

1. ממשו את המחלקה `myset<T>` לפי הגדרת טיפוס הנתונים המופשט `Set[G]` מהתרגיל הקודם תוך התחשבות בנקודות הבאות:

- המחלקה תשתמש בטיפוס סטנדרטי בשפת C++ (כגון `set<T>`, `vector<T>` ודומיהם), כלומר יש להגדיר שדה `private` בשם `rep` מהטיפוס הסטנדרטי.
- יש לממש את כל המתודות שהוגדרו כפונקציות של טיפוס הנתונים המופשט בשמן ולפי חתימתן תוך שמירה על כללי ההמרה מפונקציות למתודות (פקודות יהפכו ל `void`, הטיפוס `SET[G]` יעלם). עבור `emptyset` ראו התייחסות נפרדת בסעיף 3.
- יש לממש את פונקציית ה-`friend` המעמיסה את האופרטור `<<`:  

```
void operator<<(ostream &, myset<T>)
```

 שמדפיסה למסך קבוצה בצורה 'טבעית'. לדוגמא:  

```
myset<int> iset;
iset.extend(5);
cout << iset;
```
- התנהגות הפונקציות צריכה להיות תואמת לאקסיומות. יש לספק `main` עם דוגמאות שימוש המדגימות תאימות לכל אחת מ-19 האקסיומות. העזרו בדוגמאות באופרטור ההדפסה שהגדרתם. שימו לב: חלק מתכונות `set` הסטנדרטית "מפתיעות", לדוגמא: האם הקבוצות `{1,2}` ו-`{2,1}` שוות (אופרטור `==`) או לא?
- יש להפריד בין ממשק, מימוש ושימוש ע"י שימוש בשלושה קבצים שונים.

2. נסו ליצור קבוצות של טיפוסים פשוטים (`int`, `float`, `char`). נסו ליצור קבוצות של מחלקות שהגדרתם בעצמכם (`Point`, `Stack`) מדוע המחלקות שהגדרנו בעצמנו יוצרות טעויות קומפילציה? מה חסר בהן? (רמז: נסו להגדיר את האופרטור `<` עבור המחלקות הבעייתיות) מה זה אומר על הטיפוס `T` של `set<T>`? הגישו 2 דוגמאות הרצה של יצירה ושימוש בקבוצות של טיפוסים לא טריויאליים.

3. הפונקציה היוצרת קבוצה ריקה הוגדרה בתרגיל הקודם בתור emptyset ואולם ב C++ כדי ליצור מחלקה עלינו להשתמש בבנאי של המחלקה ששמו כשם המחלקה myset<T>. בסעיף זה נרצה לאסור את השימוש ישירות בבנאי וליצור קבוצות רק בעזרת הפונקציה emptyset לדוגמא:

```
1. myset<int> s; // Compilation Error !
2. myset<int> *s = new myset<int>; // Compilation Error !
3. myset<int> *s = emptyset(); // Good !
```

- א. איך נאסור את ביצוע השורה הראשונה והשנייה (התקינות מבחינה תחבירית)?
- ב. איך נגדיר את emptyset() כדי שתפעל כנדרש? מדוע אי אפשר לממש emptyset() כנדרש שתפעל כהלכה עבור השורה השלישית?
- ג. תקנו את שורה 3 וממשו את emptyset כך שתפעל כהלכה עבורה. (רמז: שנו את חתימת הפונקציה emptyset והגדירו אותה כפונקציה friend או הגדירו את emptyset כפונקציה סטטית).

4. נתונה התוכנית הבאה:

```
#include <iostream>
```

```
int main()
{
    cout<<"Hello, world";
}
```

התוכנית תדפיס: **Hello, world**  
עדכן את התוכנית על מנת שתדפיס: **"Hello, world"** (עם המרכאות!). מבלי לשנות את main() בשום צורה שהיא. אין להשתמש בחפיפת אופרטורים (למשל ב - <<).

5. כתוב תכנית ללא לולאות וללא רקורסיה שתדפיס על המסך 50 פעם את שמך.

6. אילו בנאים ו/או אופרטורים רצים ומדוע עבור כל אחת מהשורות הבאות:

```
1. vector<Point> v1(5);
2. vector<Point> v2[5];
3. Point v3[5];
```

הוכח את טענתך ע"י דוגמאות הרצה.