

שפת C

מותאם להנדסאי אלקטרוניקה – כיתה י"ג
מספר שאלון: 711911

1

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

1

פרק 7: מבני נתונים (union ו-structure)

יעדים

היכרות עם הפשטת מידע מתקדמת בעזרת מבנים (struct) ואיחוד מבנים (union)

תכנים

1. הרעיון והיסודות של מבנים
2. הגדרה ושימוש של מבנים
3. פונקציות ומבנים – העברת פרמטרים, ערך מוחזר
4. מערכים של מבנים ומצביעים למבנים
5. יצירת טיפוסים בעזרת ההוראה typedef
6. הקצאות דינמיות של מבנים ומבנים המצביעים לעצמם self-referential structures
7. מימוש רשימות מקושרות, עצי החלטה בינריים
8. שדות-ביטים bit-fields, הגדרה ושימוש

סיכום שעות ההוראה: 18 שעות עיוניות ו- 10 שעות התנסויות. סה"כ 28 שעות.

2

cohen.bareket@gmail.com

16 אוגוסט 25

2

פרק 7- מבני נתונים:

1. הרעיון והיסודות של מבנים

3

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

3

מבנים ו Union לומדים לארגן מידע ב-C

היום נכיר קופסה חכמה ששומרת כמה סוגי מידע יחד
struct

נכיר גם קופסה מיוחדת ששומרת בכל רגע רק סוג
אחד union

4

מה יותר נוח?

- תדמיינו שיש לכם הרבה נתונים על אותו דבר – למשל תלמיד : שם, גיל, ציון.
- אפשר לשמור כל נתון במשתנה נפרד (– name, age, grade) אבל אז זה מפוזר.
- מבנה הוא כמו קופסה או קלסר שמכילה את כל הנתונים האלה יחד במקום אחד.
- במבנה אנחנו שומרים כמה סוגי נתונים שונים – אבל כולם שייכים לאותו "דבר". במקום שהנתונים יהיו מפוזרים במקומות שונים, הם נמצאים יחד בתוך "חבילה אחת".
- **הגדרה:** מבנה הוא אוסף של נתונים, שיכולים להיות מאותו סוג או מסוגים שונים, שמוגדרים כקבוצה אחת ונשמרים אחד אחרי השני בזיכרון.

5

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

5

דוגמה 1

- **קלסר עם כיסים** – דמינו שיש לכם קלסר, ובתוכו כמה כיסים :
 - כיס עבור השם
 - כיס לכתובת
 - כיס לגיל
- הקלסר עצמו מייצג מבנה, וכל כיס מייצג שדה (field) בתוך המבנה.
- **כל הכיסים שייכים לאותו אדם.**

6

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

6

דוגמה 2

כרטיס בספריה – זה כרטיס אחד (המבנה), שיש עליו:

- שם הספר
- שם המחבר
- שנת ההוצאה

כל המידע הזה שייך לאותו ספר ונמצא על אותו כרטיס – בדיוק כמו שבו struct כל השדות שייכים לאותה ישות.

**בשתי הדוגמאות – אנחנו רואים אובייקט אחד
(קלסר / כרטיס) שמכיל כמה נתונים שונים
שקשורים לאותו נושא.**

פרק 7- מבני נתונים:

2. הגדרה ושימוש במבנים struct

9

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

9

שלב 1- מבנה struct

```
struct TypeName {
    DataType1 fieldName1;
    DataType2 fieldName2;
    ...
};
```

חשוב לזכור:

• ההגדרה לא יוצרת עדיין משתנה – היא רק מגדירה תבנית.
• בכל שדה אפשר לשים טיפוס נתונים שונה

10

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

10

דוגמה בסיסית

```
struct Student {
    char name[20]; // שם התלמיד
    int age; // גיל
    float grade; // ציון
};
```

יצרנו סוג חדש בשם Student שמכיל שלושה שדות: שם, גיל, ציון.

למה זה טוב?

