

תרגיל בית מס' 3 – Operator Overloading

תאריך הגשה: 23:55 01/01/2017

הקדמה

בעבודה זו ניצור מבנה נתונים אשר יתחזיק קבוצות של סטודנטים וננישם את נושא דריסת אופרטורים. ניצור מחלקה אשר תייצג סטודנט ובנוסף מחלקה אשר תייצג קבוצת סטודנטים, שעליה ניתן יהיה לבצע פעולות שונות.

מחלקת Student

המחלקה תייצג סטודנט עם מאפייניו הבסיסיים. מלבד שדות פרטיים שיוגדרו על ידיכם, סטודנט ייוצג ע"י השדות הבאים:

- מס' מזהה (int)
- שם (char *)
- פקולטה - enum בעל האפשרויות:
 - ENGINEERING ○
 - MEDICINE ○
 - HUMANITIES ○
 - MANAGEMENT ○
 - GENERAL ○

המחלקה צריכה לתמוך בבנאי ללא פרמטרים, בנאי עם שלושה פרמטרים ובנאי העתקה. כמו כן, על המחלקה לתמוך באופרטור השמה וב-Destructor. בנוסף, עליכם לתמוך בשיטות:

`int getID(), char* getName(), Faculty getFaculty()` שיחזירו את השדות הפרטיים של

הסטודנט.

בנאי ללא פרמטרים:

ערכים דיפולטיבים:

- מס' מזהה: מס' רץ החל מ-100
- שם פרטי: "NONE"
- פקולטה: GENERAL

372-1-2102 – תכנות מתקדם (סמסטר סתיו, תשע"ז)

לדוגמא:

```
Student s1;  
Student s2;  
Student s3;  
cout << s1 << ", " << s2 << ", " << s3 << endl;
```

יודפס הפלט:

```
{100, NONE, General},{101, NONE, General},{102, NONE, General}
```

יש ליישם בנאי העתקה ואופרטור השמה כפי שנלמד בשיעור.

הדפסת סטודנט תתבצע באופן הבא: {ID, Name, Faculty}.

לדוגמא: {202, John Smith, Engineering}

על המחלקה להיות מוגדרת כראוי בקבצים Student.h ו- Student.cpp. יש לשים לב להגדרות של כל אופרטור ולבצע העמסה מדוייקת (מתי const, reference וכדומה).

מחלקת StudentSet

המחלקה תייצג קבוצת סטודנטים. מאחר ומדובר בקבוצה (Set), לא ניתן שבאותה קבוצה יהיו שני סטודנטים בעלי אותו ID ואין חשיבות לסדר. במידה וקיים ניסיון להוסיף מס' סטודנטים בעלי אותו ת"ז לאותו StudentSet, יתווסף רק הראשון ומהיתר יש להתעלם (אין צורך לתת הודעת שגיאה).

על המחלקה לתמוך באופרטורים הבאים:

1. איחוד קבוצות (+, +=)
2. חיתוך קבוצות (/=, /)
3. הפרש בין קבוצות (-, -=)
4. החזרת מס' הסטודנטים בקבוצה (+)
5. קבלת ערך לפי אינדקס ([])
6. השוואה בין קבוצות (!=, ==)

372-1-2102 – תכנות מתקדם (סמסטר סתיו, תשע"ז)

עליכם ליישם שני בנאים:

דיפולטיבי ללא פרמטרים:

יצור קבוצה ריקה, ללא אף סטודנט.

בנאי עם שני פרמטרים:

הבנאי יקבל מערך של סטודנטים שייצגו את רשימת הסטודנטים הראשונית בקבוצה ואת גודל המערך. שימו לב – אין מניעה שבמערך הקלט יהיו מס' סטודנטים זהים ולכן עליכם לוודא שלאחר יצירת הקבוצה, כל סטודנט בה הוא יחודי.

דוגמא:

מערך הקלט (באדום – הכפילויות):

202	234	204	202	403	234
John Smith	John Doe	Dana Cohen	John Smith	Liron Levi	John Doe
ENGINEERING	MEDICINE	MEDICINE	ENGINEERING	GENERAL	MEDICINE

הקבוצה שתקבל:

202	234	204	403
John Smith	John Doe	Dana Cohen	Liron Levi
ENGINEERING	MEDICINE	MEDICINE	GENERAL

בנוסף, עליכם לתמוך בבנאי העתקה, Destructor ואופרטור השמה (=) כפי שנלמד בשיעור.

תמיכה באופרטורים:

שימו לב – לשם הנוחות בלבד, בדוגמאות הבאות סטודנט ייוצג ע"י ID בלבד ולא יצוינו שאר השדות.

1. איחוד (+, +=)

עליכם לתמוך באיחוד של קבוצות. כאשר עליכם לוודא שאם סטודנט קיים בשתי הקבוצות, הוא לא יופיע פעמיים בקבוצה החדשה.

לדוגמא, איחוד של הקבוצות {202, 654} ו- {202, 355} תחזיר את הקבוצה {202, 355, 654}

372-1-2102 – תכנות מתקדם (סמסטר סתיו, תשע"ז)

2. חיתוך (/), (=)

הפעולה תחזיר קבוצה המכילה רק את הסטודנטים הקיימים בשתי הקבוצות שביניהן מתבצע החיתוך.

