רוטברג
במה אתם רוצים להצליח?

מבחן מחצית הנדסת תוכנה סייבר בינה

5 יחידות לימוד

סמל שאלון 899381

הוראות לנבחן

שם התלמיד/ה:			

פרק ראשון – יסודות בתמ"ע
בפרק זה שלוש שאלות, ומהן יש לענות על שתיים. (10X1) + (15X1) ---- 25 נקודות
על פי ההוראות בפרק זה.

פרק שני – מבנה נתונים
בפרק זה שלוש שאלות , ומהן יש לענות על שתיים. (2X 25) ------ 50 נקודות

פרק שלישי – תכנות מונחה עצמים /מודלים
בפרק זה 2 שאלות יש לענות על שאלה אחת בלבד(25 1) ----- 25 נקודות

סה"כ - - -- 100 נקודות

חומר עזר: כל חומר עזר **כתוב** , מחשבון מדעי, אין להשתמש במחשב הניתן לתכנות ובאמצעים אלקטרונים/סלולריים כלשהם.

זמן הבחינה: התחלה 8:30 . סיום 11:30 הארכת זמן 12:15 .

הוראות מיוחדות:

1. יש להגיש בנפרד את חלקים 1+3 ואת חלק 2 . לחלק 1+3 יש להוסיף את עמוד ההוראות (עמוד זה)
2. עני על כל שאלה בדף נפרד – יש לרשום בברור את מספר השאלה ומספר הסעיף עליו עונים .
3. סמני בבירור את מספר השאלה.
4. כתבי בעט כחול או שחור(לא מחיק) אין לכתוב בעיפרון .
5. אין לכתוב תשובות על גבי השאלון (לא יבדק!)
6. ההוראות במבחן זה רשומות בלשון נקבה ומיועדות לנבחנים ונבחנות כאחת.

בהצלחה !

סמטת במעלה, רמת השרון 47100 | טל. 03-5495063 | 03-5494977
rothberg100.com | office@rotberg100.com | פקס. 03-5493582

חלק א' – 50 נקודות

שאלה מספר 1 – חובה (20 נקודות)

עבור מטבע **Coin** נשמרים ערכו value (מספר שלם) ומחרוזת head ובה שם הדמות המוטבעת על ראש המטבע. ותכונה בוליאנית isOn שערכה אמת אם המטבע מונח על head ושקר אם לא. .

א – כתבו כותרת ותכונות למחלקה Coin – 5 נקודות.

ב- הוסיפו פעולה במחלקה בשם sum המקבלת מטבע c מסוג Coin ומחזירה את סכום value של שני המטבעות **אם לשניהם יש אותה דמות בתכונה head** אם לא מוחזרת מכפלת ערכי value של המטבעות. – 6 נקודות.

ג- כתבו פעולה חיצונית sumC המקבלת מערך מטבעות C מסוג Coin ומחזירה מחרוזת ובה שם הדמות המוטבעת על המטבע שיש לו את הערך הגבוה ביותר (ניתן להניח שיש מטבע אחד כזה). 9 נקודות.

בחרי שאלה אחת מבין השאלות 2-3 (30 נקודות)

שאלה 2

לפניכם המחלקה הבאה:

```
public class MyArr {
private int []arr;
private  int n;
private boolean aboveN;  //  n התכונה תהיה אמת אם כל המספרים במערך גדולים מהתכונה
```

א- כתבי בנאי במחלקה המקבל גודל מערך size מספר (שלם), n מספר שלם הבנאי בונה מערך ריק בגודל size מעדכן את n בערך שהתקבל וקובע את abveNלהיות false – 5 נקודות.

ב- כתבי פעולה במחלקה בשם isOK המקבלת מספר שלם n ומעדכנת את n להיות בערך שהתקבל את aboveN להיות true אם כל המספרים במערך חיוביים וגדולים מ n ושקר אם לא. 10 נקודות.

