

Computer Science

2 points

מדעי המחשב

2 יחידות לימוד

Instructions for examinees

א. Duration of the exam: Three hours

הוראות לנבחן

א. משך הבחינה: שלוש שעות.

ב. Exam structure and breakdown of points:

This exam has three parts.

Section 1 — Answer all five questions 1-5,
each question — 10 points

(5 × 10) — 50 points

Section 2 — Answer two of the questions 6-8,
each question — 15 points

(2 × 15) — 30 points

Section 3 — Answer one of the questions 9-10,
one question — 20 points

(1 × 20) — 20 points

Total — 100 points

ב. מבנה השאלון ומפתח ההערכה:
בשאלון זה יש שלושה פרקים.

פרק ראשון — יש לענות על חמש השאלות 1-5,

לכל שאלה — 10 נקודות

— (10×5) 50 נקודות

פרק שני — יש לענות על שתיים מן השאלות 6-8,

לכל שאלה — 15 נקודות

— (15×2) 30 נקודות

פרק שלישי — יש לענות על אחת מן השאלות 9-10,

לשאלה — 20 נקודות

— (20×1) 20 נקודות

סה"כ — 100 נקודות

ג. Material that may be used during the exam:

All material, except a programmable computer.

ג. חומר עזר מותר בשימוש:

כל חומר עזר, חוץ ממחשב הניתן לתכנות.

ד. Special instructions:

(1) Write all of the programs you are asked to
write in one programming language only.

(2) **On the outside cover** of the answer booklet,
write which programming language you are
using — Java **or** C#.

ד. הוראות מיוחדות:

(1) כתוב בשפה אחת בלבד את כל התוכניות שאתה
נדרש לכתוב.

(2) **רשום על הכריכה החיצונית** של המחברת את שפת
המחשב שבה אתה כותב — Java **או** C#.

Note: In the programs you write, you will not lose
points for writing a capital letter instead of a lower-
case letter or vice versa.

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

כתוב במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונך לכתוב כטייטה (ראשי פרקים, חישובים וכדומה). רשום
"טייטה" בראש כל עמוד טייטה. רישום טייטות כלשהן על דפים מחוץ למחברת הבחינה עלול לגרום לפסילת הבחינה!

Write only in the answer booklet. Use separate pages for everything you want to use as
draft material (outlines, calculations, etc.). Write "טייטה" at the top of each draft page. Writing
drafts on pages outside the answer booklet may lead to the disqualification of the exam!

Good Luck!

בהצלחה !

Questions

Note: Use only one programming language for all of the programs you are asked to write.

On the outside cover of the answer booklet, write which programming language you are using – Java or C#.

Note: For each question that requires input, there is no need to check the validity of the input.

For those working in Java: For every question that requires input, assume that the following command is written in the program:

Scanner input = new Scanner (System.in);

Section 1 (50 points)

Answer all five questions 1-5 (each question – 10 points).

1. The class **Student**, which represents a student, is shown below. This class has two attributes: the student's name - name - and the student's year of birth - year .

The class is written in Java and in C#. The class does not include the implementation of the constructor method [הפעולה הבונה].

Java

```
public class Student
{
    private String name;
    private int year;
    public Student(String name, int year) {...}
}
```

C#

```
public class Student
{
    private string name;
    private int year;
    public Student(string name, int year) {...}
}
```

- א. Using Java or C#, complete the implementation of the constructor method for the class **Student**.
- ב. Assume that a main method is given for the class Program. Using Java or C#, write instructions in the main method to produce two objects of type **Student** named st1 and st2 , which have different names, but the same year of birth.

2. A segment of the main method of the class Program is shown below written in Java and in C#.

A one-dimensional array of integers `arr` is defined in the main method.

The array contains integers greater than 0.

Java	C#
<pre>for (int i = 0; i < arr.length; i = i + 2) { if ((arr[i] > 9) && (arr[i] < 100)) { arr[i] = 0; } }</pre>	<pre>for (int i = 0; i < arr.Length; i = i + 2) { if ((arr[i] > 9) && (arr[i] < 100)) { arr[i] = 0; } }</pre>

- א. An array `arr` is shown below.

	0	1	2	3	4	5	6	7
arr	12	2	324	33	67	888	9	5

Using a trace table, follow the execution of the method segment.

The trace table should include:

a column for `i`, a column for `arr[i]` and a column that indicates whether the condition in the `if` command has or has not been met.

Present the array after the execution of the program segment.

- ב. Give a representative example of an array of 8 integers in which all of the values are different from 0 after the method segment has been run.

3. Using Java or C#, write an external method that will accept a boolean variable as input, as well as a three-digit integer greater than 0.

If the value of the boolean variable is `true`, the method will return the sum of the digits of the number. Otherwise, the method will return the product [מכפלה] of the number's digits.

