

## תכנות מתקדם – 150024

### תרגיל בית מספר 4

#### אופרטורים, friend, קבצים

#### שים/י לב:

- הקפד/י על קריאות התכנית ועל עימוד (Indentation).
- הקפד/י לבצע בדיוק את הנדרש בכל שאלה.
- בכל אחת מהשאלות יש להגדיר כל מחלקה ב 2 קבצים נפרדים. קובץ H וקובץ CPP.
- בכל אחת מהשאלות יש להגדיר פונקציות עזר במידת הצורך עבור קריאות התכנית.
- יש להגיש את התרגיל על פי ההנחיות להגשת תרגילים (המופיע באתר הקורס) וביניהם:
  - השתמש/י בשמות משמעותיים עבור המשתנים.
  - יש לתעד את התכנית גם עבור פונקציות אותם הנך מגדיר/ה וכן על תנאים ולולאות וקטעי קוד מורכבים, ובנוסף, **דוגמת הרצה לכל תכנית בסוף הקובץ!**
  - הגשה יחידנית - אין להגיש בזוגות.**

**הערה חשובה:** לכל תרגיל בית מוגדר שבוע אחד בלבד להגשה, אלא אם כן קיבלת הוראה אחרת מהמרצה שלך. תיבות ההגשה הפתוחות לא מהוות היתר להגשה באיחור.

#### שלב 1:

#### מחלקת Worker

- הגדירו מחלקה בשם **Worker** לייצוג עובד בחברה. השדות במחלקה יהיו:
  - id - מספר מזהה** של עובד (מספר שלם, לא מתחיל ב-0).
  - name - שם פרטי** של העובד (ללא רווחים)
  - שימו לב:** בגלל שנמשור את השדה בקובץ **אז יש להגדיר כמערך של char**
  - salary - השכר החודשי** של העובד (מספר ממשי גדול מ 0)
- הערות:** על אף שכל השדות פרטיים. במחלקה זו **אסור להוסיף מתודות set/get** לאף אחד מהשדות. בנוסף, יש להניח שכל ערכי השדות שיקלטו יהיו תקינים. **אין צורך לבדוק תקינות קלט.**

#### ב. ערכי ברירת המחדל של המחלקה יהיו:

- id = 0
- name=""
- salary=0.0

ניתן להציב את ערכי ברירת המחדל בהגדרת השדות בקובץ **h** או להגדיר כבנאי ברירת מחדל default constructor. שים לב: אין להוסיף למחלקה אף בנאי נוסף.

#### ג. הוסף למחלקה Worker את המתודות הציבוריות הבאות:

- operator+=** מקבל כפרמטר מספר float שמייצג בונוס ומוסיף את הבונוס לשכר של העובד.
- הוסף את הפונקציות החברות הבאות (פונקציות גלובליות עם הרשאת גישה ל-private של Worker):

- operator==**

הפונקציה מקבלת כפרמטרים: עובד (worker) ומספר (int), ומחזירה true אם המספר זהה למספר הזהות של העובד, אחרת הפונקציה תחזיר false.  
שים לב - שהאופרטור == מקיים את חוק החילוף כלומר מתקיים Worker==num וגם num==Worker ולכן יש להגדיר 2 פונקציות עם החתימות הבאות:  
 bool operator==(int workerID, const Worker& worker)  
 bool operator==(const Worker& worker, int workerID)

#### • operator==

הפונקציה מקבלת כפרמטרים: עובד (worker) ומחרוזת (string), ומחזירה true אם המחרוזת זהה לשם של העובד, אחרת הפונקציה תחזיר false.  
שים לב - שהאופרטור == מקיים את חוק החילוף כלומר מתקיים Worker==str וגם str==Worker ולכן יש להגדיר 2 פונקציות עם החתימות הבאות:  
 bool operator==(string str, const Worker& worker)  
 bool operator==(const Worker& worker, string str)

- **operator>>** עבור קלט של נתוני העובד. יש לקלוט לפי הסדר: מספר זהות, שם ומשכורת.
- **operator<<** עבור פלט של נתוני העובד. ההדפסה תדפיס שורה אחת את מספר הזהות, השם והמשכורת (עם רווח בודד בניהם) ואחר כך לרדת שורה.

