

תכנות מונחה עצמים

תרגיל 2

מועד הגשה: 18/5/2021 23:50

הוראות הגשה:

1. הגשה באופן עצמאי בלבד. הגשה בקבוצות תוביל לציון 0 בעבודה.
2. אין לשתף או להעתיק את העבודה או חלקים ממנה. עבירה על הוראה זו תוביל לציון 0 בעבודה.
3. הגשה דרך מערכת מודול בלבד. שום עבודה לא מתקבלת במייל!
4. יש להכניס את החלק התיאורטי בקובץ וורד / pdf נפרד. יש להכניס את כל הקבצים מהחלק המעשי (קבצי cpp ו h בלבד!) לתיקייה בשם college ואז לכוף יחד. נדרש להגיש קובץ אחד בפורמט RAR או ZIP המכיל את כל הקבצים של כל השאלות. לקובץ המכוף יהיה שם המהווה את מספר ת.ז.

5. אסור להוסיף משתנים למחלקה, ניתן להוסיף מתודות (רגילות / אופרטורים) נוספות (עזר).

6. ניתן להשתמש בתרגיל בספריות iostream ו-string בלבד.
7. שאלות בקשר לעבודה להפנות אך ורק לאחראית התרגיל, פדות, במייל: pedutsh@ac.sce.ac.il

חלק א' – תאורטי (מענה בקובץ טקסט – PDF/Word) - 6 נקודות (1 נק' עבור כל שאלה):

1. באיזה מקרים הקומפילר יוצר אובייקטים זמניים. יש לתת לפחות 2 דוגמאות.
2. האם ניתן להעמיס (overload) שתי מתודות זהות כאשר אחת מהן מוגדרת כ-const והשנייה לא?
3. האם מתודה static של המחלקה יכולה להיות מוגדרת כ-const?
4. מהי המוטיבציה לשימוש באופרטור כ-friend?
5. האם ניתן ליצור אופרטור static למחלקה?
6. האם הטענה הבאה נכונה: לא ניתן להפעיל מתודה static ללא שימוש באובייקט. תן נימוק קצר.

חלק ב' – מעשי - הכללה ושימוש באופרטורים ובהקצאות דינמיות (ההגשה היא של קבצי ה CPP ו H-בלבד) – 94 נקודות :

שמות הקבצים שיוגשו הם

Course.h
Course.cpp
Student.h
Student.cpp
Menu.h
Menu.cpp
Driver.cpp

בחלק זה תידרשו לבנות מערכת לניהול קורסים במחלקה ספציפית. חשוב להדגיש שאין צורך ליצור את האובייקטים בכמה מקומות אלא ליצור אותם במקום יחיד ולהשתמש במצביעים שלהם. תהינה המחלקות הבאות:

מחלקת קורס – Course

לכל קורס יש שם המיוצג ע"י מחרוזת (מחלקת string של cpp), מספר קורס המיוצג ע"י מספר שלם, רשימת הסטודנטים המשוויכים לקורס המיוצגת ע"י מצביע כפול לאובייקט מטיפוס Student, מספר הסטודנטים הרשומים לקורס, מערך ציונים של הסטודנטים הרשומים לקורס. המחלקה תכיל לפחות את המתודות הבאות:

1. בנאי (default constructor) וכן בנאי המקבל אל הפרמטרים שם, מסי קורס ומאפס מצביעים) הורס
2. אופרטור =
3. אופרטור == להשוואה בין שני קורסים המקבל כפרמטר קורס ומחזיר true במידה ושני הקורסים שווים ו-false אחרת. שני קורסים שווים במידה ומספר הקורס שלהם זהה.
4. אופרטור += המקבל התייחסות לאובייקט Student ומוסיף את הכתובת שלו למערך מצביעים לסטודנטים.
5. אופרטורים עבור קלט ופלט:

אופרטור הקלט (>>) יקלוט את שם הקורס ואת מספר הקורס. דוגמא לפלט של אופרטור הקלט:

Enter course's name:

C++

Enter course's id

1

אופרטור הפלט (<<) ידפיס את שם הקורס, את מספר הקורס, את שם המחלקה שהקורס נלמד בה ואת פרטי הסטודנטים והציונים שלהם בקורסים (במידה וישנו ציון). דוגמא לאופרטור הפלט:

Course name: C++

Course ID: 1

Number of students in course: 0

No students in this course.

