

תאריך: 17.02.2000

שם הקורס: שפות תוכנה מתקדמות 1

(מועד ב')

משך הבוחן: שלוש שעות

חומר עזר: כל חומר כתוב מצולם או

מודפס (כולל ספרים)

□ משקל כל שאלה 50 נק'

- בכל תוכנית חייבים להיות הסברים, ושמות המשתנים חייבים להיות בעלי משמעות.
- אין להוסיף פונקציות אופרטורים או משתנים פרט לאלו שנדרשים בשאלות באופן מפורש.
- אסור להגדיר **class members** כ-**public**.

1. במפעל ישנם הרבה מחלקות, ובכל מחלקה ישנה יחידת מיחשוב שמתפלת בכל צורכי המיחשוב המחלקתיים. בנוסף ישנה מחלקת מיחשוב מרכזית של כל המפעל. כדי להקל על פעילות המחלקה מול המחלקות האחרות הוגדר דפוס פעולה של העברת אינפורמציה. המידע אמור להיות מרוכז ב-**class** שבו יוגדרו **operator overloading** לכל האופרטורים הנחוצים, פונקציות עזר וכו'. ב-**class** בו כל המידע מרוכז אמור להיות שדה נתונים (כמובן ב-**private**) בשם **Summary**. שדה זה יכול להיות מכל טיפוס שהו כולל **class** אחר. בכל **classes** אמורה להיות אותה **friend function** בשם **Print** אשר באמצעותה יודפס שדה ה-**Summary** של כל **class** מתוך תוכנית או פונקציה חיצונית.

דוגמא ל-class:

class Example

```
{  
    השלם את ה-declaration של הפונקציה  
    friend void Print ( .... ) ;  
    friend ostream & operator << ( ostream & , const Example & rE ) ;  
    .....  
private :  
    AnyType Summary ;  
    .....  
public :  
    .....  
}
```

שימו לב: (I) שדה ה-**Summary** יכול להיות מטיפוס שונה בכל מחלקה ומחלקה,

(II) ה-**class Example** אינו יורש שום **class** אחר.

1.1. השלימו את הגדרת הפונקציה **Print** בתוך ה-**class Example**, וכתבו את הפונקציה **Print** כך

שתוכל להדפיס כל שדה **Summary** של כל **class**.

1.2. נניח ישנו **class Point** אשר לו שני שדות של מספרים ממשים המהווים קואורדינטות של הנקודה –

fX ו-**fY**. בנוסף ישנם ה-**classes** הבאים:

Square: בעל שלוש שדות ממשים – **fX**, **fY** שהם מרכז הריבוע ו-**fSide** שהוא אורך הצלע.

Circle: בעל שלוש שדות ממשים – **fX**, **fY** שהם מרכז העיגול ו-**fRadius** שהוא רדיוס העיגול.

Box: בעל שלוש שדות ממשים – **fX**, **fY** שהם מרכז התיבה ו-**fSide** שהוא אורך הצלע.

Ball: בעל שלוש שדות ממשים – **fX**, **fY** שהם מרכז הכדור ו-**fRadius** שהוא רדיוס הכדור.

בצורות התלת-ממדיות הניחו שמרכז הצורה על ציר Z תמיד אפס.

לכל ה-**classes** ישנה הפונקציה **Center** אשר מחזירה ל-**main** את מקום מרכז האובייקט. לצורות

הדו-ממדיות גם פונקציה **Area** שמחשב שטח, ולתלת-ממדיות פונקציה **Volume** המחשבת נפח.

כתוב את כל ה-**classes**, והצדיקו כל בחירה שבצעתם.

שימו לב – אין לכתוב יותר מפונקציה אחת עבור **class** דו-ממדי, ויותר משתי פונקציות בכל **class**

תלת-ממדי.