ג- כתבו פעולה במחלקה בשם : **ninInArr** המחזירה אינדקס התא של הערך הקטן ביותר במערך. 15 נקודות

דוגמה עבור המערך arr הבא יחזור 2 (מספר התא שיש בו את הערך הקטן ביותר)

0	1	2	3	4	5	6
15	80	-65	65	4-	80	15

שאלה 3

לפניכם מחלקה המתארת טלפון סלולרי:

```
public class Mobile
{     private String model;          // דגם המכשיר
      private int price;             // מחיר לצרכן
     private double memosize;     // גודל זכרון בג'יגה בייטים

public void setMsize (double a)
{   if(a>0)
     this.memosize+=a;
}

public boolean compare(Mobile m)
{   if(this.model.equals(m.model) && this.price<m.price)
       return true;
  else
      return false;
 }
}
```

לרשותכם פעולה בונה סטנדרטית ופעולות set/get

א – כתבו פעולה חיצונית שמקבלת מערך של טלפונים ומחזירה את דגם המכשיר הכי יקר – 10 נקודות. .

ג – לפניכם קטע קוד. עקבו אחריו וציינו מה יהיה הפלט: 14 נקודות.

```
Mobile m1=new Mobile ("nexus5" , 1600,30);
Mobile m2 = new Mobile(m1.getModel() , m1.getPrice()/2 , 25);
m1.setMsize(4);
System.out.println( m2.compare(m1));
```

ד – שנו שורה אחת בקוד, כך שיתקבל פלט הפוך . נמקו בחירתכם 6 נקודות.

חלק ב' – מבנה נתונים – 100 נקודות – יש לפתור חלק זה על גבי דפים נפרדים

בחלק זה 3 שאלות – יש לענות על שתי שאלות בלבד מתוך השאלות 4-5-6 . (2*50)

שאלה 4 – (50 נקודות)

נתונה הפעולה sod.

טענת כניסה:
הפעולה מקבלת רשימת שלמים

```
public static void sod(Node<Integer> n)
{
  if (n == null || !n.hasNext())
      return ;
  Node<Integer> p=n.getNext();
   n.setNext(p.getNext());
   p.setNext(null);
   sod(n.getNext());
}
```

א בצעו מעקב אחרי זימון הפעולה sod (n) עבור הרשימה: **(20 נק)**:

n→ [1] → [2] → [3] → [4] → null

ב מה טענת היציאה של הפעולה sod**(5 נק)**

ג מה סיבוכיות הפעולה? הסבירו **(5 נק)**

ד שינו את הפעולה כך:
- הוסיפו פרמטר k לכותרת,
- שינו את תנאי העצירה
- שינו את הזימון הרקורסיבי של sod.

```
public static void sod(Node<Integer> n, int k)
{
  if (k >= 3 || n == null || !n.hasNext())
      return ;
  Node<Integer> p=n.getNext();
   n.setNext(p.getNext());
   p.setNext(null);
   sod(n.getNext(), k + 1);
}
```

טענת הכניסה: הפעולה מקבלת רשימת שלמים ומספר שלם k > 0

I. האם השתנתה טענת היציאה של sod? אם כן מהי? אם לא הסבירו מדוע **(4 נק)**

II. האם השתנתה הסיבוכיות של sod? אם כן מה הסיבוכיות החדשה? הסבירו את תשובתכם **(4 נק)**

III. מה הסיבוכיות של הפעולות הבאות? הסבר את תשובתך **(12 נק)**

```
public static void secret1(Node<Integer> n, int m)
{
   for (int i = 0; i < m; i++)
      sod(n);
}
```

```
public static void secret3(Node<Integer> n)
{
   for (int i = 0; i < 1000; i++)
      sod(n);
}
```

```
public static void secret3(Node<Integer> n, int m)
{
   for (int i = 0; i < Math.min(1000, m); i++)
      sod(n);
}
```

. **שאלה 5 – 50 נקודות**

נתונה הפעולה **size** - public static int size(Node<Integer> lst)
הפעולה מקבלת כקלט רשימה של שלמים ומחזירה את אורך הרשימה.

