

# תרגיל 3

## נושא: העמסת אופרטורים

מתרגלת אחראית – ספיר לקסמן

### הוראות כלליות

יש להגיש למערכת קובץ אחד – str.cpp

- יש לקרוא היטב את התרגיל לפני שמתחילים לעבוד ולוודא שהבנתם את כל הפרטים.
- יש לקרוא את קובץ "דרישות סגנון תכנות" היטב לפני שמתחילים לעבוד, ולמלא אחר הדרישות.
- ההגשה הינה ביחידים בלבד
- **שימו לב שהבדיקה הינה אוטומטית! תכנית שלא מתקמפלת או מתקמפלת עם אזהרות תקבל ציון 0. שימו לב – לא יתקבלו ערעורים על שגיאות/אזהרות קמפול. אנא ודאו כי אין בעיות מהסוג הזה בתרגילים אותם אתם מגישים!**
- יש להקפיד על הגדרת public ו-private עבור משתני המחלקה והמתודות. לא לשכוח להוסיף הערות בפורמט הנכון (שם ופרטים בהתחלה, הערה לפני כל מתודה, הערה לכל מחלקה, הערות בקוד עצמו). תרגיל שיוגש ללא הקפדה כל כתיבת ההערות לפי הפורמט עלול לקבל הורדת ציון של 15 נק'.
- במסגרת התרגיל תקבלו קבצי H עם המתודות שעליכם ליישם. עליכם ליישם את המתודות המתאימות בקבצי ה-CPP התואמים וכמו כן להגדיר את משתני המחלקה החסרים בקבצי ה-H שניתנו.
- **אין לשנות חתימה של מתודות בשום צורה!**
- **אין לבצע פעולות קלט/פלט בשום פנים ואופן!**
- **אין לשנות קוד קיים! ניתן להוסיף קוד לפונקציה כתובה אך אין למחוק את מה שכתוב בה!**
- לאחר כתיבת התכנית מומלץ לעבור שוב על התרגיל ולוודא שמילאתם את כל ההוראות – בכל הסעיפים ובכל ההערות.

## תיאור התרגיל

בתרגיל זה תממשו מחלקת `str` שהינה אובייקט מחרוזת ואת הפעולות הנדרשות עליה.

**שימו לב,** מומלץ לעיין ב-`mainTrain` טרם כתיבת הקוד על מנת להבין מה מצופה מכם.

אין צורך לזרוק חריגות במהלך התוכנית.

עליכם לממש את המתודות והאופרטורים הבאים:

(1) בנאי שמתאחל את המחרוזת למחרוזת "none".

```
Str();
```

(2) הרסן

```
~Str();
```

(3) בנאי העתקה

```
Str(const Str & other);
```

(4) בנאי שמקבל מחרוזת `* char` ומאתחל את המשתנה של המחלקה בהתאם (כולל הקצאה דינמית)

```
Str(const char * str);
```

(5) אופרטור להשוואת שתי מחרוזות

```
bool operator==(const Str & other) const;
```

(6) אופרטור להשוואת שתי מחרוזות

```
bool operator!=(const Str & other) const;
```

(7) משווה שתי מחרוזות לקסיקוגרפית

```
bool operator>(const Str & other) const;
```

(8) משווה שתי מחרוזות לקסיקוגרפית

```
bool operator<(const Str & other) const;
```

(9) אופרטור השמת אובייקט

```
Str & operator=(const Str & other);
```

(10) אופרטור השמת מחרוזת

```
Str & operator=(const char * str);
```

(11) מחזיר את התו במקום האינדקס

```
char & operator[](int index) const;
```

(12) מקדם אסקית כל תו ב-1

```
Str & operator++();
```

(13) מקדם אסקית כל תו ב-1

```
Str operator++(int);
```

(14) אופרטור קאסטינג – מחזיר את אורך המחרוזת

```
operator int() const;
```

(15) מחזיר את המיקום הראשון של התו `ch` במחרוזת או (-1) אם לא מוצא  
`int operator()(char ch) const;`

(16) משרשר בין מחרוזות

```
Str operator+(const Str & other) const;
```

(17) משרשר בין מחרוזות

```
friend Str operator+(const char * str, const Str & other);
```

(18) משרשר את המחרוזת שמקבל כפרמטר, num פעמים  
`friend Str operator*(int num, const Str & other);`

(19) משרשר בין מחרוזות  
`Str & operator+=(const Str & other);`