

סמסטר ב' תשע"ב  
מועד: א' 02/07/2012  
שעה: 9:00  
משך הבחינה: 3½ שעות  
חומר עזר: כל חומר עזר כתוב מותר

מספר זהות:

--	--	--	--	--	--	--	--	--	--

**בחינה בקורס: תכנות מכוון עצמים ושפת C++**

מרצה: ד"ר אלון שקלר

מתרגל: אמיר קירש

**מדבקות  
ברקוד**

**הנחיות כלליות לבחינה:**

- שימו לב, השאלון מודפס משני צדדיו.
- המבחן מורכב משני חלקים:
  - חלק א' כולל 7 שאלות אמריקאיות. משקל כל שאלה 7 נקודות, סה"כ: 49 נק'.
  - חלק ב' כולל שאלת תכנות שמשקלה הכולל 51 נק'.
- חובה לתעד בשאלת התכנות כל פעולה לא ברורה שנעשית.
- בשאלות האמריקאיות יש לסמן תשובה אחת לכל שאלה בטבלה המצורפת. במידה ומספר תשובות נראות נכונות יש לסמן את התשובה הנכונה ביותר.
- בסיום המבחן יש לרשום מס' ת.ז. על גבי טופס המבחן, לוודא שטבלת התשובות האמריקאיות נמצאת יחד עם טופס המבחן, ולהגישם בתוך מחברת הבחינה.
- המבחן הינו עם חומר פתוח. כל חומר עזר מותר למעט מכשירים אלקטרוניים למיניהם. אין להעביר חומר עזר בין תלמידים במהלך המבחן.
- נא לכתוב בכתב קריא ולא מחובר.

**בהצלחה !**



## טבלת תשובות לחלק האמריקאי

ו'	ה'	ז'	ג'	ב'	א'	
						<u>שאלה 1</u>
						<u>שאלה 2</u>
						<u>שאלה 3</u>
						<u>שאלה 4</u>
						<u>שאלה 5</u>
						<u>שאלה 6</u>
						<u>שאלה 7</u>

### הסברים לתשובות

**חובה** לספק הסבר במידה ונבחרה תשובה ו' (אף תשובה אינה נכונה).  
 מותר לצרף הסבר גם עבור תשובות אחרות. אמנם **רק** תשובה נכונה תזכה בניקוד עבור כל שאלה, אולם ניתן יהיה להסתמך על ההסבר במסגרת ערעור, אם יידרש. מומלץ לצרף הסבר לתשובה אמריקאית במיוחד במקרים בהם נראה לך שתשובתך דורשת הסבר או נימוק.

שאלה 1 :

---



---

שאלה 2 :

---



---

שאלה 3 :

---



---

שאלה 4 :

---



---

שאלה 5 :

---



---

שאלה 6 :

---



---

שאלה 7 :

---



---

חלק א' – שאלות אמריקאיות (49 נק' – 7 נק' לכל שאלה)

שאלות 1-7 מתייחסות לקטע הקוד הבא:

```

1.  class MyString
2.  {
3.      class MyChar
4.      {
5.          MyString* pStr;
6.          int index;
7.      public:
8.          MyChar(MyString* p, int i): pStr(p), index(i) {}
9.          const MyChar& operator=(char c) {
10.             pStr->set(index, c); // call copy-on-write
11.             return *this;
12.         }
13.         operator char()const{return pStr->str[index];}
14.     };
15.
16.     friend class MyChar;
17.     char* str;
18.     int* ref_counter;
19.     void attach(const MyString& s) {
20.         str = s.str;
21.         ref_counter = s.ref_counter;
22.         ++(*ref_counter);
23.     }
24.     void detach() {
25.         if(--(*ref_counter) == 0) {
26.             delete []str;
27.             delete ref_counter;
28.         }
29.     }
30.     void set(int index, char c) {
31.         // this is the copy-on-write
32.         MyString temp(str);
33.         detach();
34.         attach(temp);
35.         str[index] = c;
36.     }
37. public:
38.     MyString(char* s=""): ref_counter(new int(1))
39.     {
40.         str = new char[strlen(s) + 1];
41.         strcpy(str, s);
42.     }
43.     MyString(const MyString& s){attach(s);}

