

סמסטר ב' תשס"ד
מועד: א' 21/6/2004
משך הבחינה: 3½ שעות
חומר עזר: כל חומר עזר מותר

בחינה בקורס: תכנות מכוון עצמים ושפת ++C

מרצה: אמיר קירש

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

חובה לספק
הסבר עבור
תשובה ו'

מס' ת.ז. : _____

א	ב	ג	ד	ה	ו	
						שאלה 1
						שאלה 2
						שאלה 3
						שאלה 4
						שאלה 5
						שאלה 6
						שאלה 7

הסברים לתשובות

חובה לספק הסבר במידה ונבחרה תשובה ו' (אף תשובה אינה נכונה).
שותר 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.  using namespace std;
2.
3.  class MyString: public string
4.  {
5.  public:
6.      MyString(char* str = ""): string(str)
7.      {
8.          cout << "in MyString c'tor with " << str << endl;
9.      }
10.     MyString(const MyString& str): string(str)
11.     {
12.         cout << "in MyString copy c'tor with " << str << endl;
13.     }
14.     ~MyString()
15.     {
16.         cout << "in MyString d'tor, str = " << *this << endl;
17.     }
18.     const MyString& operator=(const MyString& str)
19.     {
20.         cout << "in MyString operator= with " << str << endl;
21.         this->string::operator=(str);
22.         return *this;
23.     }
24. };
25.
26. struct Print
27. {
28.     template<typename T> void operator()(T& printable)
29.     {
30.         cout << printable << endl;
31.     }
32. };
33.
34. struct Free
35. {
36.     template<typename T> void operator()(T& freeable)
37.     {
38.         delete freeable;
39.     }
40. };

```

