תכנות מונחה עצמים
עבודת הגשה 3

מועד הגשה: 1.5.2016 בשעה 23:50

הוראות הגשה:

1. **אנא קראו בעיון את כל תיאור העבודה בטרם תתחילו לכתוב קוד.**
2. הגשה באופן עצמאי בלבד. הגשה בקבוצות תוביל לציון 0 בעבודה.
3. אין לשתף או להעתיק את העבודה או חלקים ממנה. עבירה על הוראה זו תוביל לציון 0 בעבודה.
4. הגשה דרך מערכת מודול בלבד. שום עבודה לא מתקבלת במייל!
5. למחלקות המפורטות בעבודה ניתן להוסיף מתודות נוספות אך אסור להוסיף שדות.
6. יש למקם כל מחלקה שיהיה עליכם ליצור, בשני קבצים נפרדים H ו-CPP. יש להכניס את החלק התיאורטי בקובץ וורד נפרד. יש להכניס את כל הקבצים של החלק המעשי + סייבר + קובץ הוורד לתיקיה בשם Ex3 , ואז לכווץ יחד. נדרש להגיש קובץ אחד בפורמט RAR או ZIP המכיל את כל הקבצים של כל השאלות. לקובץ המכווץ יהיה שם המהווה את מספר ת.ז. של המגיש.
7. **שאלות ובקשות בקשר לעבודה להפנות אך ורק למרצה האחראית לתרגיל, סבטלנה רוסין, במייל: sceassign2016@gmail.com.**

חלק א' – תאורטי (מענה בקובץ טקסט – וורד): 7 נקודות

1) האם יש מקרים שבהם חייבים להשתמש בfriend ? הסבר.
2) האם קוד הבא מכיל שגיאות (לפרט באיזה שורה ומה היא השגיאה)

```
1) #include <iostream>
2) using namespace std;
3) class A
4) {
5)       int num1, num2;
6) public:
7)       A(int i):num1(i),num2(i) { }
8)       A(int i, int j):num1(i),num2(j){ }
9)       friend ostream& operator<<(ostream& os, const A& obj){
10)                    return os << num1 << ' ' << num2 << endl;
11)             }
12)      };
13)int main()
14){
15)          A a1 = 1;
16)          A a2(2);
17)          A a3(4, 5);
18)          A a5 = (A)8;
19)          cout << a1 << ' ' << a2 << ' ' << a3 << ' ' << a5;
20)          int i = a1;
21)          cout << i;
22)      }
```

3) מה ההבדל העקרוני בין אופרטור ההמרה לבין בנאי ההמרה?
4) מה הוא ההבדל בין בנאי העתקה לבין אופרטור השמה (=)?

5) מהי המוטיבציה להעמסת אופרטורים?
6) האם הטענה הבאה היא טענה נכונה :*אופרטור שמחזיר אובייקט מקומי חייב להחזיר אותו by value*. יש לתת נימוק לתשובה.
7) איך ניתן למנוע שימוש באופרטור השמה (=)?

חלק ב' – מעשי (ההגשה היא של קבצי ה- CPP ו-H בלבד) – 93 נקודות:

בתרגיל הנ"ל עליכם לממש **מערכת ניהול של המכללה**.
עליכם לממש מחלקות הבאות: Student, Course, Department ו- College .
Student - המחלקה המתארת את פרטי הסטודנט.
השדות הן:
- ✓ שם הסטודנט (שם פרטי ושם משפחה)
- ✓ ת.ז.
- ✓ מין
- ✓ גיל
- ✓ רשימה (מערך) דינאמית של המצביעים לקורסים שהסטודנט רשום עליהם (כלומר, **Course וכמות הקורסים) . כאשר אובייקט-סטודנט נוצר הרשימה של הקורסים ריקה. בהמשך יהיה ניתן לרשום סטודנט לקורס.