- סדר – כל הנתונים של אותו אובייקט מרוכזים במקום אחד.
- קלות שימוש – אפשר לשלוח את כל המידע לפונקציה בבת אחת.
- קריאות – קל להבין את הקוד.

11

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

11

שלב 2 - יצירת משתנה מסוג המבנה

- אפשר ליצור משתנה בזמן ההגדרה של המבנה.
- אפשר ליצור משתנה ממבנה שהוגדר מראש.

12

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

12

שלב 2 - יצירת משתנה מסוג המבנה

- אפשר ליצור משתנה בזמן ההגדרה:

```
struct {
    long id;
    char name[20];
    int weight;
} p1, p2;
```

אין שם למבנה, אלא מגדירים את השדות ויוצרים את המשתנים מיד בסוף ההגדרה.

13

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

13

שלב 2 - יצירת משתנה מסוג המבנה

- אפשר ליצור משתנה ממבנה שהוגדר מראש:

```
struct Student s1;
```

אתחול הנתונים:

```
strcpy(s1.name, "David"); // למחרוזות חייבים
s1.age = 17;
s1.grade = 95.5;
```

- דרך נוספת היא לאתחל את הנתונים בזמן יצירת המשתנה:

```
struct Student s2= {" David ",17, 95.5};
```

14

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

14

שלב 3 – שימוש בשדות

ניגשים לשדה בעזרת נקודה (.)

```
printf("Name: %s\n", s2.name);
printf("Age: %d\n", s2.age);
printf("Grade: %.1f\n", s2.grade);
```

15

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

15

```
#include <stdio.h>
#include <string.h>
struct Student {
    char name[20];
    int age;
    float grade;
};

int main() {
    struct Student s2;
    strcpy(s2.name, "David");
    s2.age = 18;
    s2.grade = 92.5;
    printf("Name: %s\n", s2.name);
    printf("Age: %d\n", s2.age);
    printf("Grade: %.1f\n", s2.grade);
    return 0;
}
```

התוכנית בשלמותה:

```
Name: David
Age: 18
Grade: 92.5
```

16

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

16

תרגיל 1 – יצירת משתנה ושינוי ערכים

1. צרו משתנה מסוג struct Student בשם studentA.
2. מלאו אותו בנתונים: שם = "נועה", גיל = 17, ציון = 92.5.
3. הדפיסו למסך את שלושת הנתונים.

17

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

17

תרגיל 2 – אתחול בעת יצירה

• משימה:

צרו משתנה מסוג struct Student בשם studentB עם ערכים: "אורי", 18, 88.0, בשורת קוד אחת.

18

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

18

חשוב לשים לב!

חייבת להיות התאמה מלאה בין הערכים שמאתחלים לבין הסדר והטיפוסים של השדות שהוגדרו ב-struct

למשל:

```
struct Student {
    char name[20]; // שם התלמיד
    int age; // גיל
    float grade; // ציון
};

struct Student student = {"אורי", 18, 88};
```

16 אוגוסט 25

19

דיון כיתתי

לפי דעתכם מה עדיף לשמור לכל תלמיד את הנתונים האישיים שלו: השם, הגיל והציון במשתנים נפרדים? או במבנה?

20

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

20

```
#include <stdio.h>
```

```
struct Student {
```

```
    char name[20]; // שם בלי רווחים
```

```
    int age;
```

```
    float grade;
```

```
};
```

21

אתחול נתונים על ידי קליטה מהמשתמש

```
int main() {
```

```
    struct Student s;
```

```
    printf("Enter name: ");
```

```
    scanf("%19s", s.name);
```

```
    printf("Enter age: ");
```

```
    scanf("%d", &s.age);
```

```
    printf("Enter grade: ");
```

```
    scanf("%f", &s.grade);
```

```
    printf("\nStudent details:\n");
```

```
    printf("Name: %s\n", s.name);
```

```
    printf("Age: %d\n", s.age);
```

```
    printf("Grade: %.1f\n", s.grade);
```

```
    return 0;
```

```
}
```

