

מדעי המחשב

2 יחידות לימוד

הוראות לנבחן

- א. משך הבחינה: שלוש שעות.
- ב. מבנה השאלון ומפתח ההערכה:
 פרק ראשון – יש לענות על חמש השאלות 1-5,
 לכל שאלה – 10 נקודות.
 פרק שני – יש לענות על שתיים מן השאלות 6-8,
 לכל שאלה – 15 נקודות.
 פרק שלישי – יש לענות על אחת מן השאלות 9-10,
 לשאלה – 20 נקודות.
 – (20x1) – 20 נקודות
 – סה"כ – 100 נקודות
- ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשב הניתן לתכנות.
- ד. הוראות מיוחדות: (1) כתוב בשפה אחת בלבד את כל התכניות שאתה נדרש לכתוב.
 (2) רשום על הכריכה החיצונית של המחברת את השפה שבה אתה כותב – Java או C#.

הערה: בתכניות שאתה כותב לא יודרו לך נקודות אם תכתוב אות גדולה במקום אות קטנה או להפך.

כחוב מחברת הבחינה בלבד, בעמודים נפרדים, כל מה שכרוזוך לכתוב ב**מיוצג** (ראשי פרקים, חישובים וכדומה).
 רשום "טיוטה" בראש כל עמוד טיוטה. רישום טיוטות כלשהן על דפים שמוחץ למחברת הבחינה עלול לגרום לפסילת הבחינה!

ההנחיות בשאלון זה מנוסחות בלשון זכר ומכוונות לנבחנות ולנבחנים כאחד.

בהצלחה!

/המשך מעבר לדף/

השאלות

שים לב: עליך לכתוב בשפה אחת בלבד את כל התכניות שאתה נדרש לכתוב.
 רשום על הכריכה החיצונית של המחברת את השפה שאתה כותב בה – Java או C#.
הערה: בכל שאלה שנדרשת בה קליטה, אין צורך לבדוק את תקינות הקלט.
לפותרים ב-Java: בכל שאלה שנדרשת בה קליטה, הנח שבתכנית כתובה ההוראה:
 Scanner input = new Scanner (System.in);

פרק ראשון (50 נקודות)

ענה על חמש השאלות 1-5 (לכל שאלה – 10 נקודות).
 1. לפיך מחלקה **Card** המייצגת קלף שתכונותיו: צבע – color, ומספר – num.
 המחלקה כתובה ב-Java וב-C#.
 במחלקה חסר מימוש הפעולה הבונה.

Java

```
public class Card
{
    private String color;
    private int num;
    public Card(String color, int num) {...}
}
```

C#

```
public class Card
{
    private string color;
    private int num;
    public Card(string color, int num) {...}
}
```

- א. השלם ב-Java או ב-C# את מימוש הפעולה הבונה של המחלקה **Card**.
 ב. הנח שנתונה פעולה ראשית במחלקה Program. כתוב בפעולה הראשית הוראות אשר ייצרו שני עצמים בשם card1 ו-card2 מטיפוס **Card**, שהצבעים שלהם שונים זה מזה והמספרים שלהם שונים זה מזה.

2. נתונה המחלקה **Student** שיש בה שתי תכונות: מספר התלמיד – `num`, וציון במבחן – `mark`.

במחלקה הוגדרה פעולה המחזירה `true` אם הציון הוא בין 0 ל-100 (כולל), אחרת – הפעולה מחזירה `false`.

public boolean isInRange() :Java כותרת הפעולה ב־Java

public bool IsInRange() :C# כותרת הפעולה ב־C#

במחלקה הוגדרה פעולה בונה שהכותרת שלה:

public Student(int num , int mark)

לכל תכונה הוגדרו ב־Java פעולות `get` ו־`set` וב־C# פעולות `Get` ו־`Set`.

לפניך קטע תכנית בפעולה הראשית במחלקת Program, הכתוב ב־Java וב־C#.