יש לכתוב בנאים, מתודות get ו-set רלוונטיים לסטודנט. בנוסף יש להוסיף הורס ואופרטורים:
= (השמה) - בין האובייקטים מאותו סוג (אבל אם תידרש לבצע השמה נוספת אז להוסיף גם אותה).
== (בדיקת שוויון על פי ת.ז של הסטודנט)
> (בדיקה האם הממוצע ציונים של הסטודנט גדולה יותר מהממוצע של הסטודנט-פרמטר)
>> (פלט של פרטי הסטודנט, כולל רשימה של הקורסים עליהם הוא רשום)
<< (קלט של פרטי הסטודנט מלבד קורסים)
=+ (מוסיף קורס לרשימה של הקורסים. אם קורס כבר קיים ברשימה אין להוסיף אותו שנית ולא לנקוט בפעולה נוספת)
=- (מוריד קורס מהסטודנט, יש לעדכן את גודל הרשימה בהתאם. המחיקה היא לפי מספר קורס. אם קורס לא קיים ברשימה אז לא לנקות בפעולה נוספת).
[ ] (מחזיר את הקורס באינדקס - מקבל את מיקום הקורס ברשימה ומחזיר את הקורס הממוקם במיקום זו)
ניתן להוסיף מתודות נוספות במידת הצורך אך אסור להוסיף שדות .

Course - המחלקה המתארת את פרטי הקורס.
השדות הן:
- ✓ שם הקורס
- ✓ מספר הקורס
- ✓ רשימה (מערך) דינאמית של המצביעים לסטודנטים הרשומים לקורס (כלומר
- ✓ ,** Student וכמות הסטודנטים)
- ✓ רשימה (מערך) דינאמית של הציונים עבור הסטודנטים הרשומים לקורס (מסוג int* בגודל של כמות הסטודנטים - כאשר סטודנט נרשם לקורס הציון שלו צריך להיות 1- ובהמשך יהיה ניתן לשנות את הציון).

יש לכתוב בנאים, מתודות get ו-set רלוונטיים לקורס. בנוסף יש להוסיף הורס ואופרטורים:
= (השמה) - בין האובייקטים מאותו סוג (אבל אם תידרש לבצע השמה נוספת אז להוסיף גם אותה).
== (בדיקת שוויון על פי מספר הקורס)
> (בדיקה האם כמות הסטודנטים הרשומים לקורס גדולה מכמות הסטודנטים הרשומים לקורס-פרמטר)
<< (פלט של פרטי הקורס, בנוסף יש להדפיס את פרטי כל הסטודנטים הרשומים לקורס כולל ציונים שלהם בקורס זה במידה ויש)
>> (קלט של פרטי הקורס מלבד סטודנטים והציונים)
+= (מוסיף סטודנט לקורס. אם סטודנט כבר רשום לקורס אין להוסיף אותו שנית ולא לנקות באף פעולה נוספת. יש לעדכן גם את מערך הציונים בהתאם.).
-= (מוריד סטודנט מהקורס במידה וסטודנט רשום לקורס, יש לעדכן את גודל הרשימה בהתאם. המחיקה היא לפי ת.ז. של הסטודנט יש לעדכן גם את מערך הציונים בהתאם.).
[ ] (מחזיר סטודנט באינדקס - מקבל את מיקום הסטודנט ברשימה ומחזיר את הסטודנט הממוקם במיקום זו)
ניתן להוסיף מתודות נוספות במידת הצורך אך אסור להוסיף שדות .

Department - המחלקה (class) המתארת מחלקה (department) במכללה.
השדות הן:
- ✓ שם המחלקה
- ✓ קוד המחלקה
- ✓ רשימה (מערך) דינאמית של המצביעים לקורסים הניתנים ע"י המחלקה (כלומר Course **, וכמות הקורסים)
- ✓ רשימה דינאמית (מערך) של המצביעים לכלל הסטודנטים הלומדים במחלקה (כלומר, Student ** וכמות כללית של הסטודנטים)
- ✓ רשימה (מערך) דינאמית של המצביעים לסטודנטים עם מצב אקדמי לא תקין (הממוצע ציונים של כל הקורסים קטן מ-65) (כלומר,Student ** וכמות הסטודנטים עם מצב אקדמי לא תקין).