4. Given: a class **Worker**, which represents a worker

The class has two attributes: name and salary .

In this class, a constructor method with the following declaration is defined:

in Java: public Worker(String name, int salary)

in C#: public Worker(string name, int salary)

In addition, in the class, a method is defined with the following declaration:

in Java: public boolean isBig()

in C#: public bool IsBig()

The method returns true if the salary is greater than 9000 shekels.

Otherwise, the method returns false .

For each attribute, get and set methods are defined in Java and Get and Set methods are defined in C#.

A segment from a main method of the class Program is shown below written in Java and in C#.

Java

```
Worker w1 = new Worker("Moshe", 10800);
Worker w2 = new Worker("Haim", 4800);
if (w1.isBig() && w2.isBig())
    System.out.println("Happy");
else
{
    System.out.println ("Unhappy");
    if (!w1.isBig())
        System.out.println (w1.getName());
    if (!w2.isBig())
        System.out.println (w2.getName());
}
```

C#

```
Worker w1 = new Worker("Moshe", 10800);
Worker w2 = new Worker("Haim", 4800);
if (w1.IsBig() && w2.IsBig())
    Console.WriteLine("Happy");
else
{
    Console.WriteLine("Unhappy");
    if (!w1.IsBig())
        Console.WriteLine (w1.GetName());
    if (!w2.IsBig())
        Console.WriteLine (w2.GetName());
}
```

Trace the execution of the method segment and write what the output will be. In your trace, present the objects and the values of their attributes. Next to each object, write what will be returned for it from the `isBig` method in Java or the `IsBig` method in C#.

5. In a certain city, it was decided that every building that has more than 4 apartments must have a building committee.

A class **Building** was defined. One of the attributes of this class is the number of apartments in the building: `private int numAps;`

- א. Using Java or C#, write a method in the class **Building** that will return `true` if a building committee is required and `false` if it is not.
- ב. In the main method of the class **Test**, a one-dimensional array `bdng` of type **Building** is defined. Each item in the array represents a building in a certain neighborhood of the city.
Using Java or C#, in the main method, write a program segment that will count and present as output the number of buildings in the neighborhood that should have a building committee.
Use the method that you wrote for Item א.

Section 2 (30 points)

Answer two of the questions 6-8 (each question – 15 points).

6. A decision was made to conduct a survey in 248 cities/towns, in order to examine residents' levels of satisfaction with their local schools. A community participated in the survey only if it was deemed appropriate for the survey, based on whether or not it met several conditions. Each year, a new set of conditions was formulated. The class **City** is shown below written in Java and in C#.

Java

```
public class City
{
    private String name; // name of the city/town
    private int popul; // number of residents in the city/town
    private int numOfSchools; // number of schools in the city/town
    public City(String name, int popul, int numOfSchools) {...} // constructor method
    public boolean isFit(){...} // method that returns true if the city/town is appropriate;
                                otherwise, it returns false
}
```

C#

```
public class City
{
    private string name; // name of the city/town
    private int popul; // number of residents in the city/town
    private int numOfSchools; // number of schools in the city/town
    public City(string name, int popul, int numOfSchools) {...} // constructor method
    public bool IsFit(){...} // method that returns true if the city/town is appropriate;
                                otherwise, it returns false
}
```

Assume that for each attribute, set and get methods are defined in Java and Set and Get methods are defined in C#.

א. Using Java or C#, write a main method in the class Program that will receive the following as input for each of the 248 cities/towns:

- name of the city/town
- number of residents in the city/town
- number of schools in the city/town

The main method will produce an object of type **City** for each city/town.

The main method will present the following as output:

- for each city/town: the name of the city/town and whether or not it meets the requirements for inclusion in the survey.
- the number of cities/towns that do not meet the requirements for inclusion in the survey.

ב. For 2015, the following conditions were set:

The survey would include only cities/towns that had more than 3 schools and more than 5,000 residents.

Implement the method `isFit` in Java or the method `IsFit` in C#, in keeping with the conditions set for 2015.

7. Part of the interface [ממשק] for the class **Pen** is shown below. The class has four attributes: the color of the pen – color – which is a string; the manufacturer – made – which is a string; the price of the pen – price – which is real [טיפוס ממשי] and the weight of the pen – weight – which is real.

