

סמסטר א' תשס"ד
מועד: ב' 5/3/2004
משך הבחינה: 3½ שעות
חומר עזר: כל חומר עזר מותר

בחינה בקורס: תכנות מכוון עצמים ושפת ++C

מרצים: אמיר קירש, איריס רוזנבלום-גבר

הנחיות כלליות לבחינה:

- 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. אם נבחרה תשובה ו' (אף תשובה אינה נכונה), חובה לספק הסבר במקום המיועד לכך.
- בסיום המבחן יש לרשום מס' ת.ז. במקום המיועד לכך בטופס התשובות, לוודא שטופס התשובות נמצא יחד עם טופס המבחן ולא ניתק ממנו, ולהגישם בתוך מחברת הבחינה.
- המבחן הינו עם חומר פתוח. כל חומר עזר מותר למעט מחשבים ניידים. אין להעביר חומר עזר בין תלמידים במהלך המבחן.
- נא לכתוב בכתב קריא ולא מחובר.

בהצלחה !

טבלת תשובות לחלק האמריקאי

מס' ת.ז. : _____

חובה לספק
הסבר עבור
תשובה ו'

א	ב	ג	ד	ה	ו	
						<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.  template<class T>
2.  class A
3.  {
4.      static T* pMyStaticT;
5.      static int counter;
6.  public:
7.      A(const T& t=T())
8.      {
9.          if(pMyStaticT == 0)
10.             pMyStaticT = new T(t);
11.             counter++;
12.     }
13.     ~A()
14.     {
15.         if(--counter == 0)
16.         {
17.             delete pMyStaticT;
18.             pMyStaticT = 0;
19.         }
20.     }
21.     T* operator->(){return pMyStaticT;}
22.     virtual void Print(ostream& out=cout) const
23.     {
24.         out<<"I'm A"<<endl;
25.         out<<"I hold "<<typeid(*pMyStaticT).name()<<endl;
26.         out<<"And its inner value is "<<*pMyStaticT<<endl;
27.     }
28.     friend ostream& operator<<(ostream& out, const A<T>& a)
29.     {
30.         a.Print(out);
31.         return out;
32.     }
33. };
34.
35. template<class T> T* A<T>::pMyStaticT;
36. template<class T> int A<T>::counter;
37.
38. class B
39. {
40.     int i;
```

