

תאריך: 08.02.2000

שם הקורס: שפות תוכנה מתקדמות 1

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.
1. בסוף הבחינה תמצאו תוכנית ללא הסברים ועם חלקים חסרים (מעמוד 3 והלאה).
 - 1.1. ב-`class Data` השלם את ה-`(const int) += operator`. אופרטור זה אמור לשרשר לאובייקט קיים אובייקט נוסף אם הערך `Integer` איתו מתבצעת פעולת `+= (Data += int)`.
(* שימו לב אחר ביצוע פעולה זו בעם אחת תיווצר שרשרת של שני אובייקטים, אחרי ביצוע פעולה זו `N` פעמים תיווצר שרשרת של `N+1` אובייקטים. כל איבר מתוסף לסוף השרשרת!!!)
 - 1.2. כתוב את ה-`Destructors` של שני ה-`classes`.
 - 1.3. ב-`class Chain` כתוב את ה-`(~) operator`. אופרטור זה אמור להפוך את סדר הנתונים: כלומר אם הנתונים היו `1,2,3,4,5` אזי התוצאה תהיה `5,4,3,2,1`. התוצאה של האופרטור תוחזר לאובייקט עצמו.
 - 1.4. הסבר מה עושה התוכנית, וכן כל פונקציה בשני ה-`classes`.
 - 1.5. כתוב את ה-`prototypes` של שני ה-`classes` כך שהם יהיו `Templates` במקום `Integer` בלבד.
 2. עליכם לכתוב שלושה `classes` בכל `class` רק את ה-`default constructor, destructor, PrintData` (שם הפונקציה אינו ניתן לשינוי-פונקציה זו לא מקבלת ולא מחזירה פרמטרים והיא `const`).
 - Class Word** – יש לו רק שדה אחד: `*char`. הוא מקבל `string` אשר מסתיים ב-`'\0'` וב-`constructor` מעתיק אותו לשדה.
 - PrintData** – רק מדפיסה את המילה בלי לרדת שורה.
 - Class Line** – יש רק שדה אחד: `*Word`. הוא מקבל `string` אשר מסתיים ב-`'\0'` וב-`constructor` מעתיק אותו לשדה על ידי יצירת מערך דינמי של `Words`. ההפרדה בין שתי מילים ב-`string` באמצעות `'\t'`.
 - PrintData** – מדפיסה את השורה ובסופה יורדת שורה (בין שתי מילים אמור לבוא רווח).
 - Class Page** – יש רק שדה אחד: `*Line`. הוא מקבל `string` אשר מסתיים ב-`'\0'` וב-`constructor` מעתיק אותו לשדה על ידי יצירת מערך דינמי של `Lines`. ההפרדה בין

שתי שורות ב-**string** באמצעות '\n'.

PrintData –מדפיסה את העמוד.

2.1. כתבו פונקציה חיצונית אשר תקבל כפרמטר **reference** לאחד משלושת ה-**classes** ומדפיסה את המידע.

```

/*****
/*          Main Program          */
*****/

#include <iostream.h>
#include "DataClass.h"
#include "ChainClass.h"

int main ( void )
{
    Chain          NewChain ( -3 )          ;
    unsigned short  usTest                   ;

    do
    {
        cout << "If you want to add to chain press 0: "      ;
        cin >> usTest                                         ;
        if (!usTest )
        {
            int          iNewNum                       ;

            cout << "Input New Number: "                  ;
            cin >> iNewNum                                  ;

            NewChain += Data ( iNewNum )                  ;
        }
    }
    while ( !usTest )                                       ;

    NewChain . PrintChain ( )                             ;

    ~NewChain                                             ;
    NewChain . PrintChain ( )                             ;

    return 0                                              ;
}

```

```

/*****
/*          Data Class          */
*****/

#ifndef    Data_FinalTest0199_1
#define    Data_FinalTest0199_1

#include <iostream.h>

class Data
{
    private :
        int          m_iData          ;
        Data          *   m_pData          ;
    public:
        Data ( const int iData = 0 )
        :   m_iData ( iData )
        {
            m_pData = 0                ;
        }

        ~Data ( )
        {

        }

        bool operator > ( const Data & rD )    const
        {
            return ( m_iData > rD.m_iData )    ;
        }

        bool operator < ( const Data & rD )    const
        {
            return ( m_iData < rD.m_iData )    ;
        }
}

```

השלם את ה-destructor

```

bool operator == ( const Data & rD )    const
{
    return ( m_iData == rD.m_iData )    ;
}

Data & operator += ( const int iData )
{

}

int GetData ( )    const
{
    return m_iData    ;
}

Data * GetPtr ( )    const
{
    return m_pData    ;
}

void SetData ( const int iData )
{
    m_iData = iData    ;
}

void SetPtr ( Data * const pData )
{
    m_pData = pData    ;
}

void PrintData ( )    const
{
    cout << m_iData
        << endl    ;
}
}

```

השלם אופרטור זה

```
#endif
```

```
/*
Chain Class
*/
```

```
#ifndef Chain_FinalTest0199_1
#define Chain_FinalTest0199_1
```

```
#include "DataClass.h"
```

```
class Chain
```

```
{
    private :
        Data * m_pChain ;
        Data * m_pHelpChain ;
    public:
        Chain ( const int iData = 0 )
        {
            m_pChain = new Data ( iData ) ;
            m_pHelpChain = m_pChain ;
        }

        ~Chain ( )
        {

        }

        Chain & operator += ( const Data & rD ) ;

        Chain & operator ~ ( ) ;

        void PrintChain ( ) const ;
}
```

```
#endif
```

השלם את ה-destructor

```

/*****
/*          Chain Class Functions          */
*****/

#include "DataClass.h"
#include "ChainClass.h"

Chain & Chain :: operator += ( const Data   & rD )
{
    Data   *   pD = 0 ;

    if( *m_pHelpChain > rD )
    {
        m_pHelpChain = new Data ( rD.GetData ( ) ) ;
        m_pHelpChain -> SetPtr ( m_pChain ) ;
        m_pChain = m_pHelpChain ;
        return *this ;
    }

    while ( *m_pHelpChain < rD )
    {
        if ( m_pHelpChain -> GetPtr ( ) )
        {
            pD = m_pHelpChain ;
            m_pHelpChain = m_pHelpChain -> GetPtr ( ) ;
        }
        else
        {
            *m_pHelpChain += rD.GetData ( ) ;
            m_pHelpChain = m_pChain ;

            return *this ;
        }
    }
}

```

```

    pD = new Data ( rD.GetData ( ) )          ;
    pD -> SetPtr ( m_pHelpChain )             ;
    m_pHelpChain = m_pChain                   ;

    return *this                               ;
}

```

```

Chain & Chain :: operator ~ ( )
{

}

```

השלם אופרטור זה

```

void Chain :: PrintChain ( )    const
{
    Data    *   pD = m_pChain          ;
    while ( pD )
    {
        pD -> PrintData ( )           ;
        pD = pD -> GetPtr ( )         ;
    }
}

```


תיקון לשאלה מס' 2:

SetWord

ל-**class Word** ניתן להוסיף ב-**public** פונקציה:

SetLine

ל-**class Line** ניתן להוסיף ב-**public** פונקציה:

m_usWordsNum

וב-**private** משתנה:

m_usLinesNum

ל-**class Line** ניתן להוסיף ב-**private** משתנה: