

סמסטר א' תשס"ט – מועד א'

תאריך: 12 בפברואר 2009

י"ח בשבט תשס"ט

שעה: 09:00

משך הבחינה: 3½ שעות

חומר עזר: כל חומר עזר מותר

בחינה בקורס: תכנות מכוון עצמים ושפת ++C

מרצים: רוני גינזבורג, אמיר קירש, איריס רוזנבלום

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.
- 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 מחשבים ניידים. אין להעביר חומר עזר בין תלמידים במהלך המבחן.
- נא לכתוב בכתב קריא ולא מחובר.

נא למלא

ת.ז. _____

מספר מחברת נבחן _____

בהצלחה !

טבלת תשובות לחלק האמריקאי

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 1-4 מתייחסות לקטע הקוד הבא:

```

1. class TObject
2. {
3. public:
4.     TObject(int aType = 0)
5.         : m_type(aType), m_id(++s_idCounter)
6.         { cout << m_id; }
7.     virtual ~TObject() { print(); }
8.     int getId() const { return m_id; }
9.     int getType() const { return m_type; }
10.    virtual void print() const { cout << m_id; }
11. private:
12.     static int s_idCounter;
13.     int      m_type;
14.     int      m_id;
15. };
16.
17. int TObject::s_idCounter;
18.
19. template <class T>
20. class Array : public TObject
21. {
22. public:
23.     Array(unsigned arraySize = 8)
24.         : m_data(0), m_size(arraySize)
25.         { m_data = new T[arraySize]; }
26.     ~Array() { delete [] m_data; }
27.     T & operator [] (unsigned index) {
28.         if (index >= m_size) {
29.             index = 0;
30.         }
31.         return m_data[index];
32.     }
33.     virtual void print() const {
34.         while (m_size-->0) {
35.             cout << getId();
36.         }
37.     }
38. protected:
39.     T *      m_data;
40.     mutable unsigned m_size;
41. };

```

שאלה 1

מה יודפס בעקבות ה-main הבא :

```
1. void main()
2. {
3.     Array<int> b(4);
4. }
```

א. 0

ב. 1

ג. 00

ד. 11

ה. 0110

ו. אף תשובה אינה נכונה.

שאלה 2

מה יודפס בעקבות ה-main הבא :

```
1. void main()
2. {
3.     Array<TObject> b(4);
4. }
```

א. 12344321

ב. 1234554321

ג. 1234567887654321

ד. התכנית אינה מתקמפלת כיוון שהפרמטר TObject אינו עומד בדרישות מ-T.

ה. התכנית אינה מתקמפלת כיוון ש-TObject משמש בתפקיד כפול.

ו. אף תשובה אינה נכונה.

שאלה 3

מה יודפס בעקבות ה-main הבא :

```
1. void main()
2. {
3.     Array<Array<int> > b(4);
4. }
```

א. 11

ב. 12344321

ג. 1234554321

ד. 1234567887654321

ה. התכנית אינה מתקמפלת עקב אי עמידה בדרישות מ-T.

ו. אף תשובה אינה נכונה.

שאלה 4

מה יודפס בעקבות שורה מס' 5 ב-main הבא :

```
1. void main()
2. {
3.     Array<double> b(4);
4.     b[0] = 3.5;
5.     b.print();
6. }
```

א. הקוד אינו מתקמפל בגלל שורה 4.

ב. הקוד אינו מתקמפל בגלל שורה 5.

ג. יודפס: 3.5000 (כלומר – 3.5 ולאחריו שלושה אפסים).

ד. יודפס: 3.5 ואחריו עוד שלושה ערכי זבל כלשהם.

ה. יודפס: 1111

ו. אף תשובה אינה נכונה.