Method declaration in Java	Description of the method
public Pen(String made)	Constructor method for a "red" pen from the manufacturer made , with a price of 10.0 shekels and a weight of 25.0 grams
public Pen(String color, String made, double price, double weight)	Constructor method for a pen of the color color , manufacturer made , price price and weight weight
public String getMade()	Method that returns the name of the manufacturer of the pen
public double getPrice()	Method that returns the price of the pen
public void setPrice(double x)	Method that updates the price of the pen to x
public boolean isSameMade(Pen other)	Method that returns true if the manufacturer of the pen other is the same as the manufacturer of the current pen; otherwise, it returns false
public boolean isSamePrice(Pen other)	Method that returns true if the price of the pen other is the same as the price of the current pen; otherwise, it returns false

Method declaration in C#	Description of the method
public Pen(string made)	Constructor method for a "red" pen from the manufacturer made , with a price of 10.0 shekels and a weight of 25.0 grams
public Pen(string color, string made, double price, double weight)	Constructor method for a pen of the color color , manufacturer made , price price and weight weight
public string GetMade()	Method that returns the name of the manufacturer of the pen
public double GetPrice()	Method that returns the price of the pen
public void SetPrice(double x)	Method that updates the price of the pen to x
public bool IsSameMade(Pen other)	Method that returns true if the manufacturer of the pen other is the same as the manufacturer of the current pen; otherwise, it returns false
public bool IsSamePrice(Pen other)	Method that returns true if the price of the pen other is the same as the price of the current pen; otherwise, it returns false

- א. Implement the following in the class **Pen**:
the constructor method `Pen(String made)` in Java or
the constructor method `Pen(string made)` in C#.
- ב. In the class **Pen**, implement the method `setPrice` in Java or the method `SetPrice` in C#.
- ג. In the class **Pen**, implement the method `isSamePrice` in Java or the method `IsSamePrice` in C#.
- ד. A segment of a main method of the class `TestProg` is shown below written in Java and in C#.
Trace the execution of the method segment and write the output that will be obtained.
In your trace, present the objects and the values of their attributes.
Next to each object, write the names of the methods that were used in it and, for each method, write the value that is returned.

Java

```
Pen a1 = new Pen("red" , "aaa" , 10.0 , 5.5);  
Pen a2 = new Pen("bbb");  
Pen a3 = new Pen("black" , "ccc" , 8.0 , 12.5);  
a1.setPrice(15.0);  
if(!(a1.isSamePrice(a3)))  
    if(a2.isSameMade(a1))  
        System.out.println("YES");  
    else  
        System.out.println("NO");  
else  
    System.out.println("OK");
```

C#

```
Pen a1 = new Pen("red" , "aaa" , 10.0 , 5.5);  
Pen a2 = new Pen("bbb");  
Pen a3 = new Pen("black" , "ccc" , 8.0 , 12.5);  
a1.SetPrice(15.0);  
if(!(a1.IsSamePrice(a3)))  
    if(a2.IsSameMade(a1))  
        Console.WriteLine("YES");  
    else  
        Console.WriteLine("NO");  
else  
    Console.WriteLine("OK");
```

8. The tour company Enjoy the Trip is organizing a trip for families. Families participating in the trip are required to pay for the guide and for the meal that is provided during the trip.
The price for the guide is fixed at 100 shekels per family.
The meal prices are as follows:
30 shekels per participant — for families with 3 or fewer participants
28 shekels per participant — for families with 4-5 participants
26 shekels per participant — for families with more than 5 participants
The class **Family** is shown below written in Java and in C#.

Java

```
public class Family
{
    private int num; // number of participants from the family
    private String familyName; // family name
    public Family(int num, String familyName) {...}
}
```

C#

```
public class Family
{
    private int num; // number of participants from the family
    private string familyName; // family name
    public Family(int num , string familyName) {...}
}
```

For each attribute, get and set methods are defined in Java and Get and Set methods are defined in C#.

- א. In the class **Family**, write a method `calcPrice` in Java or a method `CalcPrice` in C# that will return the amount the family needs to pay for the trip.
- ב. Using Java or C#, in the class Program, write a main method that will receive the following as input for each family participating in the trip: number of participants and family name. The input will end when 0 is entered for the number of participants from the family.
The main method will produce an object of type **Family** for each family.
The main method will present the following as output:
— for each family: the family name and the amount that family will pay for the trip
— the total number of trip participants
— the largest amount a family will pay for the trip.
Use the method that you wrote for Item א.

/Continued on page 11/

Note: The test continues on the next page.

Section 3 (20 points)

Answer one of the questions 9-10.

9. Given: the class **Card**, which represents a card. This class has two attributes: `value`, which is an integer between 1 and 9 and `color`, which is an integer, with 1 representing red, 2 representing yellow, 3 representing green and 4 representing blue.