16 אוג

21

דף תרגול STRUCTURE

22

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

22

structure סיכום

- struct מאפשר לאגד מספר שדות (משתנים) לסוג חדש אחד.
- לכל שדה יש מקום משלו בזיכרון – כל הערכים קיימים בזמנית.
- ניתן לגשת לכל שדה בנפרד, לשנות ולקרוא אותם בלי לפגוע בשדות האחרים.

```
struct Student {
    int id;
    char name[20];
    float grade;
};
```

לכל אחד מהשדות יש
מקום משלו בזיכרון
כל שדה מקבל כתובת
זיכרון נפרדת ומוקצה לו
שטח משלו במחשב

23

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

23

מעבר מ-struct ל-union

- לעיתים, **לא צריך** שכל השדות יהיו פעילים באותו זמן.
- לדוגמה: נרצה לשמור או מס' שלם או טקסט, אבל לא את שניהם יחד.
- במצב כזה – חבל לבזבז זיכרון על כל השדות.
- כאן נכנס לתמונה ה-union :
- כל השדות נמצאים **באותו מקום בזיכרון**.
- בכל רגע אפשר להשתמש רק באחד מהם.

24

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

24

מה זה union ?

- union הוא מבנה נתונים שמאפשר להגדיר מספר שדות שחולקים את אותה כתובת בזיכרון.
- בכל רגע נתון, רק שדה אחד מכיל ערך תקין.
- גודל ה-union נקבע לפי השדה הגדול ביותר.
- שימושי לחיסכון בזיכרון, במיוחד במערכות משובצות (Embedded).

הידעת?

מערכות משובצות (Embedded Systems) - הן מחשבים קטנים המוטמעים בתוך מוצרים שונים כדי לבצע משימות ספציפיות. כמעט כל מכשיר אלקטרוני מודרני מכיל מערכת משובצת שעושה אותו "חכם" ומאפשרת לו לתפקד אוטומטית.

25

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

25

הגדרת union בסיסית

```
union Data {
    int i;
    float f;
    char str[20];
};
```

i ו- f משתמשים באותו מקום בזיכרון.
אם נשנה את i, זה ישפיע על הערך של f
ו- str כי הם חופפים בזיכרון.

```
union Data data;
```

26

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

26

```
#include <stdio.h> void main ( )
#include <string.h> {
```

דוג' לשימוש

```
union Data{
    int i;
    char str;[20]
};
```

```
union Data d;
d.i = 1234;
printf("i = %d\n", d.i);

strcpy(d.str, "Hello");
printf("str = %s\n", d.str);

printf("i = %d\n", d.i);

}
```

i = 1234
str = Hello
i = 181904314

27

16 אוגוסט 25

27

יתרונות וחסרונות

חסרונות

- לא ניתן לשמור נתונים תקינים בכל השדות בו-זמנית.
- דורש זהירות – שינוי שדה אחד מוחק את המידע בשדות האחרים.

יתרונות

- חוסך מקום בזיכרון.
- מאפשר לייצג נתונים בכמה צורות באותו מקום.

28

מצגת זו נוצרה על ידי אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

28

סיכום

- struct כל השדות בזיכרון בנפרד, אפשר לשמור ערכים בכולם בו־זמנית
- union כל השדות באותו מקום בזיכרון, רק אחד תקין בכל זמן.
- שימוש ב־ union כשצריך לחסוך זיכרון או לייצג נתונים בכמה צורות שונות.

29

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

29

דף תרגול UNION

30

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

30

פרק 7- מבני נתונים:

3. פונקציות ומבנים: העברת פרמטרים, ערך מוחזר.

31

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

31

איך שולחים מבנה לפונקציה ואיך מחזירים מבנה מפונקציה?

עד כה למדנו:

- איך עובדים עם מצביעים למבנה
- תרגלנו גישה לשדות עם שימוש בנקודה
- פונקציות עם פרמטרים וערך מוחזר.

