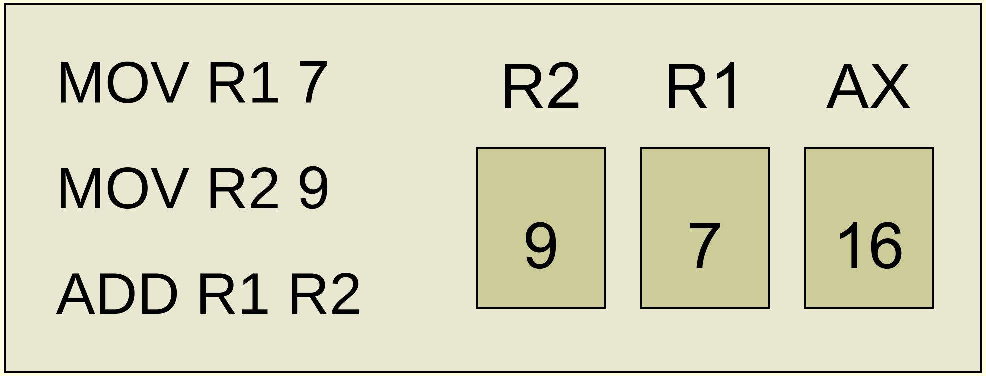
- כיום המחשב יודע לקרוא את הרצפים האלה (ההוראות) מדיסקטים, תקליטורים, וכו'

- בן אדם (נורמלי) לא מסוגל לכתוב תוכנית בשפת מכונה

```
1011100010001110011101100111100
0100110000010010000111010101010010
11111100110001100011000111100001
```

שפות תכנות: שפת אסמבלי

- שפה בסיסית עם פקודות פשוטות באנגלית
  - MOV,ADD,PUSH
  - לא צריך לכתוב יותר רצפים של 0/1
- כל פקודה מתורגמות לפקודה בשפת-מכונה על-ידי תוכנה בשם אסמבלר



- גם בשפת אסמבלי מאוד קשה לתכנת

שפות תכנות: שפות עיליות

- שפות עיליות לדוגמא: פסקל, בייסיק, C, C++, Java
  - יותר מזכירות אנגלית
  - משתמשות גם בסימנים מתמטיים

- כל פקודה בשפה עילית מתורגמת לסדרת פקודות בשפת מכונה (יותר מפקודה אחת).
- הקומפיילר (מהדר) מתרגמן לשפת מכונה
- לכל שפה יתרונות, חסרונות והתאמה ליישומיים שונים

מאפייני שפת C - השפה העילית שנלמד

יתרונות:

- שפה פרוצדוראלית:
  - ניתן להגדיר הוראות חדשות (פונקציות)
- מודולארית:
  - מאפשרת שימוש חוזר בחלקי תוכניות
- תאימות למחשבים שונים
- שפה יעילה (מהירות ביצוע)
- נותנת גישה ל"קרביים" של המחשב

חסרונות:

- נותנת גישה ל"קרביים" של המחשב
- תחביר קצת מבלבל

כלים שישמשו אותנו לתכנות בשפת C

- Editor - תוכנה לכתיבת ועריכת התוכנית בשפת C
  - התכנית נקראת: "source code"
- Compiler (מהדר) - מתרגם את התוכנית לשפת מכונה.
  - לקוד המתורגם נקרא: "object code"
- Linker - מחבר מספר קבצי object (אחד או יותר) ויוצר מהם קובץ הרצה יחיד (בשפת מכונה)
  - קובץ ההרצה נקרא: "executable"
- Debugger - מאפשר הרצה מבוקרת של התוכנית לצורך בדיקה ותיקון

שלבי יצירת התוכנית

כתיבת התוכנית - Editor ← תוכנית C

שלבי יצירת התוכנית

Editor - כתיבת התוכנית

תוכנית C

Compiler

- מתרגם פקודות ב-C ל- object code (שפת-מכונה)