ה. כתוב תוכנית ראשית שבודקת את נכונות הקוד שכתבת (אין צורך להגיש תוכנית זו)

## שלב 2

### מחלקת WorkersFile:

הגדירו מחלקה בשם **WorkersFile** לניהול קובץ טקסט פיזי של עובדים. בקובץ ישמרו נתוני העובדים (מטיפוס Worker) והמחלקה תאפשר לבצע את הפעולות הבאות:

- לפתוח את הקובץ לקריאה או לכתיבה.
- לסגור את הקובץ.
- לקלוט את נתוני כל העובדים לתוך הקובץ (אופרטור >>).
- להדפיס את פרטי של כל העובדים ששמורים בקובץ (אופרטור <<).
- לחפש ולהדפיס נתונים של עובד מסוים לפי מספר הזהות שלו (operator[](int)).
- לחפש ולהדפיס נתונים של עובד מסוים לפי שמו (operator[](string)).
- להוסיף בונים לכל עובד (operator+=).

ניתן להניח שאין שני עובדים עם אותו שם או עם אותו ת.ז. אין צורך לבדוק תקינות הקלט.

**דוגמא לתוכן קובץ טקסט בשם test.txt שמכיל פרטים (מספר מזהה, שם פרטי, שם חודשי) של 3 עובדים:**

כל עובד מופיע בשורה נפרדת. עם רווח בין השדות.

```
1111 dan 6500
11 gad 2200.55
111 shimon 3000
```

א. הגדר מחלקה בשם **WorkersFile** שתכלול את השדות הפרטיים באים:

- **string fileName** - מייצג את שם קובץ הטקסט המנוהל
- **fstream iofile** - אובייקט קישור לקובץ הטקסט הפיזי. בתחילת כל פעולה, יפתח הקובץ הפיזי מחדש לקריאה ולכתיבה על פי הצורך. ובסיום על פעולה, ייסגר הקובץ.

**הערות:** על אף שכל השדות פרטיים. במחלקה זו **אסור להוסיף מתודות set/get** לאף אחד מהשדות. בנוסף, יש להניח שכל ערכי השדות שיקלטו יהיו תקינים. אין צורך לבדוק תקינות קלט.

ב. הוסף למחלקה את מתודות העזר הפרטיות הבאות:

- **openFileForWriting** – מתודה בוליאנית. שתפקידה לפתוח את הקובץ **לכתיבה**. אם פתיחת הקובץ נכשלה, יש גם להדפיס **ERROR** ולהחזיר **false** אחרת הפונקציה תחזיר **true**.

ניתן להשתמש בשלד הבא על מנת לכתוב את המתודה `openFileForWriting`  
- הוסף את הקוד בשורות החסרות:

```
bool WorkersFile::openNewFileForWriting()
{
    ifstream file(fileName, ios::out); // לפתוח את הקובץ

    if (!file.is_open()) // טיפול בבעיה אם זה לא נתפח
    {
        // _____
        // _____
    }

    return true;
}
```

- **openFileForReading** – מתודה בוליאנית. שתפקידה לפתוח את הקובץ **לקריאה**. אם פתיחת הקובץ נכשלה, אז יש גם להדפיס **ERROR** וגם להחזיר **false**. אחרת הפונקציה תחזיר **true**.
- **closeFile** – מתודה פרטית. שתפקידה לסגור את הקובץ הפתוח (דרך השדה `iofile`).