```

```

44. ~MyString(){detach();}
45. const MyString& operator=(const MyString& s)
46.     {detach(); attach(s); return *this;}
47. operator const char*()const{return str;}
48. MyChar operator[](int index) {
49.     return MyChar(this, index);
50. }
51. char operator[](int index)const {
52.     return str[index];
53. }
54. friend ostream& operator<<
55.     (ostream& out, const MyString& s) {
56.         return out << s.str << ", ref count = "
57.             << *s.ref_counter << endl;
58.     }
59. };
60.
61. int main()
62. {
63.     MyString s1;
64.     cout << s1;
65.     MyString s2=s1, s3=s1;
66.     cout << s3;
67.     s1 = "Hello";
68.     cout << s3;
69.     s3 = s2 = s1;
70.     cout << s3;
71.     cout << "s1[0] = " << s1[0] << endl;
72.     s2[4] = '!';
73.     cout << s3 << s2;
74.     return 1;
75. }

```

**נתון שהתוכנית לעיל עוברת קומפילציה בהצלחה ורצה במלואה בלי להתרסק!**

השאלות על קטע הקוד מתחילות בעמוד הבא.

### שאלה 1

מה תדפיס שורה 64 ב-main?

א. `null, ref count = 1`

ב. `, ref count = 1`

ג. `null, ref count = 0`

ד. `s1, ref count = 0`

ה. שורה 64 לא תדפיס כלום.

ו. אף תשובה אינה נכונה.

### שאלה 2

מה תדפיס שורה 66 ב-main?

א. `null, ref count = 1`

ב. `null, ref count = 2`

ג. `, ref count = 2`

ד. `, ref count = 3`

ה. `s3, ref count = 1`

ו. אף תשובה אינה נכונה.

### שאלה 3

מה היה מודפס בשורה 66 במידה והיינו מוסיפים את המילה `explicit` ל-`c'tor` בשורה 43?

א. לא היה שינוי בהדפסה של שורה 66.

ב. לא ניתן להוסיף `explicit` ל-`c'tor`, לכן שורה 43 לא תעבור קומפילציה.

ג. שורה 65 לא תעבור קומפילציה.

ד. יודפס `ref count` נמוך באחד ממה שהודפס קודם, ללא שינוי נוסף.

ה. יודפס `ref count` גבוה באחד ממה שהודפס קודם, ללא שינוי נוסף.

ו. אף תשובה אינה נכונה.

#### שאלה 4

מה תדפיס שורה 68 ב-main?

א. null, ref count = 1

ב. null, ref count = 2

ג. , ref count = 2

ד. , ref count = 3

ה. s3, ref count = 1

ו. אף תשובה אינה נכונה.

#### שאלה 5

מה תדפיס שורה 70 ב-main?

א. Hello, ref count = 1

ב. Hello, ref count = 2

ג. Hello, ref count = 3

ד. s3, ref count = 1

ה. s3, ref count = 2

ו. s3, ref count = 3

ז. אף תשובה אינה נכונה.

#### שאלה 6

מה ידפיסו שורות 71 עד 74 ב-main? (מימין לשמאל, <= מסמן ירידת שורה).

א. Hell!, ref count = 2 <= Hello, ref count = 2 <= s1[0] = H

ב. Hell!, ref count = 2 <= Hello, ref count = 1 <= s1[0] = H

ג. Hell!, ref count = 1 <= Hello, ref count = 3 <= s1[0] = H

ד. Hell!, ref count = 1 <= Hello, ref count = 2 <= s1[0] = H

ה. Hell!, ref count = 3 <= Hello, ref count = 2 <= s1[0] = H

ו. אף תשובה אינה נכונה.

## שאלה 7

מורן שירן ולירן התכוננו למבחן ועברו על הקוד לעיל.

לירן טען: ב-main שמופיע במבחן אין בכלל שימוש באופרטור שבשורה 51 מכיון שהוא מוגדר

כ-const ולא מוגדר אף אובייקט MyString כ-const ב-main.