שלבי יצירת התוכנית

Editor - כתיבת התוכנית

תוכנית C

Compiler

Linker

- מתרגם פקודות ב-C ל- object code (שפת-מכונה)
- מחבר מספר קבצי object לקובץ הרצה

שלבי יצירת התוכנית

Editor - כתיבת התוכנית

תוכנית C

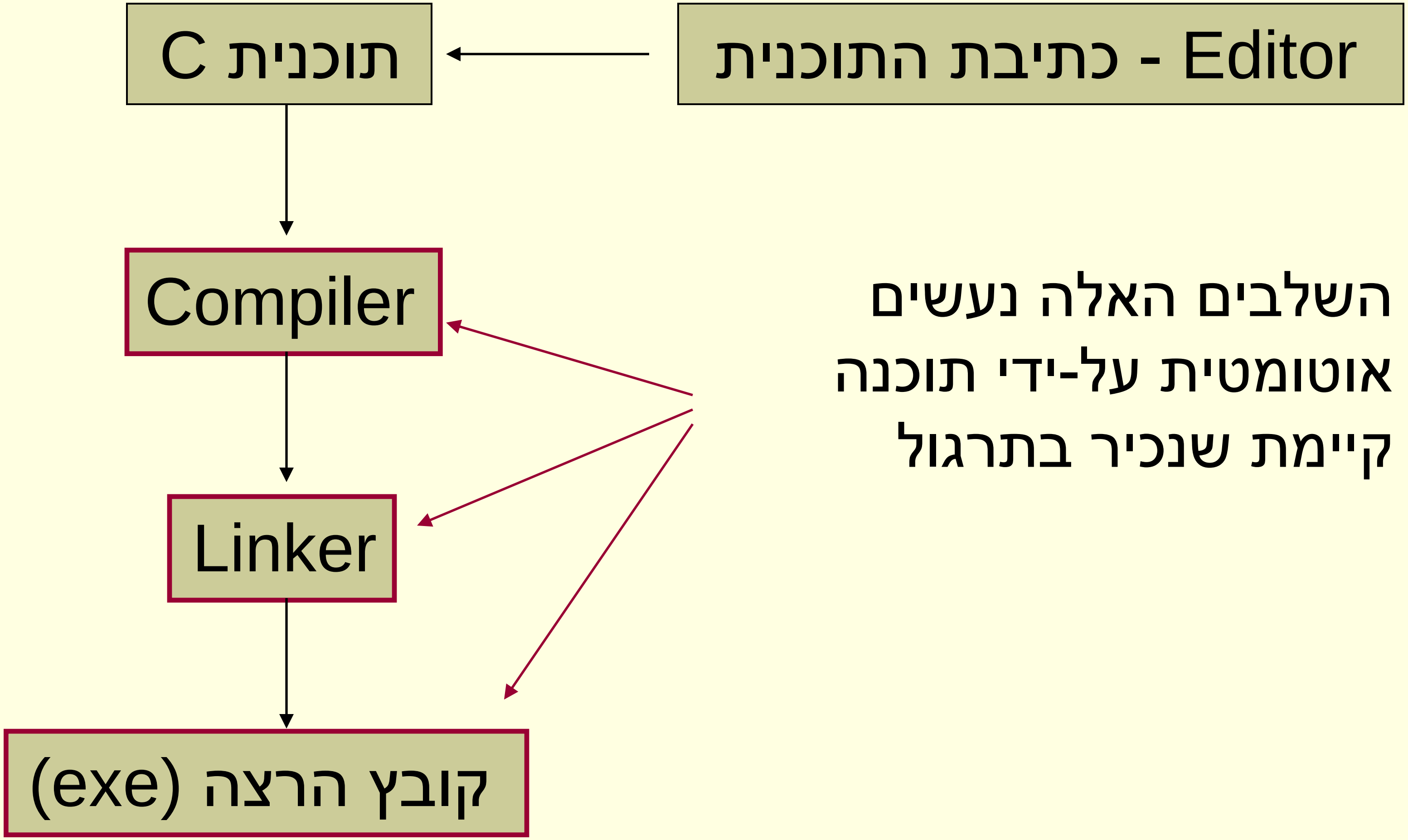
Compiler

Linker

קובץ הרצה (exe)

- מתרגם פקודות ב-C ל- object code (שפת-מכונה)
- מחבר מספר קבצי object לקובץ הרצה

שלבי יצירת התוכנית



- השלבים האלה נעשים אוטומטית על-ידי תוכנה קיימת שנכיר בתרגול

שלבי יצירת התוכנית

Editor - כתיבת התוכנית

תוכנית C

Compiler

Linker

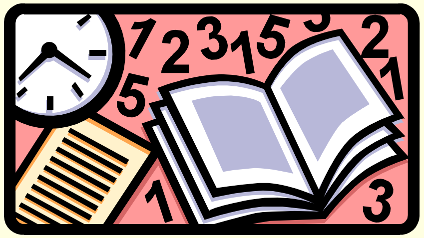
קובץ הרצה (exe)

- בקורס נלמד כיצד לכתוב תוכניות שיפתרו עבורנו בעיות שונות

סיכום ביניים

- חלקי המחשב, חומרה ותוכנה
- שפת מכונה ושפות עיליות
- תרגום תוכנית משפה עילית

איך נראית תכנית מחשב בשפת C



תוכנית ראשונה בשפת "C

- הצג על המסך:
  - “Hello World”

```
#include <stdio.h>

int main()
{
   printf("Hello World\n");
   return 0;
}
```

תוכנית ראשונה - int main()

```c
#include <stdio.h>

int main()
{
   printf("Hello World\n");
   return 0;
}
```