שאלות 5-7 מתייחסות לקטע הקוד הבא:

```
1.  template<class T>
2.  class A
3.  {
4.      T t;
5.      public:
6.          A(T p) : t(p) {}
7.          const A<T>& operator=(const A<T>& val)
8.          {
9.              t = val.t;
10.             return *this;
11.         }
12. };
13.
14. template <class T>
15. class Son : public A<T>
16. {
17.     public:
18.         Son (T p):A<T>(p) {}
19. };
20.
21. int main()
22. {
23.     A<int> a(1);
24.     Son<int> b(2);
25.     b = a;
26. }
```

שאלה 5

מה ערכו של חשדה t של b אחרי שורה 25 (נניח שנסתכל עליו עם debugger)?

- א. 2
- ב. 1
- ג. התכנית אינה עוברת קומפילציה.
- ד. התכנית קורסת בזמן ריצה.
- ה. ל-b (שהוא מסוג Son) בכלל אין שדה t.
- ו. אף תשובה אינה נכונה.

שאלה 6

הניחו כעת שה-main מוחלף ב-main-הבא:

```
1. int main()
2. {
3.     A<int> a(2);
4.     A<int> b(1);
5.
6.     Son<A<int> > s1(a);
7.     Son<A<int> > s2(b);
8.
9.     s1 = s2;
10. }
```

מה ערכו של השדה t שב-s1 אחרי שורה 9?

- אובייקט מקלאס A<int> שמכיל את הערך 2
- אובייקט מקלאס A<int> שמכיל את הערך 1
- התכנית אינה עוברת קומפילציה.
- התכנית אינה עוברת Linking (הסימבול <Son<A<int> > אינו מוכר).
- התכנית נכנסת ללולאה אינסופית.
- אף תשובה אינה נכונה.

שאלה 7

מורן שירן ולירן התכוננו למבחן ועברו על הקוד המקורי שבעמוד 6.

לירן טען: בשורה 6 מופעל copy c'tor של T.

שירן טען: אופרטור החשמה שבשורה 7 הינו מיותר כיון שפעולתו זהה ל-default-י.

מורן טען: בשורה 18 מופעל copy c'tor של A<T>.

מי מהשלושה צודק?

- א. לירן (הוא גם מצטיין דיקאן).
- ב. לירן ושירן.
- ג. מורן ולירן.
- ד. שלושתם צודקים.
- ה. שלושתם טועים.
- ו. אף אחת מהתשובות אינה נכונה.

חלק ב' – שאלת תכנות (51 נק')

עליכם לנהל תוכנה עבור חברת השמה.

דורש עבודה

- שם, מס' זהות (מותר להניח שאין שני אנשים עם אותו מס' זהות), כתובת
- ניסיון תעסוקתי קודם – מקומות עבודה ומתי עבד בהם (מאיזו שנה ועד איזו שנה) ובאיזה תפקיד. הנח שעבודה משנת 98 ועד 98 משמעותה שנת עבודה אחת, ומשנת 98 עד 99 משמעותה שנתיים.
- השכלה – מוסד אקדמי, באיזו שנה סיים לימודיו, לאיזה תואר (ראשון שני או שלישי)

חברה (מקום עבודה)

- שם חברה (מותר להניח שאין שתי חברות עם אותו השם)
- כתובת, שנת היווסדות החברה, שם המנכ"ל הנוכחי
- רשימת העובדים בחברה לפי שנים החל מהשנה בה קיימת החברה (עבור כל שנה, רשימת כל העובדים שעבדו בחברה באותה שנה)

מוסד אקדמי

- שם המוסד (מותר להניח שאין שני מוסדות עם אותו השם)
- כתובת, שנת היווסדותו, שם הנשיא הנוכחי
- רשימת מסיימי תואר לפי שנים החל מהשנה בה קיים המוסד (עבור כל שנה, רשימת כל המסיימים, ועבור כל מסיים – איזה תואר הוא סיים)

