**פתרון חלק תאורטי:**

**1.** האם ניתן ליצור מספר אופרטורים =+ השונים ביניהם בטיפוסי הפרמטרים?

כן, ניתן ליצור מספר אופרטורים =+ השונים ביניהם בטיפוסי הפרמטרים, למשל int,char,string וכו'.

כדוגמא ניקח את מעבדה 6, השתמשנו במחלקה Rational:

**Rational& operator += (int);**

**Rational& operator += (double);**

נניח שהיינו ממשים את האופרטורים האלו, אז כאשר היינו משתמשים באופרטור =+ במקרה של int או במקרה של double הקומפיילר היה יודע באופן אוטומטי להשתמש באופרטור עם ערך הקבלה הרלוונטי. באופן כללי הקומפיילר יידע "לבחור" באופרטור =+ הרלוונטי עבור אותו טיפוס (int,char,double,string…).

**2.** האם הטענה נכונה או לא נכונה: בשפת C++ ניתן ליצור משתני static רק במסגרת המחלקה.

טענה זו אינה נכונה, בC++ קיימים משתנים סטטיים מחוץ למחלקה, לדוגמא משתנה סטטי שמוגדר בתוך פונקציה (המשתנה הסטטי שאנו מכירים מ-C). בנוסף לכך בC++ ניתן להגדיר משתנה סטטי בתוך מחלקה, "אורך החיים" שלו הוא לכל אורך חיי התוכנית, הנגישות אליו היא מכל מתודה של המחלקה, והוא משותף לכל האובייקטים ממחלקה זו.

**3.** בתוכנית ספציפית רוצים לבדוק כמה פעמים הופעלה מתודה. איך ניתן ליישם את זה?

במקרה זה נוכל להשתמש במשתנה סטטי, שבכל פעם שיווצר אובייקט חדש יגדיל את עצמו ב1.

כדוגמא ניקח את מעבדה 5, השתמשנו במשתנה סטטי כמספר רץ, שממספר כל סטודנט:

בקובץ הh הגדרנו משתנה סטטי **static int studentNumber**.

בקובץ הcpp איפסנו את משתנה הסטטי: int Student::studentNumber = 0;

לאחר מכן ביצענו פעולה שמגדילה את studentNumber ב1 בתוך הבנאי, כלומר בכל פעם שנוצר אובייקט (סטודנט) חדש אז הבנאי מופעל ואז studentNumber גדל ב1:

Student::Student() : studentNumberCounter(**studentNumber++**), studentName("None"), arr(NULL) {};

**4.** במידה ורוצים למנוע שימוש באופרטור השמה, איך ניתן לעשות זאת?

במידה ונרצה למנוע שימוש באופרטור השמה בגלל ש"המתנה" מהקומפיילר לא תיהיה רלוונטית עבורינו, כלומר אופרטור ההשמה הדיפולטיבי מבצע העתקה רדודה, ובמקרים מסויימים דבר זה יכול לגרום לבעיית ההצבעה הכפולה. פשוט ניצור אופרטור השמה ריק ב private כדי למנוע שימוש.

**5.** מהו ההבדל בין בנאי מעתיק לבין אופרטור השמה?

בנאי מעתיק מאתחל זיכרון לא מאותחל, לעומת זאת אופרטור השמה עובד עם אובייקט קיים ובעל ערכים.

**6.** נתקן את קטע התוכנית בכך שבמקום this[i] = i + 1 נרשום this->num[i] = i + 1

**מאור עטר - 318301231**