א כתבו פעולה בשם **retrieve** אשר מקבלת כקלט רשימה של שלמים ומספר שלם חיובי n. הפעולה מחזירה את הערך השמור ברשימה במקום $\frac{1}{n}$ (למשל אם $n = 2$ אז יוחזר הערך שבאמצע הרשימה ואם $n = 3$ אז יוחזר הערך שנמצא בשליש הרשימה). יש להשתמש בפעולה **size**. **(20 נק)**

לדוגמא, עבור n=2 והרשימה: Lst→ [3] → [5] → [-7] → [4] יוחזר המספר 5

* ניתן להניח שהרשימה מתחלקת ב n ללא שארית

ב כתבו פעולה **spaces** אשר מקבלת רשימה של שלמים ומספר שלם חיובי n. על הפעולה spaces לרוץ בסיבוכיות הטובה ביותר שאפשר. הפעולה מחזירה תור אשר מכיל את האיברים של הרשימה אשר נמצאים במקומות $\frac{1}{n}, \frac{2}{n}, \ldots, \frac{n}{n}$. **(25 נק)**

לדוגמא, עבור n=2 והרשימה: Lst→ [3] → [5] → [-7] → [4]

יוחזר התור: head [5 | 4]

אין חובה להשתמש בפעולה retrieve מהסעיף הקודם

* ניתן להניח שהרשימה מתחלקת ב n ללא שארית

ג הסבירו מה הסיבוכיות של סעיף (ב). **(5נק)**

שאלה 6 (50נק)

א כתבו פעולה רקורסיבית בשם **zeroes** אשר מקבלת מספר שלם ומחזירה את מספר הספרות 0 במספר (אם אין במספר אף ספרה שהיא 0, הפעולה תחזיר 0).
(20 נק)

ב כתבו פעולה רקורסיבית בשם **maxZeros** אשר מקבלת כקלט שתי רשימות מטיפוס Integer (מותר להניח ששתי הרשימות באותו אורך). הפעולה תחזיר את מספר האפסים הגבוה ביותר שמופיע באיבר כלשהו בשתי הרשימות. יש להשתמש בסעיף הקודם.

לדוגמא, עבור הרשימות

lst1→ [10] → [100] →~ lst2 → [100000] → [12] →~

יוחזר 5 כי ב 100000 יש 5 אפסים **(25 נק)**

ג הסבירו מה הסיבוכיות של סעיף (ב). **(5 נק)**

חלק ג' – תכנות מונחה עצמים – 50 נקודות

בחלק זה שתי שאלות יש לענות על שאלה אחת מתוך השאלות 7-8

שאלה 7 נתון הפרויקט הבא – שימו לב, סעיפי השאלה נמצאים בעמוד הבא.

```
public class One
{ protected String str;
   protected int x;
   protected static int count=100;
   public One(String st)
   { this.x=count;
      this.str=st;
      One.count+=st.length();
   }
     public One(int x)
   {   this.x=x;
      this.str=this.count+"!!"+this.x;
      One.count++;
   }
     public One(One other)
   {     // סעיף ב'
   }
   public int sumOne()
   {return this.x+this.str.length();  }
   public void printM()
   {
      System.out.println("x:"+this.x+
"str:"+this.str+"-"+this.count);
   }
```

```
public class Two extends One
{   private int num;
   public Two()
   { super("Queen");
      this.num=2;
   }
    public Two(int n)
   {super(n);
      this.num=n*2;
   }
      public Two(String st)
   { super(st);
      this.num=st.length();
   }
   public int sumTwo(int x)
   {return this.sumOne()+x; }
   public int sumTwo()
   {return
this.sumOne()+this.num; }
   public void printM()
   {System.out.println("sum="+
this.sumTwo());
   }
}
```

```
public class Three
{ private One one;
   private int xx;

   public Three(int xx)
   {this.one=new One(xx);
    this.xx=xx;
   }
    public Three(Three other)
   {
    //   סעיף ב'
   }
   public int sumThree()
   {
      return this.one.sumOne()+this.xx;
   }
   public void printM()
   {
      System.out.println("Three"+
this.sumThree());
   }
}
```

המשך שאלה 7 – השאלות בחלק זה מתייחסות לפרויקט בעמוד הקודם.

א- שרטטי UML המתאר את הפרוייקט בשאלה זו. – 6 נקודות.

ב- השלימי את הבנאי המעתיק במחלקה One ובמחלקה Three – 12 נקודות.