Java

```
1 Student b1 = new Student(1111 , 92);
2 Student b2 = new Student(3333 , 200);
3 if (b1.isInRange())
4     System.out.println("****");
5 else
6     b1.setMark(999) ;
7 if (b2.isInRange())
8     System.out.println("yes");
9 else
10    b2.setMark(999) ;
```

C#

```
Student b1 = new Student(1111 , 92);
Student b2 = new Student(3333 , 200);
if (b1.IsInRange())
    Console.WriteLine("****");
else
    b1.SetMark(999);
if (b2.IsInRange())
    Console.WriteLine("yes");
else
    b2.SetMark(999);
```

עקוב אחר קטע התכנית ורשום מה יהיה הפלט. במעקב הצג את העצמים ואת ערכי תכונותיהם.

3. לפיך פעולה במחלקה **Klum** כתובה ב־ Java וב־ C#.

Java	C#
<pre>public static void parpar(int a , int b) { int x = 0; if (a > b) { x = a * b; System.out.println("****"+x+"****"); } else { x = a + b; System.out.println(x); } }</pre>	<pre>public static void Parpar(int a , int b) { int x = 0; if (a > b) { x = a * b; Console.WriteLine("****"+x+"****"); } else { x = a + b; Console.WriteLine(x); } }</pre>

א. עקוב בעזרת טבלת מעקב אחר ביצוע הפעולה בעבור הערכים: 8 לפרמטר a ו־ 2 לפרמטר

b, וכתוב מה יהיה הפלט.

בטבלת המעקב יש לכלול:

עמודה לכל אחד מן המשתנים, עמודה שיצוין בה אם התנאי בפקודת if מתקיים או אינו מתקיים, ועמודה בעבור הפלט.

ב. תן ערכים לפרמטרים a ו־ b, שבעבורם תהיה שורת פלט אחת בלבד.

כתוב מהו הערך של a, מהו הערך של b ומהו הפלט.

4. כתוב ב־ Java או ב־ C#, קטע תכנית שיקלוט 218 זוגות של מספרים שלמים.

בעבור זוג המספרים הראשון שייקלט יחושב סכומם.

בעבור כל אחד משאר הזוגות, החל מהזוג השני, קטע התכנית יבצע את הפעולות האלה:

— יחשב את הסכום של זוג המספרים הנוכחי.

— אם הסכום של זוג המספרים הנוכחי קטן מהסכום של זוג המספרים שקדם לו, תוצג

כפלט ההודעה: "smaller".

5. נתונה המחלקה Stars , במחלקה זו נתון קטע התכנית שלמניך, הכתוב ב־ Java וב־ C#.

Java

```
int s, x;
s = 0;
for (int i = 1; i < 4 ; i++)
{
    x = input.nextInt();
    s = s + x;
    if (s > 10)
        System.out.println("***");
    else
        System.out.println("---");
}
```

C#

```
int s, x;
s = 0;
for (int i = 1; i < 4 ; i++)
{
    x = int.Parse(Console.ReadLine());
    s = s + x;
    if (s > 10)
        Console.WriteLine("***");
    else
        Console.WriteLine("---");
}
```

(שים לב: סעיפי השאלה בעמוד הבא.)

א. עקוב בעזרת טבלת מעקב אחר ביצוע קטע התכנית בעבור הקלט (משמאל לימין): 2, 8, 6.

ורשום מה יהיה הפלט.

בטבלת המעקב יש לכלול:

עמודה לכל אחד מהמשתנים שבקטע התכנית,

עמודה שבה יצוין אם התנאי בפקודת if מתקיים או אינו מתקיים,

ועמודה בעבור הפלט.

ב. תן דוגמה מייצגת לקלט שבעבורו הפלט יהיה:

/המשך בעמוד 7/

פרק שני (30 נקודות)

ענה על שתיים מן השאלות 6-8 (לכל שאלה – 15 נקודות).
6. הנח כי הוגדרה מחלקה בשם **A** בעלת שתי תכונות: **num** – מספר שלם, ו- **word** – מחרוזת. במחלקה זו הפעולה **toString** ב-**Java** ו- **ToString** ב-**C#** מחזירה מחרוזת שהיא שרשרת של ערך התכונה **word**, **num** פעמים ברצף. הדפסת עצם מוזמנת ב-**Java** את הפעולה **toString** של המחלקה, וב-**C#** את הפעולה **ToString** של המחלקה. לפניך הפעולה הראשית במחלקה **Program** כתובה ב-**Java** וב-**C#**. הפעולה משתמשת במחלקה **A**.