**הערה:** ניתן להוסיף מתודות עזר פרטיות על פי הצורך.

ג. הוסף/י למחלקה **WorkersFile** בנאי:

**constructor** – בנאי המקבל פרמטר אחד מטיפוס `string` שהוא שם קובץ הטקסט ויבצע את הפעולות הבאות:

- `fileName` יאותחל בשם הקובץ שהתקבל כמחרוזת.
- לאחר מכן יש לייצר קובץ פיזי ריק, בשם שהתקבל כפרמטר ומיד לסגור אותו. (ניתן לעשות זאת ע"י קריאה למתודה הפרטית `openFileForWriting` ולאחר מכן למתודה הפרטית `closeFile`).

ד. הוסף למחלקה **WorkersFile** את הגדרת המתודות **החברות הגלובליות** הבאות:

- **operator>>** המאפשר לקלוט את נתוני **כל** העובדים לתוך הקובץ. לדוגמא:

```
WorkerFile wf("filename");
cin>>wf;
```

**operator>>** של `WorkerFile` משתמש בפונקציה `operator>>` שהגדרת עבור `Worker`. קליטת הנתונים תיעצר כאשר ייקלט **עובד ריק** (מספר זהות שווה 0, יש להשתמש `operator==` שהגדרת עבור `Worker`)

- **operator<<** המאפשר להדפיס את נתוני **כל** העובדים ששמורים בקובץ. לדוגמא:

```
cout<<wf;
```

<<operator של WorkerFile משתמש בפונקציה <<operator שהגדרת עבור Worker. שימו לב: בגלל שבתוך הפונקציה נצטרך לטפל בקובץ (נעדכן לפחות אחד מהשדות של המחלקה) אז יש צורך לשלוח לפונקציה את המופע של המחלקה כ-cbr (& עם)

ה. הוסיף/י למחלקה WorkersFile את מתודות המחלקה הבאות בלבד:  
**הערה:** מתוך הגוף של כל אחת מהמתודות יש להיעזר במתודות העזר הפרטיות על מנת לפתוח את הקובץ לקריאה ו/או לכתיבה, לבצע את הפעולות הנדרשות במתודה, ולאחר כל קריאה/כתיבה יש לסגור את הקובץ.

- **operator[]** (אופרטור סוגריים מרובעות) שפועל על קובץ ה WorkersFile הנוכחי. מקבל שם של עובד, כפרמטר ומחזיר משתנה מטיפוס Worker שמכיל את פרטי העובד שנמצא בקובץ עם אותו שם. במידה שהעובד לא קיים אז יחזיר Worker עם נתוני ברירת מחדל (ת.ז. 0 שם מחרוזת ריקה ומשכורת של 0.0)
- ```
wf["David"]
```

- **operator[]** (אופרטור סוגריים מרובעות) שפועל על קובץ ה WorkersFile הנוכחי. מקבל ת.ז. של עובד, כפרמטר ומחזיר משתנה מטיפוס Worker שמכיל את פרטי העובד שנמצא בקובץ עם אותו ת.ז. במידה שהעובד לא קיים אז יחזיר Worker עם נתוני ברירת מחדל (ת.ז. 0 שם מחרוזת ריקה ומשכורת של 0.0)

```
wf[1234]
```

שימו לב: בגלל שהפונקציות של operator[] מייצרים ומחזירים Worker חדש לפי נתונים שיש בקובץ (ולא רק מחזירים כתובת של משהו ששמור במערך שכבר נמצא בזיכרון) אז יש להחזיר Worker בלי & (cbv)

- **operator+=** שפועל על קובץ ה WorkersFile הנוכחי. מקבל מספר float שמייצג בונוס לכל העובדים, ומעדכן בקובץ את המשכורות של כל העובדים בהתאם.

נתון במודל קובץ בשם Main.cpp המכיל תוכנית הראשית (main) שמטרתה לבדוק נכונות המחלקה שכתבת, הורד את הקובץ ובדוק את תקינות התוכנית שלך (יש להגיש את התוכנית הראשית ביחד עם הקוד שכתבת).