

מבוא לתכנות
למערכות מידע
202-1-1041

מבוא למדעי
המחשב
202-1-1011

סמסטר א' תשס"ט
מבחן מועד א'

מר איתי בר-יוסף
מר טל גרינשפון
גב' דיקלה דותן
ד"ר מיכל זיו יוקלסון
פרופ' משה זיפר
מר שי זקוב
פרופ' מיכאל קודיש
ד"ר צחי רוזן

משך המבחן: שלוש שעות.
חומר עזר אסור.

במבחן זה 5 שאלות. יש לענות על כולן.

אנא רשמו את תשובותיכם בטופס התשובות בלבד. המחברת שקיבלתם הנה מחברת טיוטה בלבד ולא תימסר כלל לבדיקה. הקפידו לרשום על טופס הבחינה גם את מספר הנבחן וגם את מספר החדר בו אתם נבחנו. מספר השורות העומדות לרשותכם בגוף השאלון רומז על אורך התשובה הנדרשת. הקפידו על כתב יד ברור. תשובות מסורבלות או ארוכות מדי לא יזכו בניקוד מלא. מותר להסתמך על סעיפים קודמים גם אם לא עניתם עליהם.

בשאלות בהן לא מצוין אחרת, ניתן לבחור בפתרון רקורסיבי או פתרון שאינו רקורסיבי, לבחירתכם. **שימו לב:** החשיבות העליונה היא לנכונות הקוד. מאידך, יעילות וסגנון חשובים גם הם, ולכן תשובה יעילה ומסוגנת תזכה בציון גבוה יותר.

בדף האחרון של המבחן ישנה תזכורת למספר פונקציות חשובות, בהן תוכלו להשתמש במהלך המבחן.

בהצלחה!!!

דף תשובות

שאלה 1 (20 נק')

```
public static void subsets(int n, int k) {
    subsets(n, k, "");
}

public static void subsets(int n, int k, String s) {
    if (k == 0)
        System.out.println(s);
    else if (n > 0) {
        subsets(n - 1, k, s);
        subsets(n - 1, k - 1, n + s);
    }
}
```

שאלה 2 (20 נק')
השלימו את הפונקציה הרקורסיבית הבאה.

```
public String getPath(Object obj) {  
    if (this.data.equals(obj))  
        return data.toString() + " ";  
  
    String leftPath="",rightPath="";  
  
    if (left!=null)  
        leftPath=left.getPath(obj);  
    if (right!=null)  
        rightPath=right.getPath(obj);  
    if (!(leftPath.equals("")))   
        return this.data + " " + leftPath;  
    if (!(rightPath.equals("")))   
        return this.data + " " + rightPath;  
    return "";  
  
}
```

שאלה 3 (20 נק')

```
public class RandomIterator implements Iterator{
```

```
    Object[] arr;
```

```
    int size;
```

```
    public RandomIterator(Object[] inputArr){
```

```
        arr=new Object[inputArr.length];
```

```
        for (int i=0 ;i < inputArr.length;i++)  
            arr[i]=inputArr[i];
```

```
        size=arr.length;
```

```
    }
```

```
    public Object next(){
```

```
        int randomIndex = (int)(Math.random()*(size));
```

```
        Object tmp;
```

```
        Object ans = arr[randomIndex];
```

```
        tmp = arr[randomIndex];
```

```
        arr[randomIndex] = arr[size-1];
```

```
        arr[size-1] = tmp;
```

```
        size = size - 1;
```

```
        return ans;
```

```
    }
```

```
    public boolean hasNext(){
```

```
        return size>0;
```

```
    }
```

```
}//class
```

סעיף א' (5 נקודות):

סעיף ב' (5 נקודות):

סעיף ג' (5 נקודות):

סעיף ד' (5 נקודות):

שאלה 4 (20 נק')

```
public class SortedList {  
  
    private ComparableLink sentinal;  
  
    public SortedList(){  
        sentinal=new ComparableLink("sentinal");  
    }  
    public boolean isEmpty(){  
        return sentinal.next==null;  
    }  
    public boolean isSorted(){  
        if (this.isEmpty())  
            return true;  
        return sentinal.next.isSorted();  
    }  
}
```

סעיף א' (10 נקודות):

```
public void add(Comparable data){  
  
    ComparableLink l=new ComparableLink(data);  
    ComparableLink tmp=sentinal;  
  
    while(tmp.next!=null && data.compareTo(tmp.next.data)>0)  
        tmp=tmp.next;  
    l.next=tmp.next;  
    tmp.next=l;  
    }  
} //class SortedList
```