Java

```
public class Program
{
    public static void main(String[] args )
    {
        A a = new A() ;
        a.setNum(3) ;
        a.setWord("am...") ;
        System.out.println(a) ;
        A a1 = new A(a.getNum() , "la") ;
        System.out.println(a1) ;
    }
}
```

C#

```
public class Program
{
    public static void Main(string[] args)
    {
        A a = new A() ;
        a.SetNum(3) ;
        a.SetWord("am...") ;
        Console.WriteLine(a) ;
        A a1 = new A(a.GetNum() , "la" ) ;
        Console.WriteLine(a1) ;
    }
}
```

(שים לב: המשך השאלה בעמוד הבא.)

א. עליך להגדיר את המחלקה **A** כך שהפעולה הראשית תתבצע בלי שניאות הידור ו/או ריצה.

הגדר את המחלקה **A** על פי התת-סעיפים i-iv :

- i כתוב ב־ Java או ב־ C# את כותרת המחלקה ואת התכונות שלה.
- ii כתוב את הכותרות של שתי הפעולות הבונות של המחלקה **A**.
- iii ממש ב־ Java את כל פעולות `get` ו־ `set` או ב־ C# את כל פעולות `Get` ו־ `Set` הנחוצות לריצה של הפעולה הראשית.
- iv ממש ב־ Java את הפעולה `toString` או ב־ C# את הפעולה `ToString` כך שהיא תחזיר מחרוזת שהיא ערך התכונה `word` משורשר `num` פעמים.
לדוגמה: בפעולה הראשית הנתונה הפלט של הוראת ההדפסה הראשונה יהיה המחרוזת `am...am...am...`.

ב. מה יהיה הפלט של הוראת ההדפסה השנייה?

7. לפניך ממשק המחלקה **Baby** המייצגת תינוק. במחלקה ארבע תכונות: מספר תעודת הזהות – id מטיפוס מחרוזת; מין התינוק – gender ("F" מייצג נקבה, "M" מייצג זכר) מטיפוס מחרוזת; גובה התינוק – height מטיפוס שלם; משקל התינוק – weight מטיפוס ממשי.

תואר הפעולה	כותרת הפעולה ב־ Java
פעולה הבונה תינוק שמספר תעודת הזהות שלו id, מינו gender, גובהו height, ומשקלו weight.	public Baby(String id, String gender, int height, double weight)
פעולה הבונה תינוק שמספר תעודת הזהות שלו id, מינו "F", גובהו height, ומשקלו weight.	public Baby(String id, int height, double weight)
פעולה המחזירה את גובה התינוק.	public int getHeight()
פעולה המחזירה את משקל התינוק.	public double getWeight()
פעולה המוסיפה w למשקל התינוק.	public void updateWeight(double w)
פעולה המוסיפה h לגובה התינוק.	public void addHeight(int h)
פעולה המחזירה true אם התינוק other זהה בגובהו ובמשקלו לתינוק הנוכחי, אחרת – מחזירה false.	public boolean isSame(Baby other)

תואר הפעולה	כותרת הפעולה ב־ C#
פעולה הבונה תינוק שמספר תעודת הזהות שלו id, מינו gender, גובהו height, ומשקלו weight.	public Baby(string id, string gender, int height, double weight)
פעולה הבונה תינוק שמספר תעודת הזהות שלו id, מינו "F", גובהו height, ומשקלו weight.	public Baby(string id, int height, double weight)
פעולה המחזירה את גובה התינוק.	public int GetHeight()
פעולה המחזירה את משקל התינוק.	public double GetWeight()
פעולה המוסיפה w למשקל התינוק.	public void UpdateWeight(double w)
פעולה המוסיפה h לגובה לתינוק.	public void AddHeight(int h)
פעולה המחזירה true אם התינוק other זהה בגובהו ובמשקלו לתינוק הנוכחי, אחרת – מחזירה false.	public bool IsSame(Baby other)

(שים לב: המשך השאלה בעמוד הבא.)