שמחברים בין פונקציות למבנים, נוצר:

- פעולות מסודרות שעובדות על "קופסת מידע" שלמה.
- חוסך חזרה על קוד.
- עושה את הקוד ברור יותר ("מה הפעולה עושה" במקום "איך בדיוק עושים כל שלב").
- מקל על תיקונים: משנים במקום אחד—וזה עובד בכל המקומות.

32

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

32

דוג בלי שימוש בפונקציה

#include <stdio.h>

```
struct Student {
    char name[20];
    int age;
    float grade;
};
```

```
int main() {
    struct Student s1 = {"Noa", 16, 92.5};
    struct Student s2 = {"Dan", 17, 85.0};

    // הדפסת תלמיד 1
    printf("%s (%d) - grade: %.1f\n", s1.name, s1.age, s1.grade);

    // הדפסת תלמיד 2 (אותו קוד שוב)
    printf("%s (%d) - grade: %.1f\n", s2.name, s2.age, s2.grade);

    return 0;
}
```

33

16 אוגוסט 25

33

#include <stdio.h>

```
struct Student {
    char name[20];
    int age;
    float grade;
};
```

דוג בלי שימוש בפונקציה

מה יקרה אם נרצה להדפיס את תלמידים?

```
// הדפסת תלמיד 1
printf("%s (%d) - grade: %.1f\n", s1.name, s1.age, s1.grade);

// הדפסת תלמיד 2 (אותו קוד שוב)
printf("%s (%d) - grade: %.1f\n", s2.name, s2.age, s2.grade);

return 0;
}
```

34

16 אוגוסט 25

34

```
#include <stdio.h>
```

```
struct Student {
    char name[20];
    int age;
    float grade;
};
```

נשנה את התוכנית לשימוש בפונקציה:

```
void printStudent(struct Student s) {
    printf("%s (%d) - grade: %.1f\n", s.name, s.age, s.grade);
}

int main() {
    struct Student s1 = {"Noa", 16, 92.5};
    struct Student s2 = {"Dan", 17, 85.0};

    printStudent(s1);
    printStudent(s2);
    return 0;
}
```

35

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

35

תרגיל

1. גדירו מבנה בשם Student עם השדות:

- char name[20]
- int age
- float grade (טווח 0..100)

2. כתבו פונקציה בשם isExcellent שמקבלת תלמיד (struct Student) ומחזירה:

- 1 אם הציון שלו גדול או שווה ל-90
- 0 אחרת

3. בתוכנית הראשית (main):

- יצרו תלמיד בשם "Roni", גיל 17, וציון 91.0
- יקראו לפונקציה ובדקו את התוצאה
- אם התלמיד מצטיין – הדפיסו: Roni is excellent!
- אחרת – הדפיסו: Roni is not excellent.

36

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

36

תרגיל

1.גדירו מבנה בשם Student עם השדות:

•char name[20]

•int age

•float grade(טווח 100..0)

2.כתבו פונקציה בשם isExcellent שמקבלת תלמיד (struct Student) ומחזירה:

•1 אם הציון שלו גדול או שווה ל-90

•0 אחרת

3.בתוכנית הראשית (main):

•יצרו תלמיד בשם "Roni", גיל 17, וציון 91.0

•קראו לפונקציה ובדקו את התוצאה

•אם התלמיד מצטיין – הדפסו: Roni is excellent!

•אחרת – הדפסו: Roni is not excellent.

הרחבה (רשות): נסו לשנות את הציון ל-85.0 ולהריץ שוב. מה יקרה בפלט?

37

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

37

תרגיל כיתה

הגדירו מבנה Car עם:

model[20], year, price.

כתבו פונקציה printCar(struct Car c) שמדפיסה את ערכי המבנה

בצורה הבאה:

Model: ____ | Year: ____ | Price: ____

אתחלו שני רכבים עם ערכים כרצונכם והפעילו עליהם את פונקצית ההדפסה

38

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

38

פרק 7- מבני נתונים: 4. מצביעים למבנים

39

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

39

מה יקרה אם נשנה ערך שדה של מבנה בתוך פונקציה?