```
41. void f1()
42. {
43.     list<MyString> myList;
44.     // be aware: the strings are inserted to myList.begin()
45.     myList.insert(myList.begin(), "Wow");
46.     myList.insert(myList.begin(), "Shiga'on");
47.     for_each(myList.begin(), myList.end(), Print());
48. }
49.
50. void f2()
51. {
52.     list<MyString*> myList;
53.     // be aware: the strings are inserted to myList.end()
54.     myList.insert(myList.end(), new MyString("Madlik"));
55.     myList.insert(myList.end(), new MyString("Teruf"));
56.     for_each(myList.begin(), myList.end(), Print());
57.     for_each(myList.begin(), myList.end(), Free());
58. }
59.
60. void main()
61. {
62.     f1();
63.     f2();
64. }
```

תזכורת והנחיות עזר:

- כאשר נשלח איבר למבנה נתונים של ה-STL, הוא מתקבל ב-ByRef ומועתק לתוך מבנה הנתונים.
- הנח שה-d'tor של רשימה מקושרת של ה-STL מוחק את האיברים מההתחלה לסוף.

השאלות עצמן מתחילות בעמוד הבא.

להלן שורות פלט אפשריות שיכולות להתקבל במהלך הרצת התוכנית :

- (A) in MyString c'tor with
- (B) in MyString copy c'tor with
- (C) in MyString d'tor with str =
- (D) in MyString operator= with
- (E) Wow
- (F) Shiga'on
- (G) Madlik
- (H) Teruf
- (I) <some address>

שאלה 1

מה יודפס בעקבות הרצת שורה 45 ?

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

שאלה 2

מה יודפס בעקבות הרצת שורה 47 ?

- א. לא יודפס דבר.
- ב. $(E) \leq (F)$
- ג. $(F) \leq (E)$
- ד. $(E) \leq (C) \leq (E) \leq (E) \leq (B) \leq (F) \leq (C) \leq (F) \leq (F) \leq (B)$
- ה. $(F) \leq (C) \leq (F) \leq (F) \leq (B) \leq (E) \leq (C) \leq (E) \leq (E) \leq (B)$
- ו. אף תשובה אינה נכונה.

שאלה 3

נניח שהיינו משנים את אופרטור () של Print כך שיקבל את הפרמטר T כ-ByVal במקום ByRef. מה היה הפלט של שורה 47 ?

- א. עדיין לא היה מודפס דבר.
- ב. $(E) \leq (E) \leq (F) \leq (F)$
- ג. $(F) \leq (E) \leq (E) \leq (F)$
- ד. $(E) \leq (C) \leq (E) \leq (E) \leq (B) \leq (F) \leq (C) \leq (F) \leq (F) \leq (B)$
- ה. $(F) \leq (C) \leq (F) \leq (F) \leq (B) \leq (E) \leq (C) \leq (E) \leq (E) \leq (B)$
- ו. אף תשובה אינה נכונה.

שאלה 4

מה יודפס בעקבות שורה 48 (הסוגר המסולסל של סוף הפונקציה f1) ?

- א. לא יודפס דבר. (למה שיודפס משהו?)
- ב. $(E) \leq (F) \leq (C)$
- ג. $(F) \leq (E) \leq (C)$
- ד. $(E) \leq (C) \leq (F) \leq (C)$
- ה. $(F) \leq (C) \leq (E) \leq (C)$
- ו. אף תשובה אינה נכונה.

שאלה 5

מה יודפס בעקבות הקריאה ל- f2 בשורה 63 ?

- א. $(H) \leq (G)$
- ב. $(H) \leq (C) \leq (G) \leq (C) \leq (I) \leq (I)$
- ג. $(G) \leq (C) \leq (H) \leq (C) \leq (H) \leq (G)$
- ד. $(H) \leq (C) \leq (G) \leq (C) \leq (H) \leq (G) \leq (H) \leq (A) \leq (G) \leq (A)$
- ה. $(G) \leq (C) \leq (H) \leq (C) \leq (I) \leq (I) \leq (H) \leq (A) \leq (G) \leq (A)$
- ו. אף תשובה אינה נכונה.

שאלה 6

מה יודפס בעקבות שורה 64 (הסוגר המסולסל של סוף ה-main) ?

- א. לא יודפס דבר. (למה שיודפס משהו?)
- ב. $(H) \leq (C) \leq (G) \leq (C)$
- ג. $(I) \leq (I)$
- ד. $(I) \leq (C)$
- ה. $(I) \leq (I) \leq (C)$
- ו. אף אחת מהתשובות אינה נכונה.

שאלה 7

מורן שירן דירן ונירן התווכחו מה יקרה אם נוותר על שורה 57.

מורן טען ששורה 57 מיותרת מכיון שממילא ה-STL משחרר את הרשימה, לכן לא יקרה דבר אם נוותר עליה. מצד שני, גם אם נשאיר אותה לא יקרה כלום (אנחנו פשוט חוסכים עבודה ל-STL).

נירן טען ששורה 57 מיותרת וחובה למחוק אותה. אמנם בתוכנית שלנו היא לא מזיקה כי אנחנו לא עושים כלום עם הרשימה לאחר שורה 57, אבל באופן עקרוני אסור למחוק איברים ברשימה – זהו תפקיד של ה-STL.

שירן טוענת ללא שורה 57 התוכנית תתרסק כיון שה-STL ינסה לשחרר איברים בשימוש.

דירן טוען שללא שורה 57 יישאר בתוכנית זיכרון צף (אבל התוכנית לא תתרסק).

מי מהחבורה צודק?

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

חלק ב' – שאלת תכנות (51 נק')

פעולות של הקצאה ושחרור זיכרון נחשבות פעולות "יקרות" מבחינת זמן ריצה. מעוניינים לכתוב מחלקה שתייעל את ההקצאות הדינמיות במהלך הריצה של התוכנית, וזאת ע"י יצירת Pool של אובייקטים שיעמוד לטובת התוכנית וישתחרר בסיום הריצה. בכל פעם שהמתכנת יצטרך הקצאה דינמית של אובייקט מסויים, הוא יוכל לפנות בקוד ל-Pool ולבקש את האובייקט המתאים. בסיום השימוש באובייקט המתכנת לא ישחרר אותו מהזיכרון אלא יחזיר אותו ל-Pool לשימוש חוזר.

להלן דוגמה לשימוש אפשרי:

```
void f1()
{
    // get a Rectangle from the Rectangles ObjectPool
    Rectangle& rect = ObjectPool<Rectangle>::getObject();
    // do something ...
    f2(rect);
    // release the Rectangle back to its ObjectPool
    rect.release();
}