- שורה קבועה שבה מתחיל ביצוע תוכנית בשפת C
- המילה main() מציינת שזוהי הפונקציה הראשית בתכנית
- הסוגריים המסולסלים { } מגדירים מסגרת לאוסף הפקודות שיש לבצע

תוכנית ראשונה - int main()

```
#include <stdio.h>

int main()
{
   printf("Hello World\n");
   return 0;
}
```

- int - קיצור ל integer
- מייצגת את טיפוס הערך המוחזר מהפונקציה
- במקרה של main הערך מוחזר למערכת ההפעלה בתום ריצת התוכנית
- בשיעור הבא נלמד על טיפוסים שונים של ערכים

תוכנית ראשונה - return

```
#include <stdio.h>

int main()
{
   printf("Hello World\n");
   return 0;
}
```

- פקודת ה- return מחזירה ערך מהפונקציה ובכך מסיימת את פעולת הפונקציה
  - הערך חוזר אל מפעיל הפונקציה
  - במקרה שלנו, הערך 0 מוחזר למערכת ההפעלה

החזרת הערך 0 מסמלת סיום תקין של ההפעלה

- ; - סימן שמציין עבור המהדר סוף של פקודה

printf – תוכנית ראשונה

```
#include <stdio.h>

int main()
{
   printf("Hello World\n");
   return 0;
}
```

כמו כל פקודה, גם זו מסתיימת ב- ;

- printf - פונקציה שהוגדרה בשפת C
- רצף התווים בתוך הסוגריים מורה לפונקציה מה לעשות
- printf מציגה אל המסך רצף תווים שנמצא בין זוג סימני גרשיים
- \n - סימן מיוחד שמשמעותו שורה חדשה

תוכנית ראשונה – #include

```
#include <stdio.h>

int main()
{
   printf("Hello World\n");
   return 0;
}
```

- הפונקציה printf אינה מיושמת על ידנו ולכן יש לומר למהדר היכן היא נמצאת.
- ‎#include - מאפשר שימוש בקטעי קוד שנכתבו ע"י אחרים ונשמרו בקובץ ששמו נמצא בין הסוגרים < >.
- stdio.h הוא קובץ "ספריה" של פקודות המטפלות בקלט / פלט (standard input/output)

תוכנית ראשונה – #include

```
#include <stdio.h>

int main()
{
   printf("Hello World\n");
   return 0;
}
```

- משמעות השורה:
  - הוראה למהדר לאפשר שימוש בפקודות המפורטות בקובץ stdio.h
- המהדר יעתיק את תוכן הקובץ stdio.h לתכנית שלנו
- הסימן # מציין עיבוד מוקדם על ידי המהדר לפני התרגום לשפת מכונה
  - מבוצע ע"י רכיב במהדר שנקרא preprocessor

תוכנית ראשונה – סיכום

```
#include <stdio.h>

int main()
{
   printf("Hello World\n");
   return 0;
}
```

- מסגרת לפקודות
- שימוש ב printf להדפסת ההודעה הרצויה
- שימוש ב- stdio.h לצורך שימוש ב-printf
- החזרת ערך לסיום תקין
- סיום פקודה ב - ;

תוכנית שנייה: חישוב מספר הדקות ביממה

- נתון:
  - ביממה יש 24 שעות
  - בשעה יש 60 דקות
- חשב כמה דקות יש ביממה
- הצג את התוצאה למסך

```c
#include <stdio.h>

int main()
{
   int hours, minutes, total;
   hours = 24;
   minutes = 60;
   total = hours * minutes;
   printf("Minutes in a day: %d\n", total);
   return 0;
}
```

תוכנית שנייה: חישוב מספר הדקות ביממה

```
#include <stdio.h>

int main()
{
   int hours, minutes, total;
   hours = 24;
   minutes = 60;
   total = hours * minutes;
   printf("Minutes in a day: %d\n", total);
   return 0;
}
```