סעיף ב' (10 נקודות):

```
public class ComparableLink {  
    public ComparableLink next;  
    public Comparable data;  
  
    public ComparableLink (Comparable data, ComparableLink next){  
        this.data=data;  
        this.next=next;  
    }  
    public ComparableLink (Comparable data){  
        this(data,null);  
    }  
  
    public boolean isSorted(){  
  
        if (next==null)  
            return true;  
    }  
}
```

```
        if (data.compareTo(next.data)>0)
            return false;
        return next.isSorted();
    }
} //class ComparableLink
```

שאלה 5 (20 נק')

בשאלה זו 5 סעיפים (א-ה). בסעיפים א'-ד' עליכם לבחור תשובה אחת נכונה מבין ארבע האפשרויות. עליכם להקיף בעיגול את התשובה שבחרתם (שתי נקודות) ולענות בקצרה על השאלות המצורפות לסעיף שבחרתם (עוד שתי נקודות).

סעיף א' (4 נקודות)

הניחו כי `<a>` מוחלף בקוד הבא:

```
p.printClassName1();
```

מה יקרה כאשר ננסה להריץ את התכנית?

(1) תודפס המחרוזת "Person".

(2) תודפס המחרוזת "Student".

(3) הקוד לא יעבור קומפילציה (*בתשובה לשאלה זו התייחסו רק לקוד בשטה `printClassName1()` במחלקה `Person`)

(4) התכנית תיפול בזמן ריצה.

תשובה: (1) – השיטה `printClassName1()` נמצאת במחלקה `Person`, ולכן ניגשת לשדה `className` באותה מחלקה (שערכו הוא "Person").

סעיף ב' (4 נקודות)

הניחו כי `<a>` מוחלף בקוד הבא:

```
p.printClassName2();
```

מה יקרה כאשר ננסה להריץ את התכנית?

(1) תודפס המחרוזת "Person".

(2) תודפס המחרוזת "Student".

(3) הקוד לא יעבור קומפילציה (*בתשובה לשאלה זו התייחסו רק לקוד בשיטת `printClassName2()` במחלקות `Person` ו-`Student`)

(4) התכנית תיפול בזמן ריצה.

תשובה: (2) – השיטה `printClassName2()` הנמצאת ב-`Student`, דורסת את השיטה במחלקה `Person`, ולכן ניגשת לשדה `className` באותה מחלקה (שערכו הוא "Student").

סעיף ג' (4 נקודות)

הניחו כי <a> מוחלף בקוד הבא:

```
p.printName();
```

מה יקרה כאשר ננסה להריץ את התכנית?

(1) תודפס המחרוזת "Name: Moti".

(2) תודפס המחרוזת "Name: Moti (student)".

(3) הקוד לא יעבור קומפילציה (*התייחסו אך ורק לקוד בשיטות getName() ו-printName() במחלקות Person ו-Student).

(4) התכנית תיפול בזמן ריצה.

תשובה: (3) – השיטה getName() הנמצאת ב- Student, מנסה לגשת לשדה name שהינו שדה private של Person, ולא נמצא ב- Student.

סעיף ד' (4 נקודות)

הניחו כי <a> מוחלף בקוד הבא:

```
p.printId();
```

מה יקרה כאשר ננסה להריץ את התכנית?

(1) תודפס המחרוזת "Id: 1234".

(2) תודפס המחרוזת "Id: 1111".

(3) הקוד לא יעבור קומפילציה (*התייחסו אך ורק לקוד בשיטות getId() ו-printId() במחלקות Person ו-Student).

(4) התכנית תיפול בזמן ריצה.

תשובה: (1) – השיטה getId() הנמצאת ב- Person היא private, ולכן כאשר קוראים ל- getId() מהשיטה printed() מוחזר השדה מהמחלקה A.

סעיף ה' (4 נקודות)

הקיפו בעיגול "נכון" או "לא נכון" על כל אחת מהטענות הבאות:

1. מחלקה יכולה לממש (implement) ממשק אחד בלבד. נכון \ לא נכון

2. מחלקה יכולה לרשת (extend) ממחלקה אחת בלבד. נכון \ לא נכון
3. מחלקה אבסטרקטית לא חייבת להכיל שיטות אבסטרקטיות. נכון \ לא נכון
4. שתי השיטות הבאות, הנמצאות באותה המחלקה, מהוות דוגמה נכונה להעמסה (overloading):

```
int add(int x, int y) {...}
```

```
double add(int x, int y) {...}
```

נכון \ לא נכון

תשובה:

- (1) לא נכון – מחלקה יכולה לממש יותר מממשק אחד.
- (2) נכון – מחלקה יכולה לרשת לכל היותר ממחלקה אחת.
- (3) נכון – ניתן להגדיר מחלקה אבסטרקטית שכל השיטות בה ממומשות.
- (4) לא נכון – יש לשתי השיטות אותה חתימה, לא עובר קומפילציה.