כתבו פונקציה `void makePerfect(struct Student s)`
הפונקציה משנה את `s.grade` ל-100.

הפעילו את הפונקציה והדפיסו את הציון אחרי הקריאה.

מה קורה? למה?

40

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

40

מה יקרה אם נשנה ערך שדה של מבנה בתוך פונקציה?

כתבו פונקציה

```
void makePerfect(struct Student s)
```

הפונקציה משנה את s.grade ל-100.

הפעילו את הפונקציה והדפיסו את הציון אחרי הקריאה.

מה קורה? למה?

```
void makePerfect(struct Student s)
{
    s.grade = 100.0f;
}

int main(){
    struct Student st = {"Dana", 16, 70.0};
    makePerfect(st);
    printf("%.1f\n", st.grade);
}
```

41

מצגת ז
cohen.bareket@gmail.com

16 אוגוסט 25

41

העברת מבנה לפונקציה לפי מצביע - Pass by Pointer

בחלק הקודם ראינו שכאשר אנחנו מעבירים מבנה לפונקציה **לפי ערך (by value)** נוצר עותק של המבנה, וכל שינוי שנעשה בתוך הפונקציה **לא ישפיע** על המשתנה המקורי בזיכרון.

אבל לפעמים, אנחנו דווקא רוצים לשנות את הערכים של השדות במבנה המקורי.

כדי לעשות זאת – נעביר לפונקציה **כתובת (מצביע) של המבנה**, וכך נוכל לגשת ישירות למבנה בזיכרון ולשנות אותו.

42

מצגת זו נוצרה על ידי אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

42

הסבר הרעיון

- **במקום** לשלוח לפונקציה את המבנה עצמו, שולחים לה **מצביע** למבנה (`struct Student*` לדוגמה).
- בתוך הפונקציה, ניגשים לשדות של המבנה דרך **אופרטור החץ** (`->`) במקום הנקודה (`.`).
- **כל שינוי שנבצע על שדות המבנה – ישפיע על המשתנה המקורי, כי אנחנו עובדים על אותה כתובת בזיכרון.**

43

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

43

```
#include <stdio.h>
#include <string.h>
```

```
struct Student {
    char name[20];
    int age;
    float grade; // 0..100
};
```

```
void updateStudent(struct Student *s)
{
    s->age = 18;
    s->grade = 95.5;
}
```

```
void printStudent(struct Student s) {
    printf("שם: %s\n", s.name);
    printf("גיל: %d\n", s.age);
    printf("ציון: %f\n", s.grade);
}
```

44

```
int main()
{
```

```
    struct Student s1 = {"Dana", 16, 82.0};
```

```
    printf("\n לפני העדכון:");
    printStudent(s1);
```

```
    updateStudent(&s1);
```

```
    printf("\n אחרי העדכון:");
    printStudent(s1);
```

```
    return 0;
```

```
}
```

תרגיל דוגמה

תרגיל כיתה
מה יהיה הפלט????

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

44

דף תרגול

45

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

45

פונקציה שמחזירה מבנה (Return)

- כדי לא לגעת במקור
- ◆ כדי להחזיר כמה ערכים ביחד

46

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

46

פונקציה שמחזירה מבנה (Return)

◆ כדי לא לגעת במקור

לפעמים אנחנו לא רוצים לשנות את הנתונים שקיבלנו לפונקציה, אלא ליצור מהם עותק חדש.
לדוגמה: אם יש לנו תלמיד עם ציון 70 ואנחנו רוצים לבדוק מה היה קורה אם היינו מוסיפים לו 5 נקודות – בלי לשנות את הנתונים האמיתיים.

