

## תרגול 5

### חלק א: static, delete, <<

1. הגדירו מחלקת Person שנתונה הם:
  - a. שם האדם (char \*) וגילו (int).
  - b. Id (int) – מספר רץ ייחודי לכל אדם אשר יתחיל מ 1000.
  - c. הגיל (int) שבו אדם יכול להוציא רישיון נהיגה.הערה: תכונת גיל הרישיון משותפת לכל האנשים ומתעדכנת מדי פעם.
  - d. אדם נחשב לבוגר בגיל 18. יש לשמור נתון זה. נתון זה זהה לכל בני האדם ואינו יכול להשתנות. יש לתת גישה ישירה לנתון זה.
2. אדם מוגדר כאדם חי במידה ונוצר אובייקט ממנו. ברגע שהאובייקט משתחרר, האם אינו חי יותר. כתבו פונקציה המחזירה את מספר האנשים החיים ברגע נתון.
3. הגדירו בנאי המקבל שם וגיל. הגדירו בנאי העתקה
4. הגדירו מתודת set שתעדכן את הגיל בו ניתן להוציא רישיון נהיגה.
5. ממשו את << שתציג את נתוני האדם וכן האם הוא יכול או לא יכול להוציא רישיון נהיגה.  
תזכורת: `friend ostream& operator<<(ostream& os, const Person& obj);`
6. אין לאפשר להשתמש באופרטור השמה. יש לממש את החסימה בעזרת delete.
7. ב main צרו 2 מצביעים לאנשים והקצו אותם, הדפיסו את פרטיהם, והדפיסו את מספר האנשים החיים.
8. שחררו אדם אחד, ובדקו כמה אנשים חיים כרגע.
9. שחררו את האדם השני ובדקו כמה אנשים חיים כרגע.
10. צרו 2 אובייקטים בבולוק, הדפיסו את הנתונים ובדקו כמה אנשים חיים.
11. לאחר מהיציאה מהבולוק, בדקו שוב כמה אנשים חיים.

### חלק ב: mutable, const function

12. הגדירו פונקציה חדשה ב Person בשם canDrive (וודאו שהפונקציה מצהירה שאינה משנה את האובייקט) אשר תחזיר אמת אם האדם רשאי לנהוג, אחרת שקר. אדם רשאי לנהוג אם גילו מעל או שווה לתכונת גיל הרישיון.
13. הגדירו: `p3("Dan", 20) const Person` והדפיסו אם הוא רשאי לנהוג בעזרת קריאה ל canDrive.
14. הגדירו משתנה פרטי חדש ב Person אשר סופר כמה פעמים נקראה הפונקציה canDrive. הוסיפו שורת קוד ל canDrive אשר מבצעת ++ למשתנה זה.
15. שנו את גיל תכונת רשאי לנהוג ל 1000, ובדקו ש Dan אינו יכול לנהוג.

### חלק ג: move ctor

16. הגדירו בנאי העברה ובדקו שהוא עובד (העזרו ב std::move).