מחלקת סטודנט – Student

לכל סטודנט יש שם המיוצג ע"י מחרוזת (מחלקת string של cpp), ת"ז המיוצגת ע"י מחרוזת(תעודת הזהות תיוצג ע"י מחלקת string של cpp ותכיל ספרות בלבד), מגדר המיוצג ע"י התווים M או F, גיל המיוצג ע"י מספר שלם, רשימת קורסים המיוצגת ע"י מצביע כפול לאובייקט מטיפוס Course, מספר קורסים. ניתן להניח כהפרמטרים חוקיים ואין צורך לבדוק. המחלקה תכיל לפחות את המתודות הבאות:

1. בנאי (default constructor) וכן בנאי המקבל אל שם הסטודנט, ת.ז., מגדר וגיל ומאפס את המצביע)
2. הורס

3. אופרטור =
4. אופרטור == להשוואה בין שני סטודנטים המקבל כפרמטר סטודנט ומחזיר true במידה ושני הסטודנטים שווים ו-false אחרת. שני סטודנטים שווים במידה תעודת הזהות שלהם זהה.
5. אופרטור += המקבל התייחסות לאובייקט Course ומוסיף את הכתובת שלו למערך מצביעים לקורסים.
6. אופרטורים עבור קלט ופלט:
- אופרטור הקלט (>>)** יקלוט את שם הסטודנט, את מספר תעודת הזהות, את המין ואת גיל הסטודנט. דוגמא לפלט של אופרטור הקלט:

Enter student's name

Doe John

Enter student's id

123456789

Enter student's gender <F-female/M-male>

M

Enter student's age

22

אופרטור הפלט (<<) ידפיס את שם הסטודנט, את מספר תעודת הזהות, את המין, את הגיל, את שם המחלקה שהסטודנט לומד בה ואת פרטי הקורסים (במידה וישנם קורסים). דוגמא לאופרטור הפלט:

Name: John Doe

ID: 123456789

Gender: M

Age: 22

The student is not registered to any courses.

מחלקת תפריט - Menu

מדובר במחלקה שתנהל לנו את המערכת. המחלקה אמורה לשמור את רשימת קורסים המיוצגת ע"י מצביע כפול לאובייקט מטיפוס Course, מספר קורסים וגם כן את רשימת הסטודנטים המיוצגת ע"י מצביע כפול לאובייקט מטיפוס Student, מספר הסטודנטים.

בנוסף default ctor והורס יש להוסיף אליה הפונקציה Run. היא תדפיס למשתמש את כל האופציות הנתונות בפניו בתפועל המערכת ותיתן לו היכולת לבחור מה להפעיל. המתודה תיתן למשתמש את האפשרויות הבאות:

המחלקה תכלול לפחות את המתודות הבאות:

1. default constructor

2. הורס

3. מתודה להצגה ותפעול התפריט (Run)

4. מתודה להוספת קורס חדש

5. מתודה להוספת סטודנט חדש

6. מתודה לרישום סטודנט לקורס

7. מתודה למתן ציון לסטודנט בקורס

8. מתודה להדפסת פרטי קורס

9. מתודה להדפסת פרטי סטודנט

משתמש מקיש	שם הפעולה	הערות על הפעולה
1	הוספת קורס חדש	לאחר הקשה, המשתמש יתבקש להזין את פרטי קורס (שם ומספר) ולהוסיף אותו לרשימת הקורסים.
2	הוספת סטודנט חדש	לאחר הקשה, המשתמש יתבקש להזין את פרטי הסטודנט (שם, מספר, מגדר וגיל) ולהוסיף אותו לרשימת הסטודנטים.
3	רישום סטודנט לקורס	לאחר הקשה, המשתמש יתבקש להזין את מספר הקורס ומספר הסטודנט. בתאם לכך יש לעדכן רשימת סטודנטים בקורס הזה וגם כן את רשימת הקורסים לסטודנט, כמו כן יש לעדכן את מערך הציונים ולאתחל את הציון בערך 1- שיועדכן בהמשך ע"י אופציה 4.
4	מתן ציון לסטודנט בקורס	לאחר הקשה, המשתמש יתבקש להזין את מספר הקורס ואת תעודת הזהות של הסטודנט והציון, בהתאם לכך המערכת מעדכנת לסטודנט ציון בקורס הנבחר.

5	הדפסת פרטי קורס	לאחר הקשה, המשתמש יתבקש להזין את מספר הקורס והמערכת תדפיס את כל הפרטים של הקורס, כולל לפרטי הסטודנטים הרשומים (כולל ציון או הודעה במידה ולא קיים עדיין)
6	הדפסת פרטי סטודנט	לאחר ההקשה, המשתמש יתבקש להזין את תעודת הזהות והמערכת תדפיס את פרטי הסטודנט כולל שמות הקורסים שהוא רשום אליהם או הודעה מתאימה במידה והסטודנט לא נרשם עדיין לאף קורס.
7	יציאה	יציאה מהתפריט

ה-main מייצר אובייקט מטיפוס תפריט ומריץ את המתודה Run.

Driver.cpp

```
#include "Course.h"
#include "Student.h"
#include "Menu.h"
int main() {
    Menu menu;
    menu.Run();
    return 0;
}
```

עבודה פוריה !!!