- א. ממש במחלקה **Baby** :
 ב־ Java את הפעולה הבונה `public Baby(String id, int height, double weight)`
 או ב־ C# את הפעולה הבונה `public Baby(string id, int height, double weight)`
- ב. ממש במחלקה **Baby**, ב־ Java או ב־ C#, את הפעולה המוסיפה w למשקל תינוק.
- ג. ממש במחלקה **Baby**, ב־ Java או ב־ C#, את הפעולה הבודקת אם תינוק other זהה לתינוק הנוכחי בגובה ובמשקל.
- ד. לפניך קטע מפעולה ראשית במחלקה Program הכתוב ב־ Java וב־ C#. עקוב אחר ביצוע קטע הפעולה, ורשום את הפלט שיתקבל. לכל עצם רשום את השמות של כל התכונות שלו, את הערכים של כל התכונות ואת השינויים שהיו בהם.

<u>Java</u>	<u>C#</u>
<code>Baby b1 = new Baby("1111", "M", 65, 3.2);</code>	<code>Baby b1 = new Baby("1111", "M", 65, 3.2);</code>
<code>Baby b2 = new Baby("2222", "F", 70, 2.6);</code>	<code>Baby b2 = new Baby("2222", "F", 70, 2.6);</code>
<code>Baby b3 = new Baby("3333", 60, 3.5);</code>	<code>Baby b3 = new Baby("3333", 60, 3.5);</code>
<code>b1.updateWeight(0.3);</code>	<code>b1.UpdateWeight(0.3);</code>
<code>b1.addHeight(5);</code>	<code>b1.AddHeight(5);</code>
<code>b2.updateWeight(0.2);</code>	<code>b2.UpdateWeight(0.2);</code>
<code>b3.addHeight(10);</code>	<code>b3.AddHeight(10);</code>
<code>if (b1.isSame (b3))</code>	<code>if (b1.IsSame (b3))</code>
<code>System.out.println("***");</code>	<code>Console.WriteLine("***");</code>
<code>if (b2.isSame (b3))</code>	<code>if (b2.IsSame (b3))</code>
<code>System.out.println ("!!!");</code>	<code>Console.WriteLine("!!!");</code>
<code>else</code>	<code>else</code>
<code>System.out.println ("*!!*");</code>	<code>Console.WriteLine("*!!*");</code>

8. נתונה המחלקה לקוח - **Customer** שיש לה שתי תכונות:

שם לקוח - name מטיפוס מחרוזת

מספר טלפון של לקוח - telNum מטיפוס מחרוזת

הנח שלכל תכונה הוגדרו ב־ Java פעולות get ו־ set, וב־ C# פעולות Get ו־ Set.

א. כתוב ב־ Java או ב־ C# פעולה בונה של המחלקה **Customer**, שתקבל כפרמטרים ערכים לכל אחת משתי התכונות.

ב. נתונה המחלקה חנות - **Store** שיש לה שתי תכונות:

מערך חד־ממדי - arrCust מטיפוס **Customer**

מספר הלקוחות הנוכחי - current מטיפוס שלם, שערכו קטן מגודל המערך.

הנח שלכל תכונה הוגדרו ב־ Java פעולות get ו־ set, וב־ C# פעולות Get ו־ Set.

במחלקה **Store** הוגדרה הפעולה הבונה ב־ Java וב־ C#:

```
public Store()
{
    this.arrCust = new Customer[100];
    this.current = 0;
}
```

i כתוב ב־ Java או ב־ C# את כותרת המחלקה **Store** ואת התכונות שלה.

ii כתוב ב־ Java או ב־ C#, במחלקה **Store**, פעולה שתקבל לקוח - **Customer** ותוסיף אותו לחנות. הנח שיש מקום להוסיף את הלקוח.

iii כתוב ב־ Java או ב־ C# במחלקה Program, פעולה חיצונית שתקבל s מטיפוס **Store**, ומספר שלם num בין 0 ל־ 99.

הפעולה תחזיר את מספר הטלפון של הלקוח - **Customer**, ש־ num הוא המציין של מקומו במערך arrCust.

אם אין לקוח כזה - תחזיר הפעולה "no".