```

41. public:
42.     B(int i1=1):i(i1){}
43.     void Print()const{cout<<"i="<<i<<endl;}
44.     void Increase(){++i;}
45.     friend ostream& operator<<(ostream& out, const B& b)
46.     {
47.         return out<<"\nI'm B with int i="<<b.i;
48.     }
49. };
50.
51. void f1()
52. {
53.     A<int> aInt1(10);
54.     A<int> aInt2(20);
55.     cout<<aInt2;
56. }
57.
58. void f2()
59. {
60.     A<B> b1(3);
61.     b1->Print();
62.     A<B> b2;
63.     b2->Increase();
64.     b2->Print();
65.     b1->Print();
66. }
67.
68. void f3()
69. {
70.     A<B>* pb1 = new A<B>;
71.     (*pb1)->Increase();
72.     pb1->Print();
73.     delete pb1;
74.     A<B> b2;
75.     b2->Print();
76. }
77.
78. void main()
79. {
80.     f1();
81.     f2();
82.     f3();
83. }

```

השאלות עצמן מתחילות בעמוד הבא.

להלן שורות פלט אפשריות:

- (A) I'm A
- (B) I hold int
- (C) And its inner value is 20
- (D) i=1
- (E) i=2
- (F) i=3
- (G) i=4
- (H) i=10
- (I) i=20
- (J) I hold class B
- (K) And its inner value is
- (L) I'm B with int i=1
- (M) I'm B with int i=2
- (N) I'm B with int i=3

שאלה 1

מה יודפס בעקבות הקריאה ל-f1 בשורה 80?

- א. $(C) \leq (B) \leq (A)$
- ב. $(I) \leq (B) \leq (A)$
- ג. (I)
- ד. (H)
- ה. לא יודפס דבר!
- ו. אף אחת מהתשובות אינה נכונה.

שאלה 2

מה יודפס בעקבות הקריאה ל-f2 בשורה 81:

א. $(F) \leq (F) \leq (E)$

ב. $(F) \leq (E) \leq (E)$

ג. $(G) \leq (G) \leq (F)$

ד. $(G) \leq (F) \leq (F)$

ה. לא יודפס דבר!

ו. אף תשובה אינה נכונה.

שאלה 3

מה יודפס בעקבות הקריאה ל-f3 בשורה 82:

א. $(E) \leq (L) \leq (K) \leq (J) \leq (A)$

ב. $(D) \leq (M) \leq (K) \leq (J) \leq (A)$

ג. $(E) \leq (M) \leq (K) \leq (J) \leq (A)$

ד. $(E) \leq (K) \leq (J) \leq (A)$

ה. לא יודפס דבר!

ו. אף תשובה אינה נכונה.

שאלה 4

מה היה מודפס בקריאה ל-f3 במידה והיינו מעבירים את שורה 73 לסוף הפונקציה f3?

א. $(E) \leq (L) \leq (K) \leq (J) \leq (A)$

ב. $(D) \leq (M) \leq (K) \leq (J) \leq (A)$

ג. $(E) \leq (M) \leq (K) \leq (J) \leq (A)$

ד. $(E) \leq (K) \leq (J) \leq (A)$

ה. לא יודפס דבר!

ו. אף אחת מהתשובות אינה נכונה.

שאלה 5

כמה פעמים נעבור בשורה 10 במהלך הריצה של התוכנית ?

א. 0

ב. 1

ג. 2

ד. 3

ה. 4

ו. אף תשובה אינה נכונה.

שאלה 6

כמה פעמים נעבור בשורה 17 במהלך הריצה של התוכנית ?

א. 0

ב. 1

ג. 2

ד. 3

ה. 4

ו. אף תשובה אינה נכונה.

שאלה 7

מורן שירן דירן ונירן התווכחו האם שורה 18 היא שורה נדרשת בתוכנית.

מורן טען שהשורה מיותרת מכיון שממילא בשורה 17 אנחנו משחררים את המשתנה pMyStatic.

נירן טען שהשורה מיותרת מכיון שערך ה-default למשתנה סטטי הוא ממילא 0.

שירן טענה שהשורה היא חשובה, אמנם התוכנית היתה רצה אותו דבר בלי שורה זו, אבל היה נשאר זיכרון צף.

דירן טען שהשורה היא הכרחית ובלעדיה התוכנית תעוף בזמן ריצה על שחרור כפול של זיכרון.

מי מהחבורה צודק?

- א. מורן.
- ב. נירן.
- ג. שירן.
- ד. דירן.
- ה. כולם צודקים.
- ו. אף תשובה אינה נכונה.

חלק ב' – שאלת תכנות (51 נק')

חברת TipTop מעוניינת לשלוח ב-SMS* ידיעות חמות ומעניינות לבעלי טלפונים סלולריים, בהתאם לבקשת הלקוחות.

ה-SMS-ים המשודרים מחולקים לשלושה סוגים:

1. ידיעות שונות: מבזקי חדשות, ידיעות כלכליות, ידיעות בנושאי מחשבים וכו'.
2. הודעות פרסומיות בנושאים שונים: הצגות, קולנוע, מסעדות, מסיבות וכו'.
3. שערי סוף יום של מניות בבורסה בת"א.

להלן פרטים לגבי שלושת סוגי ה-SMS-ים המשודרים:

1. ידיעות שונות

הידיעות השונות מסווגות לפי נושאים. בנוסף, לכל ידיעה נקבע קוד של רמת חשיבות (רמה 1 – ידיעה חשובה, כמו למשל: סדאם חוסיין נתפס ע"י חיילי ארה"ב; רמה 3 – ידיעה פחות חשובה, כמו למשל: סגן ממלא מקום ראש עיריית דימונה מאיים בהתפטרות).

כל לקוח יכול להחליט מהם הנושאים המעניינים אותו ובכל נושא מה רמת הידיעות שהוא מעוניין לקבל. לכל רמת חשיבות נקבע מחיר שונה. מחיר הידיעות בתחומי עניין שונים עבור רמה נתונה הוא אחד.

2. הודעות פרסומיות

ההודעות הפרסומיות מחולקות אף-הן לנושאים שונים. נושאים אלו שונים מתחומי העניין של הידיעות השונות. הלקוח יכול לקבוע באיזה נושאים הוא מעוניין לקבל פרסומות. הודעות פרסומיות מועברות ללקוח ללא תשלום.

3. שערי סוף יום של מניות

הלקוח יכול לבקש SMS בנושא שערי סוף יום של מניות. כל מניה מזוהה ע"י מספר מניה מוסכם (ID). הלקוח יכול לקבוע אילו מניות מעניינות אותו ובסוף כל יום הוא יקבל SMS אחד שיכיל את שערי סוף היום של כל המניות שאותן בחר. מחיר SMS בנושא שערי מניות מורכב ממחיר X עבור משלוח ההודעה ועוד מחיר Y כפול מספר המניות שנבחרו, כאשר X ו-Y נקבעים ע"י החברה.

החברה מעוניינת ליצור מערכת שתאפשר את הפעולות הבאות:

- ניהול מאגרי הקטגוריות לנושאי הידיעות ולנושאים הפרסומיים (הנח שניתן רק להוסיף נושאים – לא ניתן לשנות או להוריד נושאים).
- ניהול רשימת מניות (כולל הוספת מניות לרשימת המניות אך ללא צורך במחיקה).
- רישום לקוחות חדשים (הנח שלא ניתן למחוק לקוחות).
- קבלת העדפות של לקוחות (נדרשת אפשרות גם לשינוי העדפות).

* הודעות טקסט לטלפון נייד.

- קליטה של SMS-ים של ידיעות שונות ושל פרסומות לצורך שידורם למשתמשים. אין צורך לאגור את ה-SMS-ים. הקליטה לא תתבצע מהמקלדת – יש לדאוג לפונקציה מתאימה שתקבל מחרוזת (הידיעה לשידור), ופרמטרים נוספים בהתאם (למשל, סיווג הידיעה לנושא המתאים ורמת חשיבות).
- שליחת ה-SMS-ים ללקוחות בהתאם להעדפות שהגדיר כל לקוח.
- קביעת מחירים לחיוב עבור ה-SMS-ים השונים (כולל אפשרות לעדכון מחירים) וחישוב החשבון לתשלום של כל לקוח (אין צורך לשמור פירוט חשבון).

לצורך קבלת שערי סוף היום של המניות ישירות מהבורסה, החברה רכשה מנוי לקו ישיר לבורסה. הקו מגיע יחד עם מחלקה אבסטרקטית שנקראת `StockExchangeConnection`, המכילה פונקציה טהורה שעליך לדאוג לממש במחלקה נגזרת:

```
virtual void EndOfDayRates(int num, long stockID[], double stockRate[]) = 0;
```

המחלקה קוראת לפונקציה `EndOfDayRates` באופן אוטומטי כאשר מפורסמים שערי סוף היום, עם הפרמטרים הבאים:

- `num`: מספר המניות בשידור.
 - `stockID[]`: מערך ממויין של מספרי מניות.
 - `stockRate[]`: מערך שערי מניות עבור כל מניה במערך `stockID[]`.
- למחלקת הבסיס יש `Default Empty Constructor` שיוזע להתחבר לשידור לבורסה.

הנח שלצורך משלוח ה-SMS-ים בפועל, קיימת כבר מחלקה בשם `SMS`, המכילה את הפונקציה:

```
static void Send(const string& sms, int num, long phoneNumber[]);
```

הפונקציה שולחת את ההודעת `sms` לכל מספרי הטלפונים במערך `phoneNumber`, שגודלו `num`.

ממשק המשתמש למערכת יהיה כנראה `Web`-י (כלומר – אינטרנטי), לכן אין צורך (ואסור!) ליצור במערכת הדפסות למסך או קבלת קלט מהמקלדת. המטרה היא שהמערכת תציע את הפונקציונליות הנדרשת על-מנת שיהיה אפשר לחבר אליה ממשק משתמש בהמשך.

הנחיות עזר לשאלה:

1. יש לשים לב שישנם למעשה שני סוגי SMS-ים במערכת:
 - א) SMS ידיעות ופרסומות, שמוזן למערכת באופן יזום – כולל פרמטר סיווג (באמצעות הפעלת פונקציה/יות במערכת), ואז משודר ללקוחות, בהתאם לסיווג, לרמת העניין (של ידיעות) ולהעדפות המשתמשים.
 - ב) SMS מניות שנשלח אוטומטית ללקוחות (בהתאם לבקשתם).

2. רצוי לייצר מחלקה מנהלת שתציע את כל הפונקציונליות הנדרשת באופן מרוכז ותנהל את הנתונים בתוכנית (תוך שימוש במחלקות נוספות במידת הצורך, כמובן).
3. אין צורך ב-main בתוכנית. אין ליצור הדפסות בתוכנית ואין לקבל קלט מהמקלדת. יש ליצור פונקציות שממשק משתמש עתידי יוכל להפעיל.

ענה על הסעיפים הבאים באופן מסודר:

סעיף א' (23 נקודות)

הצג את המחלקות הנדרשות לפתרון (prototypes מסודרים, ללא מימוש).

סעיף ב' (7 נקודות)

ממש את הקוד הנדרש לניהול לקוחות וקליטת הגדרותיהם.

סעיף ג' (7 נקודות)

ממש את הקוד הנדרש לניהול מחירי השירותים.

סעיף ד' (7 נקודות)

ממש את הקוד הנדרש לניהול ה-SMS-ים של ידיעות ופרסומות (ניהול הקטגוריות השונות, קליטת ה-SMS-ים למשלוח, שליחתם וחיוב חשבונות הלקוחות).

סעיף ה' (7 נקודות)

ממש את הקוד הנדרש לניהול ה-SMS-ים בנושא שערי מניות (קבלת השידור מהבורסה, שליחת ה-SMS-ים ללקוחות וחיוב חשבונות הלקוחות).

סוף !