בזמן יצירה של המחלקה כל הרשימות ריקות, בהמשך יהיה אפשר להוסיף קורסים וסטודנטים למחלקה.

יש לכתוב בנאים, מתודות get ו-set רלוונטיים לקורס. בנוסף יש להוסיף הורס ואופרטורים:
= (השמה) - בין האובייקטים מאותו סוג (אבל אם תידרש לבצע השמה נוספת אז להוסיף גם אותה).
> (בודק האם כמות הסטודנטים עם מצב האקדמי תקין גדולה יותר מאשר במחלקה-פרמטר)
<< (פלט של פרטי המחלקה - יש להדפיס שם מחלקה, קוד מחלקה ,פרטי הקורסים הניתנים ע"י המחלקה, את רשימת הסטודנטים הלומדים במחלקה ואת רשימת הסטודנטים עם מצב אקדמי לא תקין (אם כבר חושבו).
>> (קלט של שם מחלקה וקוד מחלקה)
+= (יש לממש 2 אופרטורים, אחד מוסיף קורס למחלקה ושני מוסיף סטודנט למחלקה. במידה וקורס או סטודנט כבר קיימים במחלקה אין להוסיף שנית ולא לנקוט בשום פעולה נוספת)
ניתן להוסיף מתודות נוספות במידת הצורך אך אסור להוסיף שדות .

College - המחלקה המתארת תפריט. מדובר במחלקה שתנהל לנו את המערכת.
השדות הן :

- ✓ רשימה (מערך) דינאמית של מצביעים למחלקות הקיימות במכללה (כלומר Department**, וכמות המחלקות)
- ✓ רשימה (מערך) דינאמית של המצביעים לכלל הסטודנטים הלומדים במכללה (כלומר, Student ** וכמות כללית של הסטודנטים)

בנוסף לבנאים הדרושים והורס יש להוסיף מתודה **menu** אשר תדפיס למשתמש את כל האופציות הנתונות בפניו בתפעול המערכת ותיתן לו את היכולת לבחור מה להפעיל. המתודה תיתן למשתמש את האפשרויות הבאות:

1- *Add department* - הוספת מחלקה למכללה. אחרי הקשת 1 המשתמש יתבקש להזין שם וקוד של המחלקה. במידה וקוד המחלקה קיים במערכת יש להדפיס הודעה מתאימה.

2 - ***Add course to department*** - הוספת קורס למחלקה. לאחר הקשת 2 המשתמש יתבקש להזין את קוד המחלקה. אם המחלקה לא קיימת יש להדפיס הודעה מתאימה ולחזור לתפריט. אם המחלקה קיימת יש להזין את פרטי הקורס. יש להוסיף את הקורס למחלקה רק אם מספר הקורס לא מופיע במחלקה. אם קורס קיים כבר במחלקה יש להדפיס הודעה מתאימה.

3 - ***Add student to department*** - הוספת סטודנט למחלקה. לאחר הקשת 3 המשתמש יתבקש להזין את הקוד של המחלקה. אם המחלקה לא קיימת יש להדפיס הודעה מתאימה ולחזור לתפריט. אם המחלקה קיימת יש להזין את פרטי הסטודנט. יש להוסיף את הסטודנט למחלקה רק אם הסטודנט לא לומד במחלקה. אם הסטודנט כבר לומד במחלקה יש להדפיס הודעה מתאימה.

4 - ***Register student to course*** - רישום סטודנט לקורס. לאחר הקשת 4 המשתמש יתבקש להזין את קוד המחלקה, את מספר הקורס ואת מספר ת.ז. של הסטודנט. אם המחלקה לא קיימת במכללה או קורס לא קיים באותה מחלקה או סטודנט לא רשום באותה מחלקה יש להדפיס הודעה מתאימה. אחרת יש לרשום את הסטודנט לקורס - יש להוסיף את הסטודנט לרשימה של הסטודנטים בקורס ויש להוסיף את הקורס לרשימה של הקורסים של הסטודנט.