◆ כדי להחזיר כמה ערכים ביחד

פונקציה רגילה מחזירה ערך אחד (מספר, מחרוזת וכו').
 אבל אם אנחנו צריכים להחזיר כמה נתונים ביחד – מבנה מאפשר "לאחד אותם בחבילה אחת" ולהחזיר את הכל בבת אחת.

47

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
 cohen.bareket@gmail.com

16 אוגוסט 25

47

תרגיל חימום – חבילת פירות 🍇 🍌 🍏

• אנחנו רוצים פונקציה שתחזיר "חבילה" עם 3 סוגי פירות וכמות מכל אחד. הגדירו מבנה בשם **FruitBox** עם שלושה שדות:

- **apples** - מספר תפוחים
- **bananas** - מספר בננות
- **grapes** - מספר ענבים

• כתבו פונקציה **makeBox(a, b, g)** המקבלת שלושה מספרים ומחזירה מבנה מלא.

• ב־ main צרו חבילה עם 5 תפוחים, 3 בננות ו־20 ענבים והדפיסו.

48

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
 cohen.bareket@gmail.com

16 אוגוסט 25

48



תרגיל מס 2 – קופסת צבעים

- הגדירו מבנה בשם **ColorBox** עם שלושה שדות:
 - **red** מספר עפרונות אדומים
 - **blue** מספר עפרונות כחולים
 - **green** מספר עפרונות ירוקים
- כתבו פונקציה **makeColors(r, b, g)** שמחזירה מבנה מלא.
- ב־ **main** צרו קופסה עם 10 אדומים, 7 כחולים ו־5 ירוקים והדפיסו.

49

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

49

תרגיל מס 3 – ספר

- הגדירו מבנה בשם **Book** עם שלושה שדות:
 - **pages** מספר עמודים
 - **year** שנת הוצאה לאור
 - **price** מחיר
- כתבו פונקציה **makeBook(p, y, pr)** שמחזירה מבנה מלא.
- ב־ **main** צרו ספר עם 250 עמודים, שנת 2020, ומחיר 99.90 ₪ והדפיסו.

50

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

50

פונקציה שמקבלת מצביע למבנה ומחזירה תוצאה

לפעמים אנחנו צריכים:

- גם **לעדכן** את המבנה המקורי (למשל, להעלות ציון/מחיר).
- וגם לדעת אם העדכון **הצליח** (תנאי תקינות, טווחים וכו').

לכן נשתמש בפונקציה ש:

- מקבלת **כתובת** למבנה → משנה את השדות באמת.
- **מחזירה** int/char (למשל 1/0) כדי לדווח אם העדכון הצליח.

51

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

51

תרגיל

- הגדירו **מבנה עם שדות: שם, גיל, ציון**

- כתבו פונקציה שמקבלת מצביע למבנה וציון חדש ומחזירה:

- 0 אם העדכון לא הצליח (ציון לא בטווח 100..0)
- 1 אם העדכון הצליח

- כתבו גם פונקציה הדפסה שמקבלת מבנה ומדפיסה בשלוש שורות: שם, גיל, ציון.

- **ב־ main הדגימו:**

- ניסיון עדכון לא חוקי (למשל 150) והדפסו הודעת כשל.
- ניסיון עדכון חוקי (למשל 95.5) והדפסו הודעת הצלחה.
- בכל פעם הדפסו לפני/אחרי עם `printStudent`.

52

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

52

השוואת מבנים

```
struct Student {
    char name[20];
    int age;
    float grade;
};
```

```
struct Student s1 = {"Roni", 17, 91.0};
struct Student s2 = {"Roni", 17, 91.0};
```

```
if (s1 == s2)
    printf("חשוים");
else
    printf("חשונים");
```

מה יהיה
פלט
התוכנית??