פעולות

1. מגיע עובד למאגר – מוסיפים אותו למאגר.
2. מגיע מוסד אקדמי למאגר – מוסיפים אותו למאגר.
3. מגיעה חברה למאגר – מוסיפים אותה למאגר. ייתכן והחברה מתווספת אך ורק לצורך אימות נתונים (ראו סעיף 5) – במקרה זה היא אינה זקוקה לעובדים. באם החברה זקוקה לעובדים אזי היא מציינת דרישות וכמה עובדים נדרשים עבור כל דרישה, הדרישה כוללת שם משרה (שיש לו משמעות תיאורית בלבד) וכן מצוינים שני מאפיינים:
 - מינימום תארים נדרש (בין 0 ל-3).
 - מינימום שנות ניסיון נדרשות (בסך כל מקומות העבודה הקודמים).
4. שאילתא – כמה עובדים בעלי תואר שלישי ישנם במאגר? (ייתכנו שאילתות נוספות אך אין עליכם לממש אותן).
5. אימות נתוני דורש עבודה – הפונקציה מקבלת מס' זהות של עובד ובודקת האם המועמד הצהיר אמת ואכן עבד באותם מקומות עבודה כפי שהוא טוען ואכן סיים תואר במוסד האקדמי שטוען שסיים בו תואר.
6. פעולת השמה לחברה – הפונקציה מקבלת שם חברה ומנסה להשיג לחברה עובדים על פי דרישתה.

סעיף א' (20 נקודות)

כתבו את המחלקות המשתתפות בשאלה. אין צורך לממש דבר בסעיף זה אלא אך ורק לכתוב שדות (data members) ו-prototypes של הפונקציות העיקריות בכל מחלקה.

סעיף ב' (16 נקודות)

ממשו את הפעולות (1) – (4) המוגדרות למעלה. חלוקת ניקוד: 4 נק' לכל פעולה.

סעיף ג' (15 נקודות)

ממשו את פעולות (5) ו- (6) המוגדרות למעלה. חלוקת ניקוד: 7 נק' לפעולה (5) ו-8 נק' ל- (6).

בפעולה (5) יש להוציא דו"ח ברור של הבדיקה שיודיע האם העובד הצהיר אמת לגמרי, או שהנתונים שציין אינם נכונים (ואז יש לציין את כל הנתונים שנמצאו שגויים לגבי העובד) וכן יש לציין בנוסף במידה ולא ניתן היה לבדוק נתונים מסוימים במקרה שחברות או מוסדות שהעובד ציין אינם מופיעים במערכת.

בפעולה (6) יש להציע לכל משרה את כמות העובדים הנדרשים אם ישנם כאלו במאגר. אסור להציע את אותו עובד למספר משרות עבור אותה חברה, אבל אין צורך לכתוב אלגוריתם מתוחכם שינסה למקסם את ההשמה עבור אותה חברה – כלומר, אם למשל, דרישות חברה מסוימת הינן ל-3 עובדים כדלקמן:

- (1) עובד אחד בעל תואר ראשון לפחות ו-5 שנות ניסיון לפחות – למשרת ברברן
- (2) עובד אחד בעל תואר ראשון לפחות ושנת ניסיון אחת לפחות – למשרת פלספן
- (3) עובד אחד בעל תואר שני לפחות ו-5 שנות ניסיון לפחות – למשרת דוברמן

ובמאגר ישנם 3 עובדים:

דון – תואר שני ו-5 שנות ניסיון, יון – תואר שני ושנת ניסיון, שון – תואר ראשון ו-6 שנות ניסיון

עלול להיווצר מצב כזה:

דון יאייש את המשרה הראשונה (הוא עומד בדרישות)

יון יאייש את המשרה השנייה (הוא עומד בדרישות)

והמשרה השלישית תישאר פנויה כי שון לא עומד בדרישות שלה.

(אלגוריתם מתוחכם היה משים את דון למשרת השלישית, משרת הדוברמן, ומאפשר לשבץ את שון למשרה הראשונה ואת יון לשנייה. אך כאמור אין צורך לממש אלגוריתם שכזה.)

סוף !