void f2(Rectangle& rect)
{
    // do something with the Rectangle...
}
```

להלן הדרישות מהמחלקה ObjectPool:

- המחלקה צריכה לאפשר יצירה של כל סוג של אובייקט, באמצעות שימוש ב- Templates.
- למחלקה תהיה מחלקת בסיס בשם GenericObjectPool שתכריז על פונקציה טהורה בשם releasePool (פונקציית void שלא מקבלת פרמטרים).
- למחלקה תהיה פונקציה סטטית בשם getObject שתחזיר אובייקט שאינו בשימוש מתוך ה-Pool.
- בפעם הראשונה שקוראים לפונקציה getObject תתבצע הקצאה דינמית של כמות כלשהי של אובייקטים (למשל – 100, אך רצוי להחזיק מספר זה כקבוע מסודר).
- כאשר המשתמש מבקש אובייקט וכל האובייקטים ב-Pool נמצאים כבר בשימוש ה-Pool ייגדל אוטומטית בכמות כלשהי של אובייקטים (למשל – 50, אך רצוי להחזיק מספר זה כקבוע מסודר).
- החזרת אובייקט ל-Pool תתבצע באמצעות הפעלת הפונקציה release על האובייקט. יש לשים לב שהאובייקט צריך לזכור מיהו ה-Pool שלו על-מנת להחזיר את עצמו ל-Pool (בקריאה ל-release לא נשלח ה-ObjectPool!).
- כאשר המשתמש מחזיר אובייקט ל-Pool ה-ObjectPool יפעיל על האובייקט פונקציה בשם clean, על-מנת לנקות את האובייקט משאריות לפני החזרתו ל-Pool.

אנחנו מעוניינים שלא נצטרך לממש את הפונקציה release בכל מחלקה שמתכוונת להתארח ב-ObjectPool, כיון שהלוגיקה של פונקציה זו היא אחידה. משום כך אנחנו מעוניינים ליצור מחלקת בסיס בשם PoolableObject.

להלן הדרישות מהמחלקה PoolableObject:

- המחלקה תהיה מחלקה אבסטרקטית.
- אובייקט מסוג PoolableObject יקבל בבנאי פוינטר ל-ObjectPool שלו.
- המחלקה תממש את הפונקציה release.
- המחלקה תצהיר על הפונקציה clean כפונקציה טהורה (0=).
- למחלקה יהיה virtual d'tor.

בנוסף, יש ליצור מחלקה בשם ObjectPoolsManager שתפקידה לנהל את כל ה-ObjectPools. הדרישות ממחלקה זו הן:

- למחלקה תהיה פונקציה סטטית בשם registerPool שבאמצעותה כל ה-ObjectPools יירשמו בעת יצירתם.
- למחלקה תהיה פונקציה סטטית בשם releasePools שתשחרר את כל ה-Pools באמצעות הפעלת הפונקציה releasePool על כל ה-Pools (המטרה היא שה-main יוכל לקרוא לפונקציה זו לפני סיום התוכנית). הפונקציה תדאג לשחרר את האובייקטים שממתינים ב-Pools אך לא תטפל בשחרור אובייקטים שנלקחו מה-Pools ולא שוחררו.

הנחיות עזר (לכל סעיפי השאלה):

המחלקה ObjectPool

- המחלקה יכולה להחזיק ב-private משתנה סטטי שייצג את ה-Pool של האובייקטים, למשל list של פוינטרים ל-PoolableObject. הפונקציה הסטטית getObject בודקת ממילא שה-list איננו ריק לפני שהיא מנסה להחזיר אובייקט מתוך ה-list, כך שבהתבסס על בדיקה זו יוחלט האם לייצר אוסף אובייקטים נוספים ולדחוף אותם ל-list (בקריאה הראשונה ה-list ריק ולכן תתבצע הקצאה של אובייקטים, אין צורך לדאוג לכך באופן מיוחד!).
- על-מנת לדעת אם לקרוא לפונקציה הסטטית ObjectPoolsManager.registerPool רק פעם אחת אפשר להחזיק דגל בוליאני שיסמן האם כבר קראנו לפונקציה זו.
- כאשר הפונקציה getObject מחזירה למשתמש אובייקט לשימוש, יש למחוק אובייקט זה מה-list כיון שאסור לנו להשתמש בהאובייקט זה שוב עד שיבוצע עליו release.

המחלקה PoolableObject

- הפונקציה release צריכה לדאוג להחזיר את האובייקט (this) ל-list של האובייקטים הפנויים ב-ObjectPool שלי.

המחלקה ObjectPoolsManager

- המחלקה צריכה להחזיק list של פוינטרים ל-GenericObjectPool (מחלקת הבסיס של כל ה-ObjectPools).

ענה על הסעיפים הבאים באופן מסודר:

סעיף א' (27 נקודות)

כתוב את הגדרת המחלקות הנדרשות לפתרון השאלה ללא מימושים.
הקפד על תיאור מפורט של ה-data members במחלקות.
מותר כמובן להוסיף פונקציות ומשתנים במידת הצורך, מעבר למה שתואר בהנחיות העזר.

סעיף ב' (8 נקודות)

כתוב מימוש מלא למחלקה ObjectPool ולמחלקת הבסיס שלה GenericObjectPool.

סעיף ב' (8 נקודות)

כתוב מימוש מלא למחלקה PoolableObject.

סעיף ד' (8 נקודות)

כתוב מימוש מלא למחלקה ObjectPoolsManager.

סוף !