53

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

53

איך כן משווים?

```
int equalStudents(struct Student a, struct Student b) {
    if (a.age != b.age) return 0;
    if (a.grade != b.grade) return 0;
    if (strcmp(a.name, b.name) != 0) return 0;
    return 1; // כל השדות שווים
}
```

```
struct Student s1 = {"Roni", 17, 91.0};
struct Student s2 = {"Roni", 17, 91.0};
```

```
if (equalStudents(s1, s2))
    printf("שוים\n");
else
    printf("שונים\n");
```

54

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

54

תרגיל

- הגדירו מבנה Car עם שדות: brand[20], year, price.
- כתבו פונקציה equalCars שמשווה בין שני רכבים:
- מחזירה 1 אם כל השדות שווים.
- אחרת מחזירה 0.
- ב־ main צרו שני רכבים:
- אחד: "Toyota", 2019, 75000
- שני: "Toyota", 2019, 75000
- בדקו והשוו ביניהם.

55

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

55

הרחבה לתרגיל (לאחר פתרון של המרצה)

- נסו לשנות רק את השנה של אחד הרכבים . מה יקרה?
- נסו לשנות את המותג ל "Honda" מה יקרה?
- האם אפשר להשוות רכבים רק לפי מחיר?
- (רמז: צריך לכתוב פונקציה אחרת).

56

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

56

פרק 7- מבני נתונים:

4. מערכים למבנים

57

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

16 אוגוסט 25

57

מבוא

- עד עכשיו הכרנו מבנה בודד
- למשל הגדרנו מבנה של סטודנט (struct Student) אבל אנחנו רוצים להתייחס לכיתה/קורס, ויש שם יותר מסטודנט אחד - **נוח לשמור אותם במערך.**
- מערך של מבנים = כמו מערך רגיל, רק שכל תא הוא מבנה שלם.

58

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

58

תרגיל

- הגדירו מבנה Student עם שדות: שם, גיל, ציון.
- צרו מערך של 3 תלמידים והכניסו לתוכו נתונים התחלתיים.
- הדפיסו את כל התלמידים אחד אחרי השני.

59

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

59

תרגיל המשך.

כתבו פונקציה שמקבלת מערך תלמידים ואת מספרם, ומחזירה את שם התלמיד בעל הציון הגבוה ביותר.

60

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

60

המשך תרגיל

כתבו פונקציה שמקבלת מערך תלמידים, גודל המערך ותוספת נקודות הפונקציה תוסיף את הנקודות לכל תלמיד, עד מקסימום 100.

61

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

61

תרגול מסכם

62

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

62

פרק 7- מבני נתונים:

5. יצירת טיפוסים בעזרת הוראה typedef

63

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

63

מה זה typedef ?

- הוראה מיוחדת ב־C שמאפשרת לתת שם חדש לטיפוס קיים.
- לא יוצרת טיפוס חדש, אלא רק שם נוסף נוח יותר.
- מונעת קוד מסורבל וחוזר.

64

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

64

שימוש ב- typedef עם struct

• עם שימוש ב typedef: <pre>typedef struct { char name[20]; int age; float grade; } Student; </pre> • עכשיו פשוט אפשר: <pre>Student s1;</pre>	• עד עכשיו הגדרנו מבנה באופן הבא: <pre>struct Student { char name[20]; int age; float grade; }; </pre> ואז כל פעם שרצינו להגיד סטודנט: <pre>struct Student s1;</pre>
---	--

65

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

65

למה זה טוב?

- חוסך כתיבה מיותרת של struct.
- מקל על הקריאות של הקוד.
- שימושי במיוחד במערכים, פונקציות ומבנים מורכבים.

66

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

66

דוגמה מס' 1

```
typedef struct {
    char brand[20];
    int year;
} Car;
```

```
Car myCar = {"Toyota", 2020};
```

67

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

67

דוגמה מס' 2

```
typedef struct {
    char title[30];
    float price;
} Book;
```

```
Book *pBook;
```

68

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

68

חזרה על תרגיל קודם

תרגיל

- הגדירו מבנה Student עם שדות: שם, גיל, ציון.
- צרו מערך של 3 תלמידים והכניסו לתוכו נתונים התחלתיים.
- הדפיסו את כל התלמידים אחד אחרי השני.