פרק שלישי (20 נקודות)

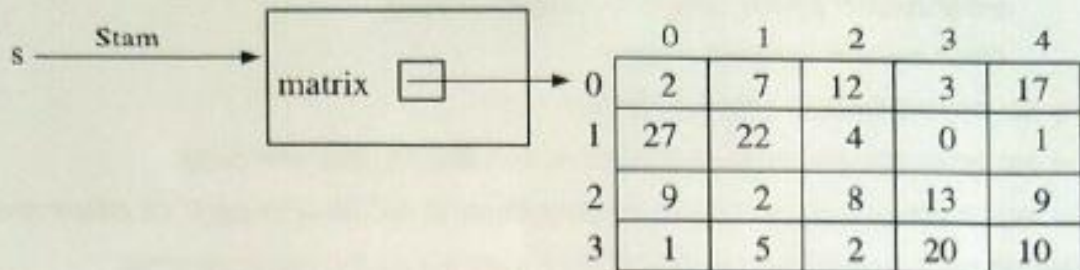
ענה על אחת מהשאלות 9-10.

9. נתונה מחלקה **Stam** שיש לה תכונה אחת – מערך דו-ממדי מטיפוס שלם בשם `matrix`. במחלקה הוגדרו, בין היתר, הפעולות האלה:

תיאור הפעולה	כותרת הפעולה ב־ Java
פעולה הבונה עצם מטיפוס Stam , וקולטת מספרים שלמים וגדולים מ־ 0 למערך <code>matrix</code> בגודל <code>row x col</code> .	<code>public Stam(int row, int col)</code>
פעולה המחזירה <code>true</code> אם כל איברי המערך <code>matrix</code> שונים זה מזה. אחרת – הפעולה מחזירה <code>false</code> .	<code>public boolean isDistinct()</code>
פעולה המקבלת מספר שלם <code>num</code> שנמצא במערך, ומחזירה את מספר השורה שבה נמצא אחד ממופעי המספר.	<code>public int rowNum (int num)</code>
פעולה המקבלת מספר שלם <code>num</code> שנמצא במערך, ומחזירה את מספר העמודה שבה נמצא אחד ממופעי המספר.	<code>public int colNum (int num)</code>
פעולה המחזירה את המספר הקטן ביותר במערך <code>matrix</code> .	<code>public int min()</code>

תיאור הפעולה	כותרת הפעולה ב־ C#
פעולה הבונה עצם מטיפוס Stam , וקולטת מספרים שלמים וגדולים מ־ 0 למערך <code>matrix</code> בגודל <code>row x col</code> .	<code>public Stam(int row, int col)</code>
פעולה המחזירה <code>true</code> אם כל איברי המערך <code>matrix</code> שונים זה מזה, אחרת – הפעולה מחזירה <code>false</code> .	<code>public bool IsDistinct()</code>
פעולה המקבלת מספר שלם <code>num</code> שנמצא במערך, ומחזירה את מספר השורה שבה נמצא אחד ממופעי המספר.	<code>public int RowNum (int num)</code>
פעולה המקבלת מספר שלם <code>num</code> שנמצא במערך, ומחזירה את מספר העמודה שבה נמצא אחד ממופעי המספר.	<code>public int ColNum (int num)</code>
פעולה המחזירה את המספר הקטן ביותר במערך <code>matrix</code> .	<code>public int Min()</code>

ההוראה `Stam s = new Stam(4, 5)` תיצור עצם `s`, שבו התכונה `matrix` היא מערך דו־ממדי בגודל 4×5 , ותקלוט ערכים למערך.
לדוגמה:



תתי־מערך־מספרי של איבר k, j (k מייצג שורה, j מייצג עמודה) הנמצא במערך, הוא מערך דו־ממדי שהאיבר k, j הוא הפינה השמאלית העליונה שלו, והפינה הימנית התחתונה שלו היא הפינה הימנית התחתונה של המערך.
לדוגמה, במערך `matrix` שבעצם `s` ה**תתי־מערך־המספרי** של האיבר $1,1$ מסומן באפור.