לדוגמא, חיתוך של הקבוצות {202, 234, 654} ו- {202, 355} תחזיר את הקבוצה {202}

3. הפרש (-), (=)

הפעולה תחזיר קבוצה המכילה רק הסטודנטים הקיימים בקבוצה הראשונה ואינם מופיעים בשניה. לדוגמא, חיסור של הקבוצות {202, 234, 654} ו- {202, 355} תחזיר את הקבוצה {234, 654}

4. החזרת מס' סטודנטים בקבוצה (אופרטור אונארי +)

נרצה לקבל את מס' הסטודנטים בקבוצה מסוימת ע"פ האופרטור +. לדוגמא, אם `studentSetExample = {101, 102, 103}` אז :

```
int x;
```

```
x = +studentSetExample; // x = 3
```

בנוסף לכך, יש לממש פונקצית `getSize()` המחזירה את גודל ה-`set`, באותו אופן כמו האופרטור מעלה.

```
int x;
```

```
x = studentSetExample.getSize(); // x = 3
```

5. אופרטור אינדקס ([])

הפעולה תחזיר את הסטודנט שנמצא באינדקס המצוין בקבוצה. מאחר ולסדר אין חשיבות בקבוצה, פעולה זו תשמש בעיקר למעבר איטרטיבי על קבוצה מסוימת. ניתן להניח שלא ינתן אינדקס גדול מגודל הקבוצה. שימו לב! עליכם לתמוך באופרטור משני הכיוונים! לדוגמא, אם `studentSetExample = {101, 102, 103}` אז :

```
Student x;
```

```
x = studentSetExample[0];
```

```
// x = {101, Name, Faculty}, studentSetExample = {101, 102, 103}
```

```
Student y(200, Yossi, GENERAL);
```

```
studentSetExample[0] = y;
```

```
// x = {101, Name, Faculty}, studentSetExample = {200, 102, 103}
```

372-1-2102 – תכנות מתקדם (סמסטר סתיו, תשע"ז)

6. השוואה בין קבוצות (=, !=)
עבור האופרטור "==" הפעולה תחזיר true במידה ובשתי הקבוצות זהות בגודלן ומכילות את אותם הסטודנטים בדיוק (לאו דווקא באותו הסדר), false אחרת. עבור האופרטור "!=" הפעולה תחזיר true אם יש שוני בין הקבוצות.

הדפסה של קבוצה:

קבוצה תודפס למסך באופן הבא:

```
{Number of Students, {Student1}, {Student2} ,...}
```

לדוגמא:

```
{2, {202, John Smith, Engineering}, {234, John Doe, Medicine}}
```

בנוסף, עליכם לתמוך בפונקציית PrintAll, אשר מדפיסה למסך את כל הקבוצות הקיימות במערכת, כאשר בשורה הראשונה מופיע מס' הקבוצות במערך ולאחר מכן כל קבוצה תודפס בשורה נפרדת.

לדוגמא:

3

```
{3, {202, John Smith, Engineering}, {234, John Doe, Medicine}, {204, Dana  
Cohen, Medicine}}
```

```
{1, {202, John Smith, Engineering}}
```

```
{2, {202, John Smith, Engineering}, {234, John Doe, Medicine}}
```

דליפות זכרון

העבודה תבדק לזיהוי דליפות זכרון. ודאו כי אתם מוחקים את כל הערכים שיש למחוק ולא משאירים זכרון ללא מצביע אליו בשום שלב של התוכנית. בכדי לזהות דליפות זכרון העזרו בתוכנה [Valgrind](#) כפי שמתואר [כאן](#).

קובץ Main

לעבודה זו מצורף קובץ Main קטן לדוגמא, ע"מ לוודא שה- Makefile שלכם ושמות הפונקציות הן כנדרש. במהלך הבדיקה נחליף את קובץ ה- Main לקובץ מורכב יותר.

372-1-2102 – תכנות מתקדם (סמסטר סתיו, תשע"ז)

הנחיות הגשה:

ההגשה בזוגות בלבד.

Makefile - קובץ Make ידני שיאחד את כל הקבצים שלכם, בתוספת קובץ **Main.cpp** המכיל פונקציה אחת בלבד : main. העבודה תיבדק עם קובץ Main.cpp חדש, שייכתב על ידינו.

קובץ ההרצה צריך להקרא בשם myStudentSet (באופן מדויק, ללא תוספות).

אין ליצור את קובץ ה-Make בעזרת editor אוטומטי כלשהו, אלא לבנות אותו בעצמכם.

התרגיל יעבור בדיקה פרונטלית. כדי שהתוכנית תעבור בשלום את הבדיקה, אין לתת שמות יצירתיים בעברית לקבצים שלכם, אין לשנות את סדר או אופי הפעולות שהוגדרו לכם. אין צורך לבצע בדיקת תקינות קלט במקומות שלא הוגדרו לכם.

הגישו תיקייה המכילה את כל קבצי ה-h וקבצי ה-cpp שלכם, בתוספת קובץ ה-makefile שבניתם.

את הקבצים שימו בתוך תיקייה ששמה מורכב ממספרי הזהות של המגישים המופרדים עם " _ " בלבד, וקבצו את התיקייה לקובץ zip יחיד (לדוגמא, 012345678_876543210).

יש להגיש את הגרסה הסופית של העבודה דרך אחד המגישים בלבד למערכת ההגשה של הקורס בכתובת:

<https://subsys.ise.bgu.ac.il/submission/login.aspx>

שאלות לגבי העבודה ניתן יהיה לשאול בפורום מיוחד שיפתח באתר הקורס.

בהצלחה !!!