שירן טען: האופרטור שבשורה 51 מצליח להתקמפל ולחיות בשלום ביחד עם זה שבשורה 48

מכיון שיש להם **ערך מוחזר** שונה.

מורן טען: האופרטור שבשורה 51 מצליח להתקמפל ולחיות בשלום ביחד עם זה שבשורה 48

מכיון שאחד מהם מוגדר כפונקציית const והשני אינו מוגדר כ-const.

מי מהשלושה צודק ?

א. לירן (הוא גם מצטיין דיקאן).

**ב. לירן ומורן.**

ג. לירן ושירן.

ד. שלושתם צודקים.

ה. שלושתם טועים.

ו. אף אחת מהתשובות אינה נכונה.

## חלק ב' – שאלת תכנות (51 נק')

חברת BetterParking מפתחת ומשווקת תוכנה לניהול חניונים. התוכנה מטפלת בלקוחות פרטיים (לקוחות מזדמנים) וכן בלקוחות מוסדיים (חברות וארגונים) שחותמים על הסכמי חניה מול החניון לטובת עובדיהם. לקוח מזדמן שמגיע לחניון בוחר מאפייני חניה ומופנה למקום חניה מתאים לפי צרכיו. כאשר הלקוח חוזר לקחת את רכבו המערכת מפנה אותו לפי מספר הרכב למקום בו הרכב נמצא ומחשבת את עלות החניה ללקוח.

מאפייני החניה הינם שירותים / אילוצי חניה שלקוח יכול לבקש, לדוגמה :

- שירותים : רחיצה, טעינה לרכב חשמלי, קישוט רכב חתן-כלה וכו'.
- אילוצים : חניה מוצלת, חניה לרכב מונע בגז, חניה לרכב גבוה וכו'.

לקוח יכול לבקש לכל היותר שירות אחד וכן מספר כלשהו של אילוצים. במידה ולא נותרו מקומות פנויים בחניון המאפשרים את הקומבינציה של השירות שדרש הלקוח ביחד עם כל האילוצים שנדרשו, על המערכת לדאוג להחזיר ערך שיצביע על כך שבקשת הלקוח לא יכולה להתקיים. ממשק המשתמש יודיע ללקוח ויבקש בחירה חדשה אך אין זה באחריותך.

במבחן זה עליך לטפל בלקוחות מוסדיים. לקוח מוסדי חותם על הסכם עם החניון לחניה של עובדיו ומספק לחניון רשימה של כל מספרי הרכב המהווים את מנויו בחניון. החניון לא מאפשר למנויים המוסדיים לקבוע אילוצים, מאידך, החניון עצמו עדיין מעוניין להחזיק את האילוצים המתקיימים בכל אזור חניה לצורך הלקוחות הפרטיים.

שים לב : במבחן זה הינך חלק מצוות המנוע ואסור לך לכתוב ממשק משתמש.

ישנו מיגוון של הסכמים עם מנויים מוסדיים, הסכמים אלו נבדלים במספר תכונות :

### [א] אזור חניה פרטי – כן/לא

1. לקוחות עם אזור חניה פרטי – כל עוד ישנם מקומות באזור הפרטי של לקוח זה, ניתן לחנות, אחרת, אם נגמרו המקומות באזור הפרטי לא תתאפשר כניסת מנויים של לקוח זה. האזור הפרטי של הלקוח מוגבל לחניית מנויים של לקוח זה בלבד!
2. לקוחות עם חניה חופשית בכל אזורי החניון מנויים של לקוח זה יכולים לחנות בכל אזורי החניון (פרט לאזורים הפרטיים של לקוחות אחרים – כפי שתואר בסעיף הקודם). בהסכם זה ניתן להגביל את כמות מנויי הלקוח החונים בו זמנית בחניון אך ניתן גם שלא להגביל כלל (ואז המגבלה היא כל עוד יש מקום בחניון באזורים שאינם פרטיים).