	0	1	2	3	4
0	2	7	12	3	17
1	27	22	4	0	1
2	9	2	8	13	9
3	1	5	2	20	10

תתי־מערך־מספרי זוגי של איבר k, j הוא **תתי־מערך־מספרי** של האיבר, שכל איבריו זוגיים (כולל האיבר k, j).

א. כתוב פעולה במחלקה `Stam`, שתקבל:

מספר שלם k המציין שורה במערך

מספר שלם j המציין עמודה במערך

הפעולה תחזיר `true` אם ה**תתי־מערך־המספרי** של האיבר k, j הוא **תתי־מערך־מספרי זוגי**. אחרת – תחזיר הפעולה `false`.

ב. כתוב פעולה ראשית במחלקה `Program` אשר תקלוט שני מספרים שלמים, ותיצור עצם מסוג `Stam` שתכונתו מערך דו־ממדי שהמספר הראשון שנקלט הוא מספר השורות שבו, והמספר השני שנקלט הוא מספר העמודות שבו. הפעולה תבדוק אם כל הערכים במערך שונים זה מזה. אם לא – תדפיס הודעה מתאימה, אם כן – תבדוק אם ה**תתי־מערך־המספרי** של האיבר שהערך שלו הוא הקטן ביותר במערך, הוא **תתי־מערך־מספרי זוגי**.

הפעולה תדפיס הודעה מתאימה.

עליך להשתמש בפעולה שכתבת בסעיף א.

אתה יכול להשתמש בפעולות המחלקה `Stam` בלי לממש אותן.

10. בעיר מסוימת הוחלט לרכז את הנתונים על התושבים במערכת ממוחשבת.

לשם כך הוגדרה המחלקה משפחה – **Family**, שיש לה התכונות האלה:

שם משפחה – name

מספר תלמידי התיכון במשפחה – numStudents

מספר הנפשות במשפחה – num

יש שני סוגים של הנחות בתשלום המסים:

– אם מספר הנפשות במשפחה גדול מ-6, היא מקבלת 100 ש"ח הנחה.

– אם יש במשפחה תלמידי תיכון, היא מקבלת הנחה של 40 ש"ח בעבור כל תלמיד תיכון.

משפחה הזכאית לשני סוגי ההנחות, תקבל רק את סוג ההנחה הגבוה מביניהם.

לפניך חלק מן הממשק של המחלקה **Family**:

כותרת הפעולה ב־ Java	תיאור הפעולה
public Family(String name, int numStudents, int num)	פעולה הבונה משפחה: שם המשפחה name, מספר תלמידי התיכון numStudents, מספר הנפשות במשפחה num.
public int getNumStudents()	פעולה שמחזירה את מספר תלמידי התיכון במשפחה.
public int reduction()	פעולה שמחזירה את ההנחה בתשלום המסים שהמשפחה זכאית לה. אם אינה זכאית להנחה, הפעולה מחזירה 0.

כותרת הפעולה ב־ C#	תיאור הפעולה
public Family(string name, int numStudents, int num)	פעולה הבונה משפחה: שם המשפחה name, מספר תלמידי התיכון numStudents, מספר הנפשות במשפחה num.
public int GetNumStudents()	פעולה שמחזירה את מספר תלמידי התיכון במשפחה.
public int Reduction()	פעולה שמחזירה את ההנחה בתשלום המסים שהמשפחה זכאית לה. אם אינה זכאית להנחה, הפעולה מחזירה 0.

(שים לב: המשך השאלה בעמוד הבא.)

א. ממש במחלקה **Family**, ב־ Java או ב־ C#, את הפעולה המחשבת את ההנחה בתשלום המסים שהמשפחה זכאית לה.

בנוסף הוגדרה המחלקה עיר – **City** שיש לה התכונות האלה:

מערך חד־ממדי מטיפוס **Family** בגודל 100,000, מספר המשפחות הגרות בעיר `numFamilies`, `numFamilies` קטן מ־ 100,000 או שווה לו.

ב. נתונה המחלקה `Program`. במחלקה זו יש פעולה ראשית שבה הוגדרה העיר `ct` מטיפוס **City**.

כתוב בפעולה הראשית, ב־ Java או ב־ C#, קטע תכנית שידפיס את מספר המשפחות בעיר שלא קיבלו הנחה.