Method declaration in Java	Description of the method
<code>public Card(int value, int color)</code>	Method that constructs a card with a value <code>value</code> and color <code>color</code>
<code>public boolean isSame(Card other)</code>	Method that returns <code>true</code> if the value of the card <code>other</code> is the same as that of the current card and the color of the card <code>other</code> is <u>also</u> the same as that of the current card; otherwise, it returns <code>false</code>
<code>public int compareVal(Card other)</code>	Method that returns 1 — if the value of the current card is greater than that of the card <code>other</code> 2 — if the value of the current card is less than that of the card <code>other</code> 0 — if the value of the current card and that of the card <code>other</code> are identical

Method declaration in C#	Description of the method
<code>public Card(int value, int color)</code>	Method that constructs a card with a value <code>value</code> and color <code>color</code>
<code>public bool IsSame(Card other)</code>	Method that returns <code>true</code> if the value of the card <code>other</code> is the same as that of the current card and the color of the card <code>other</code> is <u>also</u> the same as that of the current card; otherwise, it returns <code>false</code>
<code>public int CompareVal(Card other)</code>	Method that returns 1 — if the value of the current card is greater than that of the card <code>other</code> 2 — if the value of the current card is less than that of the card <code>other</code> 0 — if the value of the current card and that of the card <code>other</code> are identical

Given: the class **Deck**, which has one attribute — a one-dimensional array `allCards` of type **Card**.

Method declaration in Java	Description of the method
<code>public Deck()</code>	Constructor method for a deck of cards: the method defines an array <code>allCards</code> that is 36 in size and contains 36 cards. No two cards in the array are the same.
<code>public Card seeCard(int i)</code>	Method that returns the card located in position <code>i</code> of the array <code>allCards</code>

Method declaration in C#	Description of the method
<code>public Deck()</code>	Constructor method for a deck of cards: the method defines an array <code>allCards</code> that is 36 in size and contains 36 cards. No two cards in the array are the same.
<code>public Card SeeCard(int i)</code>	Method that returns the card located in position <code>i</code> of the array <code>allCards</code>

8. In the class **Deck**, implement the method `seeCard` in Java or the method `SeeCard` in C#.

A card game for two players is described below.

At the start of the game, each of the players has a deck of cards.

Each turn: Each player randomly chooses a number `i` between 0 and 35 (inclusive) and presents the card located at position `i` of the array `allCards` from his or her deck of cards.

- If the two cards are identical (in value and color), each player receives one point.

- A player who has a card whose value is greater than the card of the other player receives 3 points.

Also:

- Any player who picks a green card receives 2 points.

The game finishes when one of the players has earned at least 28 points.

9. Using Java or C#, write a segment of a main method in the class Program that will simulate the card game.

The program segment will print the number of points that each player accumulated by the end of the game.

You should use methods from the classes **Card** and **Deck**, but you are not required to implement them.

Assume that in the classes **Card** and **Deck**, for each attribute, `get` and `set` methods are defined in Java and `Get` and `Set` methods are defined in C#.

10. A sparse matrix [מטריצה דלילה] is a two-dimensional array of integers in which most of the values are 0's.

As an example, see the two-dimensional 4×5 array shown below.

	0	1	2	3	4
0	0	1	0	0	0
1	0	0	0	0	9
2	0	0	0	0	0
3	7	0	0	0	0

In order to save space in the memory of a computer, it is possible to save only those items in the two-dimensional array that are different from 0. The value of each item is saved and so is its position in the two-dimensional array: row number and column number.

For the two-dimensional array shown in the example, the 3 items that are different from 0 will be saved like this:

value = 1	value = 9	value = 7
row = 0	row = 1	row = 3
col = 1	col = 4	col = 0

The class **Item**, which represents an item in the two-dimensional array that is different from 0, was defined. Its attributes are: the value of the item in the two-dimensional array and its position, that is, its row number and column number.

- ⌘. (1) Using Java or C#, write the declaration and the attributes of the class **Item**.
- (2) Using Java or C#, write a constructor method in the class **Item** that will accept values as parameters for each attribute.

An additional class, **Sparse**, was defined with the following attributes:
a one-dimensional array `itemAr` of type **Item**,
`rows` — an integer representing the number of rows in the **sparse matrix**,
and
`columns` — an integer representing the number of columns in the
sparse matrix.

- ב. Using Java or C#, write the declaration of the class **Sparse** and its attributes.

In the class **Sparse**, a given method has the following declaration
in Java: `public int countNoZero(int[ ][ ] mat)`
in C#: `public int CountNoZero(int[ , ] mat)`

The method accepts a two-dimensional array and returns the number of items in that array that are different from 0.

- ג. Using Java or C#, write a constructor method in the class **Sparse** that will accept as a parameter a two-dimensional array of integers that represents a **sparse matrix**.
Use the given method.

Good Luck!

All rights reserved by the State of Israel.
Copying or publication without the consent of the
Ministry of Education is prohibited.

בהצלחה!

זכות היוצרים שמורה למדינת ישראל
אין להעתיק או לפרסם אלא ברשות משרד החינוך