69

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

69

#include <stdio.h>

// typedef הגדרה עם

typedef struct {

char name[20];

int age;

float grade;

} Student;

void printStudent(Student s) {

printf("שם: %s\n", s.name);

printf("גיל: %d\n", s.age);

printf("ציון: %f\n", s.grade);

}

int main() {

Student class[3] = {

{"Roni", 17, 91.0},

{"Dana", 16, 88.5},

{"Avi", 17, 72.0}

};

for (int i = 0; i < 3; i++) {

printStudent(class[i]);

}

return 0;

}

פתרון

70

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

70

חזרה על תרגיל קודם

תרגיל המשך.

כתבו פונקציה שמקבלת מערך תלמידים ואת מספרם, ומחזירה את שם התלמיד בעל הציון הגבוה ביותר.

71

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

25 אוגוסט 17

71

```
#include <stdio.h>
#include <string.h>
typedef struct {
    char name[20];
    int age;
    float grade;
} Student;
Student find(struct Student arr[], int n)
{
    Student best = arr[0];
    for (int i = 1; i < n; i++) {
        if (arr[i].grade > best.grade) {
            best = arr[i];
        }
    }
    return best;
}
```

```
int main() {
    Student class[3] = {
        {"Roni", 17, 91.0},
        {"Dana", 16, 88.5},
        {"Avi", 17, 72.0}
    };

    Student top = findTop(class, 3);
    printf("top: %s, %d", top.name, top.grade);
}
```

פתרון

מצגת זו נוצרה
cohen.bareket@gmail.com

25 אוגוסט 17

72

חזרה על תרגיל קודם

המשך תרגיל

כתבו פונקציה שמקבלת מערך תלמידים, גודל המערך ותוספת נקודות הפונקציה תוסיף את הנקודות לכל תלמיד, עד מקסימום 100.

73

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

73

```
#include <stdio.h>
#include <string.h>
typedef struct {
    char name[20];
    int age;
    float grade;
} Student;

void addPoints(Student arr[], int n, float bonus) {
    for (int i = 0; i < n; i++) {
        arr[i].grade += bonus;
        if (arr[i].grade > 100)
            arr[i].grade = 100;
    }
}
```

פתרון

74

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

74

פרק 7- מבני נתונים:

6. הקצאות דינמיות של מבנים ומבנים המצביעים לעצמם self-referential

75

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

75

מה זה "הקצאה דינמית" ולמה צריך?

- עד עכשיו יצרנו משתנים "רגילים" שבחיים שלהם נקבעים בזמן הקומפילציה/הרצה המקומית של הפונקציה.
- לפעמים אנחנו רוצים ליצור מבנה **בזמן ריצה** – כשהתוכנית כבר רצה, ומספר הפריטים לא ידוע מראש.
- בשביל זה יש: malloc הקצאה free - לשחרר
- זוכרים: תמיד מה ש־ malloc ממחזיר—שומרים במצביע תואם; כשלא צריך יותר—free.

76

מצגת זו נוצרה על ידי: אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

76

<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> typedef struct { char name[20]; int age; float grade; } Student; void printStudent(Student s){ printf("שם: %s\n", s.name); printf("גיל: %d\n", s.age); printf("ציון: %f\n", s.grade); }</pre>	<pre>Student* makeStudent(const char* n, int a, float g) { Student *p = (Student*)malloc(sizeof(Student)); if(!p) return NULL; // בדיקת כשל בהקצאה strcpy(p->name, n); p->age = a; p->grade = g; return p; } int main(){ Student *s = makeStudent("Roni", 17, 91.0f); if(!s){ printf("Allocation failed\n"); return 1; } printStudent(*s); // העברה בערך להדפסה free(s); // חשוב לשחרר! return 0; }</pre>
---	--

77

מצגת זו נוצרה על ידי אינג' ברקת כהן.
cohen.bareket@gmail.com

17 אוגוסט 25

77