- אותן השורות כמו בתכנית הראשונה

תוכנית שנייה: חישוב מספר הדקות ביממה

- הגדרת משתנים

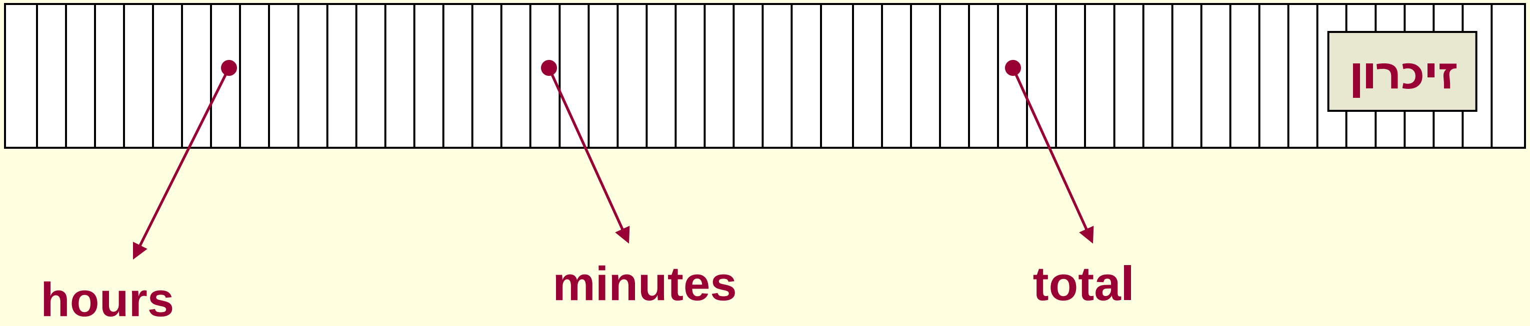
```c
#include <stdio.h>

int main()
{
   int hours, minutes, total;
   hours = 24;
   minutes = 60;
   total = hours * minutes;
   printf("Minutes in a day: %d\n", total);
   return 0;
}
```

הגדרת משתנים

- משתנה:
  - מקום מסוים בזיכרון
  - יכול להכיל ערך מסוג קבוע מראש
  - כל משתנה מקבל שם
  - לפני השימוש במשתנים צריך להגדיר אותם

הגדרת משתנים

- השורה `int hours, minutes, total;` מגדירה:
  - שלושה משתנים (מקומות בזיכרון)
  - לכל משתנה נותנים שם
  - התוכנית יכולה לשמור ערכים במשתנים
  - התוכנית יכולה לקרוא ערכים מהמשתנים
  - ההגדרה קובעת שערכי המשתנים יהיו מספרים שלמים (מטיפוס int)



תוכנית שנייה: חישוב מספר הדקות ביממה

```
#include <stdio.h>

int main()
{
   int hours, minutes, total;
   hours = 24;
   minutes = 60;
   total = hours * minutes;
   printf("Minutes in a day: %d\n", total);
   return 0;
}
```

- השמה:
  - הצבת ערכים במשתנים.

השמה

```
hours = 24;
minutes = 60;
```

- בזיכרון של המשתנה hours נשמור את הערך 24
- בזיכרון של המשתנה minutes נשמור את הערך 60
- הסימן = מיצג השמה

total	minutes	hours	
	60	24	זיכרון

תוכנית שנייה: חישוב מספר הדקות ביממה

- ביצוע פעולה חשבונית בנוסף להשמה

```c
#include <stdio.h>

int main()
{
   int hours, minutes, total;
   hours = 24;
   minutes = 60;
   total = hours * minutes;
   printf("Minutes in a day: %d\n", total);
   return 0;
}
```

total	minutes	hours	
1440	60	24	זיכרון

ביצוע פעולה בנוסף להשמה

```
total = hours * minutes;
```

- זו פעולת השמה
- הסימן * מייצג כפל
- פעולת הכפל מתבצעת בין הערכים ששמורים במשתנים
- סדר הפעולות:
  - חישוב הערך שמתקבל מהפעולה שמימין להשמה
    - סדר קדימויות מתמטי רגיל
  - שמירת התוצאה במשתנה שרשום בצד שמאל
- הסימנים /,-,+,* מייצגים חילוק, חיסור, חיבור וכפל

תוכנית שנייה: חישוב מספר הדקות ביממה

- הדפסת התוצאה למסך.

```
#include <stdio.h>

int main()
{
   int hours, minutes, total;
   hours = 24;
   minutes = 60;
   total = hours * minutes;
   printf("Minutes in a day: %d\n", total);
   return 0;
}
```

הדפסה למסך

```
Printf("Minutes in a day: %d\n", total);
```

- בעזרת printf ניתן להדפיס למסך ערך של משתנים
  - יש לציין את סוג המשתנה בעזרת סימן מיוחד
  - %d מציין משתנה המכיל מספר שלם
  - לאחר הגרשיים יש לציין את שם המשתנה שערכו יודפס

תוכנית שנייה: חישוב מספר הדקות ביממה

```
#include <stdio.h>

int main()
{
   int hours, minutes, total;
   hours = 24;
   minutes = 60;
   total = hours * minutes;
   printf("Minutes in a day: %d\n", total);
   return 0;
}
```

- הגדרת משתנים
- השמה
- פעולה חשבונית
- הדפסת ערך משתנה

סיכום

- מבוא לתכנות ולשפת C
- מבנה בסיסי של תוכנית C
- הדגמה של שימוש במשתנים

בשיעורים הקרובים נבין טוב יותר מה עומד מאחורי הדוגמאות שראינו, ונכיר עוד דברים שניתן לעשות בשפת C

להתראות ☺