ג- נתון קטע התכנית הבא – בצעי מעקב עצמים ורשמי בברור מה יהיה פלט התכנית - 20 נקודות.

```
One o1=new One(5);
 Two t1=new Two("MyTest");
  Three t2=new Three(25);
  One o2=new One("Good");

  o1.printM();
  t1.printM();
  t2.printM();
  o2.printM();
```

ד. נתונים ההגדים הבאים – עבור כל אחד מהם רשמי נכון /לא נכון – יש ללוות את התשובה בהסבר והדגמה מתוך הפרוייקט בשאלה זו. – 12 נקודות

			הסבר/הדגמה בעזרת הפרוייקט
א	ניתן לגשת ישירות לתכונות המחלקה One מתוך המחלקה Three	נכון/לא נכון	
ב	בפרויקט זה מתקיים עקרון הדריסה	נכון/לא נכון	
ג	בפרויקט זה מתקיים עקרון ההעמסה	נכון/לא נכון	

שאלה 8

במאפיית האופים רצו למחשב את מערכת המידע שלהם לצורך זה איפיינו את המחלקות הבאות:

המחלקה מאפה **Pastry** – תכונות המחלקה שם המאפה name (מחרוזת) , מחיר לק"ג מאפה price (ממשי) , האם נטול גלוטן isFree (בוליאני - אמת אם נטול גלוטן ושקר אם לא) .

המחלקה קרואסון – **Croissant** – סוג של מאפה מוסיפה את התכונות – מילוי filling (מחרוזת)

מתוק isSweet (בוליאני אמת אם מתוק ושקר אם לא) .

המחלקה בורקיטס **Burkitts** (סוג של מאפה) – מוסיפה את התכונות הבאות – האם חלבי isDairy (בוליאני – אמת אם כן ושקר אם לא)

מחירי המאפים נקבעים בדרך הבאה :

עבור מאפה רגיל (לא קוראסון או בורקיטס) – משקל * מחיר לק"ג. אם נטול גלוטן 20% יותר.

עבור קרואסון – משקל*מחיר לק"ג אם נטול גלוטן 20% יותר אם יש מילוי שוקולד "chocolate " או חמאה " butter" תוספת של 10% נוספים.

עבור בורקיטס – משקל*מחיר לק"ג אם חלבי תוספת של 30% .

המחלקה הזמנה **Order** – מתעדת הזמנה של לקוח תכונות המחלקה :

מספר הזמנה num (מספר שלם אוטומטי הנוצר בבנאי המחלקה)

מונה הזמנות count - מספר שלם– מאותחל באפס – תכונת מחלקה

האם יש מבצע השבוע discount – בוליאני – תכונה מחלקתית מאותחל ב false.

מאפה p מסוג Pastry, מאפה קרואסון c מסוג 'Croissant , מאפה בורקיטס b מסוג Burkitts

חישוב מחיר סופי של הזמנה יקבל את משקלי מאפה, קוראסון ובורקיטס ויחשב את המחירים לפי הנחיות החישוב במחלקות השונות – אם יש הנחה יורדו 10% מהמחיר הסופי.

א- ציירי תרשים uml המתאר את הקשר בין המחלקות – 6 נקודות.
ב- כתבי כותרת ותכונות למחלקה Order – 6 נקודות.
ג- כתבי **בנאי** למחלקה Order הפעולה מקבלת מאפה p מסוג Pastry, מאפה קרואסון c מסוג 'Croissant , מאפה בורקיטס b מסוג Burkitts מעדכנת את הערכים שהתקבלו, יוצרת מספר הזמנה ומקדמת את מונה ההזמנות. 10 נקודות.
ד- ממשי את הפעולה price במחלקות הממשות. – 12 נקודות.
ה- כתבי פעולה חיצונית בשם totalSum המנהלת את ההזמנות עבור יום עבודה במאפייה, הפעולה **מקבלת** מערך הזמנות orders **וקולטת** עבור כל הזמנה משקלי מאפה, קרואסון ובורקיטס. הפעולה מדפיסה עבור כל הזמנה את מספר ההזמנה ואת המחיר הסופי של ההזמנה. בנוסף, הפעולה תחזיר את סכום המחירים של כל ההזמנות שבוצעו באותו יום.
– 16 נקודות.

- , ניתן להניח שנכתבו בנאים (כולל בנאי מעתיק), פעולות שרות get/set עבור כל התכונות במחלקות השונות. פעולות עזר נוספות – ניתן לכתוב – יש לרשום בברור באיזה מחלקה נכתבו.