שטח מעטפת המעגל: $4\pi r^2$

נפח מעגל: $\frac{4}{3}\pi r^3$

1.3. הניחו שהשדה **Summary** הוא מטיפוס **Box**. הפלט בהפעלת פונקציה **Print** יראה באופן הבא:

Left Up (-2.1 , 7.6)

Left Down (-2.1 , 2.6)

Right Down (3.1 , 2.6)

Right Up (3.1 , 7.6)

כלומר 4 שורות בצורה שהודגמה.

השלם את ה-**class Box** כך שהדבר יהיה אפשרי (כלומר הגדירו פונקציות, אופרטורים ומשתנים

הכרחיים). **נמקו את בחירתכם.**

2. בסוף השאלה ישנה פונקציה **LoadPersonsData** אשר בעזרתה שמרו נתונים בקובץ.

2.1. הסבירו את הפונקציה, האם חסר משהו בפונקציה לעבודה מסודרת עם **DataFile**.

2.2. הוגדרו שני **classes** כאופציה לטעינת הנתונים:

```
class Data1
```

```
{
```

```
private :
```

```
char * m_pName ;
```

```

short  m_sAge    ;
char   m_cSez    ;
public :
Data1 ( )
{
                                                    constructor השלימו
}
~Data1 ( )
{
                                                    destructor השלימו
}
void LoadData ( const char * szFileName ,      כיתבו את הפונקציה
                const unsigned uiPersonsNumber ) ;
}
;
class Data2
{
private :
char   m_Name [ 13 ] ;
short  m_sAge    ;
char   m_cSez    ;
public :
Data1 ( )
{
                                                    constructor השלימו
}
~Data1 ( )
{
                                                    destructor השלימו
}
void LoadData ( const char * szFileName ,      כיתבו את הפונקציה
                const unsigned uiPersonsNumber ) ;
}
;

```

עליכם להשלים את ה-constructors, destructors ו-LoadData functions.
הפונקציה LoadData מקבלת את שם הקובץ בו נמצאים הנתונים וטוענת את פירטי האדם שנימצא

במקום ה-**uiPersonsNumber** בקובץ. הפונקציה ניגשת ישירות למקום ממנו צריך להתחיל לקרוא את הנתונים וטוענת אותם לתוך האובייקט.

2.3. כתבו פונקציה אלטרנטיבית ל-**LoadPersonsData** לשמירת נתונים עבור **class Data2**. מותר להשתמש רק פעם אחת בפונקציית **write**.

2.4. האם אותה הפונקציה טובה גם עבור **class Data1**. נמקו תשובתכם.

2.5. נניח שהנתונים נשמרו עם הפונקציה מסעיף 2.3. האם הפונקציות לקריאת נתונים מקובץ בהכרח יעבדו כשורה עם הפונקציות שכתבתם בסעיף 2.2? נמקו תשובתכם. אם התשובה שלילית תקנו את הפונקציות.

```
#include <iostream.h>
#include <fstream.h>

Void LoadPersonsData ( const char * szFileName ,
                        char szName [ 13 ] ,
                        short sAge ,
                        char cSex )
{
    ofstream DataFile ( szFileName , ios :: app ) ;
    DataFile . write ( reinterpret_cast < const char * > ( szName ) ,
                      sizeof ( szName ) ) ;
    DataFile . write ( reinterpret_cast < const char * > ( & sAge ) ,
                      sizeof ( sAge ) ) ;
    DataFile . write ( reinterpret_cast < const char * > ( & cSex ) ,
                      sizeof ( cSex ) ) ;
}
```

• שאלת בונים של 10 נק'

3. נתון ה-class הבא:

```
class Int
{
private :
    int m_iNum ;
public :
    Int ( const int iNum )
    : m_iNum ( iNum )
{
}
```

```

    }
}
;

```

נתון ה-main הבא:

```

int main ( void )
{
    Int I1 ;
    Int I2 ( 3 ) ;
    int iValue ;
    .....
    I1 = I2 + 7 - iValue * Int ( 5 ) / 10 ;
    return 0 ;
}

```

בצעו **operator overloading** לשני אופרטורים בלבד כדי לפתור ביטוי זה (אין להשתמש ב- **friend** functions).

בהצלחה