5 - ***Grading students in course*** - מתן ציון לסטודנטים בקורס מסוים. לאחר הקשת 5 המשתמש יתבקש להזין את מספר הקורס. אם הקורס לא קיים באף מחלקה יש להדפיס הודעה מתאימה. אחרת המשתמש יתבקש להזין ציון לכל סטודנט הרשום לקורס. ניתן להניח שכל קורס קיים אך ורק במחלקה אחת ולא יתכן שהוא קיים במספר מחלקות.

6 - ***Print course information*** - הדפסה של פרטי הקורס. לאחר הקשת 6 המשתמש יתבקש להזין את מספר הקורס. אם הקורס לא קיים באף מחלקה יש להדפיס הודעה מתאימה. אחרת יש להדפיס את שם וקוד המחלקה שקורס נמצא בה, את פרטי הקורס ופרטי כל הסטודנטים הרשומים לקורס. אם ניתן ציון לסטודנט אז יש להדפיס גם את הציון. אם יש ציונים לכל הסטודנטים בקורס יש להדפיס גם ממוצע ציונים בקורס.

7 - ***Print student information*** - הדפסה של פרטי הסטודנט. לאחר הקשת 7 המשתמש יתבקש להזין את מספר ת.ז. של הסטודנט. אם הסטודנט לא לומד במכללה יש להדפיס הודעה

המכללה האקדמית להנדסה סמי שמעון
קמפוס באר שבע ביאליק פינת בזל 84100 | קמפוס אשדוד ז'בוטינסקי 84, 77245 | www.sce.ac.il | חייג: *מהפלפון

מתאימה. אחרת, יש להדפיס את שם וקוד המחלקה שהוא נמצא בה, את פרטי הסטודנט ואת פרטי הקורסים שהסטודנט רשום עליהם.

8 - ***Print department information*** הדפסה של פרטי המחלקה. לאחר הקשת 8 המשתמש יתבקש להזין את קוד המחלקה. אם המחלקה לא קיימת יש להדפיס הודעה מתאימה. אחרת יש להדפיס את שם המחלקה, קוד מחלקה, את שמות הקורסים הניתנים ע"י המחלקה, את רשימת הסטודנטים הלומדים במחלקה ואת רשימת הסטודנטים עם מצב אקדמי לא תקין (אם כבר חושבו באופציה 9).

9 - ***Find students with bad academic status*** - מציאת סטודנטים עם מצב אקדמי לא תקין. לאחר הקשת 9 המשתמש יתבקש להזין את קוד המחלקה. אם המחלקה לא קיימת יש להדפיס הודעה מתאימה. אחרת, יש למצוא את כל הסטודנטים עם מצב אקדמי לא תקין ולשמור אותם ברשימה המיועדת לכך.

10 - ***Remove student from course*** - מחיקת סטודנט מהקורס. לאחר הקשת 10 המשתמש יתבקש להזין את מספר הקורס ומספר ת.ז. של הסטודנט. אם הקורס או סטודנט לא קיים או שסטודנט לא רשום לקורס יש להדפיס הודעה מתאימה. אחרת, יש למחוק את הסטודנט מהקורס. יש לעדכן גם רשימת הקורסים של הסטודנט.

11 - ***Exit*** - יציאה מהתוכנית. חובה לוודא יציאה מסודרת תוך שחרור כל הזיכרון המוקצה דינאמית!

ניתן להוסיף למחלקה הזאת מתודות נוספות במידת הצורך אך הן חייבות להיות private . אסור להוסיף שדות .

הערה: במהלך הבנייה של המערכת, כל קורס, סטודנט ומחלקה צריכים להיווצר רק פעם אחת, ובכל המקומות יש להחזיק מצביעים לאובייקטים. יש לממש את הבנאי העתקה ואופרטור = שיוצרים עותקים, אך אין להשתמש בהם בבניית המערכת.

להלן הפונקציה הראשית (main):

```
#include "College.h"

int main(){

      College c;
      c.menu();
      return 0;
}
```

סעיף סייבר (20 נקודות):

יש לממש את כל הרשימות של סטודנטים, קורסים ומחלקות ע"י שימוש ***ברשימות מקושרות בלבד*** (**ללא שימוש במערכים**).

עבודה פוריה !!!