### [ב] שירותים אפשריים

1. ישנם לקוחות שלא כוללים קבלת שירותים בהסכם ואז לקוחותיהם לא יקבלו שירותים.
2. ישנם לקוחות שמאפשרים למנויים שלהם לקבל X פעמים בחודש שירות כלשהו (המספר X יכול להיות שונה בין שירות לשירות ואף יכול להיות אפס עבור חלק מהשירותים). כאשר מנוי צרך כבר שירות מסוים בכמות המותרת לשירות זה, החניון לא יאפשר למנוי לצרוך את השירות בחודש זה.

### [ג] אופן החיוב עבור החניה עצמה (לא כולל עלות השירותים)

1. חיוב חודשי מוסכם מראש – בשיטה זו, הלקוח המוסדי משלם לחניון סכום קבוע מידי חודש עבור החניה של כל מנויו, בלי תלות בניצול בפועל.



2. חוב בהתאם לניצול החניה בחישוב יומי – בשיטה זו הלקוח המוסדי משלם על כל יום חניה בפועל של כל אחד ממנוייו סכום מוסכם מראש (אם מנוי יצא ונכנס שוב באותו יום הוא ייספר רק פעם אחת עבור אותו יום).
3. חוב בהתאם לניצול החניה לפי דקות – בשיטה זו הלקוח המוסדי משלם סכום מוסכם לדקה, מוכפל בסך כל הדקות שחננו כל מנויו במהלך התקופה.

#### [ד] תשלום בגין קבלת השירותים

1. חלק מהלקוחות חותמים על תשלום חודשי קבוע ללא קשר לצריכה בפועל.
2. חלק מהלקוחות חותמים על תשלום לפי צריכה בפועל, במחיר מוסכם מראש של P עבור כל שירות (כאשר P יכול להיות שונה משירות לשירות).
3. לקוחות שלא מאפשרים צריכת שירותים כלל – אין להם כמובן הסכם תשלום על צריכת שירותים.

#### הגעה ויציאה מהחניון :

כאשר רכב נכנס לחניון ממשק המשתמש מעביר אלייך כפרמטרים את: מספר הרכב + שירות אחד או אפס שבו הוא מעוניין + רשימת אילוצים (שחייבת להיות ריקה במקרה של מנוי). המערכת אמורה לזהות האם זהו לקוח מזדמן או מנוי ובסופו של דבר יש להחזיר כערך מוחזר תשובה לשאלה האם ניתן לחנות או לא, ואם כן איפה (כערך מוחזר אחד). אם לא ניתן לחנות יש להחזיר את הסיבה כמחרוזת, כחלק מהערך המוחזר או כ-Exception (למשל: "אינך רשאי לצרוך שירותים" או "אינך רשאי לצרוך שירות זה" או "ניצלת את מספר הפעמים המותר של השירות המבוקש" או "אין מקום בחניון" או "אין מקום באזור החניה המיועד" או "לא נותרו חניות עבור מנויי הלקוח" או "אינך רשאי להציג אילוצי חניה"). ביציאה מועבר אלייך מספר רכב יוצא בלבד. הנח שרכב לא יכול להיות רשום כמנוי ע"י מספר לקוחות מוסדיים בו-זמנית.

#### הנח כי קיימות הפונקציות הגלובליות הבאות :

1. `long getTime()` – מחזירה ערך מספרי כלשהו שמייצג את הזמן הנוכחי.
2. `int getMinutesDiff(long from, long to)` – מחזירה את משך הזמן בדקות שעבר בין שני ערכים שחזרו מהפונקציה `getTime`.
3. `string getDateTime(long timeValue)` – מחזירה ייצוג במחרוזת של התאריך והשעה, בהתאם לערך שחזר מהפונקציה `getTime`.
4. `int getDayInMonth(long timeValue)` – מחזירה את היום בחודש (מספר בין 1 ל-31), בהתאם לערך שחזר מהפונקציה `getTime`.

ההתחשבויות מתבצעות ע"י הפעלה ידנית של פונקציית התחשבויות מייד בסוף החודש. פונקציה זו צריכה להפיק קבצי טקסט של החשבוניות, קובץ לכל לקוח בפורמט שיוסבר בסעיפי השאלה, וכן לאפס את כל הנתונים לצורך פתיחת חודש חדש. עליך לכתוב פונקציה זו בסעיף ג'. באחריות מפעיל החניון לקרוא לפונקציה (במידה והפונקציה לא נקראה, ההנחה היא שהחודש עדיין לא נגמר).

#### שים לב :

יש לענות על סעיפי השאלה באופן מסודר לפי הסעיפים בעמוד הבא.

## סעיף א' (21 נקודות)

כתוב את הגדרת המחלקות הנדרשות לפתרון (כל ה-`prototypes`, ללא מימושים).  
עליך להציג את המחלקות במלואן, גם עבור נושאים שאינך נדרש לממש (ללא ממשק משתמש).  
(למשל: על החניון להציג שדות ופונקציות לטיפול באילוצים, למרות שלא נדרש לגבי מנויים).  
הקפד על שימוש נכון ב-`public`, `protected`, `private` ו-`const`.

הצעת עזר לפתרון (מומלץ אך לא חובה):

חניון מאופיין באופן הבא:

- לחניון ישנם מספר אזורים בעלי מאפיינים שונים.
- כל אזור בחניון מתאפיין באוסף המאפיינים הספציפיים שלו, כלומר – לאיזה אילוצים הוא עונה ואילו שירותים הוא יכול לתת: האם מותר להכניס לאזור זה רכבים מונעי גז, האם האזור מוצל או פתוח לשמש, האם זהו אזור רחיצת מכוניות וכו'.
- אזור חניון שמשורין כאזור פרטי למנויי לקוח מוסדי מסויים כדאי שיסומן ככזה.
- רצוי שניתן יהיה במהירות ובקלות להגיע:
  - ממספר רכב למיקומו בחניון בלי צורך לסרוק את כל החניון.
  - ממספר רכב למנוי שאליו הוא משוייך (או אם איננו מנוי, לזהות שאיננו מנוי).
- עבור כל לקוח מוסדי יש להחזיק את כל רכיבי ההסכם עם החניון, שיטה אפשרית הינה להחזיק שדה עבור כל תכונה אפשרית, כפוינטר למחלקת בסיס ובפועל להכניס לכל שדה כזה את המימוש הרלבנטי כפוינטר לאובייקט של מחלקה נגזרת – בהתאם לסעיף בפועל כפי שנחתם עם הלקוח. אפשר לעטוף את כל שדות ההסכם במחלקה לתיאור הסכם.

## סעיף ב' (15 נקודות)

ממש באופן מלא את כל הפונקציות הנדרשות לקבלת רכב של מנוי בחניון (מנוי הינו, כפי שהוסבר, עובד של לקוח מוסדי שמספר הרכב שלו נמסר לחניון כמנוי). יש לטפל בהשמטתו של המנוי לחניה אם ישנו מקום, כולל צריכת השירות שביקש המנוי, אם ביקש. במידה ולא ניתן להיענות לבקשת החניה יש להחזיר את הסיבה כמחרוזת, כחלק מהערך המוחזר או כ-`Exception`, כפי שתואר.

שים לב כי כאשר הרכב מגיע לחניון לא ידוע שזהו רכב של מנוי (אם מדובר בלקוח מזדמן אפשר לשלוח את הרכב לטיפול בפונקציית עזר לטיפול בלקוחות מזדמנים ואין צורך לממשה).

## סעיף ג' (15 נקודות)

ממש באופן מלא את חישוב החשבון החודשי ללקוחות מוסדיים, כולל כתיבה לקובץ טקסט של פרטי החשבונות – קובץ טקסט נפרד לכל לקוח מוסדי. כל קובץ טקסט חייב להכיל בשורה הראשונה או האחרונה את הסכום לתשלום כשורה בפני עצמה, בפורמט: `SUM = <sum>`  
ובשאר השורות את פירוט הנתונים הרלבנטיים שיאפשרו ללקוח לבדוק את החשבון, בפורמט טקסט שתמצא לנכון.

סוף !