סדנא ב- C++ – 150018

תרגיל בית מספר 6

ירושה

שים/י לב:

א. הקפד/י על קְרִיאוּת התכנית ועל עימוד (Indentation).
ב. הקפד/י לבצע בדיוק את הנדרש בכל שאלה.
ג. בכל אחת מהשאלות יש להגדיר פונקציות במידת הצורך עבור קריאות התכנית.
ד. 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.

שאלה מס' 1:

במכללת "עבדים היינו", קיימים שני סוגי עובדים:

- עובד במשרה מלאה (fulltime)
- ועובד במשרה חלקית (parttime)

עבור כל עובד רוצים לשמור את הנתונים הבאים:

- שם פרטי של העובד
- ת.ז של העובד
- מספר שנות הוותק של העובד בעבודה
- אחוז ההפרשה למס הכנסה

מדיניות התשלום במכללה היא פעם בחודש, לפי הנתונים הבאים:

- עבור עובד במשרה מלאה, החישוב הוא משכורתו השנתית לחלק ל-12.
  - לכן עבור עובד במשרה מלאה יש לשמור גם את משכורתו השנתית
- עבור עובד במשרה חלקית, החישוב הוא שעות העבודה בחודש כפול התשלום שהובטח לו לשעת עבודה.
  - לכן עבור עובד במשרה חלקית יש לשמור גם שעות עבודה ותשלום לשעה.

הנהלת המכללה החליטה לתת לכל עובד בונוס לחודש תשרי, לפי הנוסחה הבאה:

- במידה והעובד הוא בעל וותק בעבודה של 5 שנים או פחות, הבונוס יהיה 500 שקל.
- במידה והעובד בעל וותק בעבודה של יותר מ- 5 שנים, הבונוס יהיה 25% ממשכורתו החודשית לפני הבונוס.

לאור הנתונים לעיל:

א. הגדר/י מחלקה בסיסית בשם Employee, המייצגת עובד במכללת "עבדים היינו".
המחלקה תכלול את התכונות הבאות:

- name – שמו הפרטי של העובד (string)
- id – ת.ז של העובד (int)
- seniority – שנות וותק של העובד (int)
- pay – שכר לחודש (float)

וכן את המתודות הבאות:

- constructor עבור אתחול התכונות (את התכונה pay יש לאתחל ב-0)
<u>אין להגדיר במחלקה זו default-constructor</u>.
- get ו- set עבור הצבה ואחזור של תכונות המחלקה – במידת הצורך
- salaryAfterBonus – לעדכון המשכורת עבור בונוס תשרי
- אופרטור << עבור קליטת נתוני העובד (יש לקלוט את התכונות עפ"י סדר הגדרתן במחלקה) בפורמט הבא:

Enter employee details:

- אופרטור >> עבור הדפסת נתוני העובד בפורמט הבא:

Employee:
Employee ID:
Years Seniority:

ב. הגדר/י מחלקה בשם FullTime המייצגת עובד במשרה מלאה, היורשת ממחלקת Employee.
עליך להגדיר את המחלקה FullTime כך שניתן יהיה להפעיל עבור אובייקט מהמחלקה את המתודות הבאות:

- constructor – המקבל כפרמטרים את פרטי העובד כולל משכורת שנתית (שים/י לב, תכונה נוספת למחלקה זו) ומאתחל את התכונות עם ערכים אלו.
- default constructor – עם ערכי ברירת מחדל, מחרוזת ריקה ואפסים בהתאמה
- get ו- set עבור הצבה ואחזור של תכונות המחלקה – במידת הצורך
- salary – לחישוב משכורת חודשית של העובד
- salaryAfterBonus – לעדכון המשכורת עבור בונוס תשרי
- אופרטור << עבור קליטת נתוני העובד (יש לקלוט את התכונות עפ"י סדר הגדרתן במחלקה)

Enter employee details:

- אופרטור >> עבור הדפסת נתוני העובד בפורמט הבא:

Employee:
Employee ID:
Years Seniority:
Salary per Month:

שים/י לב! יש להימנע משכפול קוד ואין לכתוב במחלקה זו קוד שאינו נצרך. כלומר, במידה וקיימת מתודה או חלק מהקוד שבה אצל האב – אין לכתוב קטע קוד נוסף המבצע את אותה פעולה אצל הבן!!

ג. באופן דומה, הגדר/י מחלקה בשם PartTime המייצגת עובד במשרה חלקית, היורשת ממחלקת Employee.
עליך להגדיר את המחלקה PartTime כך שניתן יהיה להפעיל עבור אובייקט מהמחלקה את המתודות הבאות:

- constructor – המקבל כפרמטרים את פרטי העובד כולל שעות עבודה ותשלום לשעה (שים/י לב, תכונות נוספות למחלקה זו) ומאתחל את התכונות עם ערכים אלו.
- default constructor – עם ערכי ברירת מחדל, מחרוזת ריקה ואפסים בהתאמה
- get ו- set עבור הצבה ואחזור של תכונות המחלקה – במידת הצורך
- salary – לחישוב משכורת חודשית של העובד
- salaryAfterBonus – לעדכון המשכורת עבור בונוס תשרי
- אופרטור << עבור קליטת נתוני העובד (יש לקלוט את התכונות עפ"י סדר הגדרתן במחלקה)

Enter employee details:

- אופרטור >> עבור הדפסת נתוני העובד בפורמט הבא:

Employee:
Employee ID:
Years Seniority:
Hours:
Salary per Month:

שים/י לב! יש להימנע משכפול קוד ואין לכתוב במחלקה זו קוד שאינו נצרך. כלומר, במידה וקיימת מתודה או חלק מהקוד שבה אצל האב – אין לכתוב קטע קוד נוסף המבצע את אותה פעולה אצל הבן!!

בכל מתודה בה עלולה להופיעה שגיאה יש לזרוק exception ERROR.
שים/י לב, במקרה של שגיאה, למרות השגיאה, יש לקלוט את כל נתוני העובד.

נתונה התכנית הראשית הבאה, הבוחנת את נכונות המחלקה:

```
#include "FullTime.h"
#include "PartTime.h"
#include <iostream>
using namespace std;
int main()
{
	FullTime arrF[3];
	for (int i = 0; i < 3; i++)
	{
		try
		{
			cin >> arrF[i];
		}
		catch (const char* str)
		{
			cout << str << endl;
			i--;
		}
	}


	PartTime arrP[3];
	for (int i = 0; i < 3; i++)
	{
		try
		{
			cin >> arrP[i];
		}
		catch (const char* str)
		{
			cout << str << endl;
			i--;
		}
	}


	for (int i = 0; i < 3; i++)
	{
		cout << arrF[i];
		cout << "After Bonus: " << arrF[i].salaryAfterBonus() << endl;
	}


	for (int i = 0; i < 3; i++)
	{
		cout << arrP[i];
		cout << "After Bonus: " << arrP[i].salaryAfterBonus() << endl;
	}
	return 0;
}
```

<u>שאלה מס' 2:</u>

רשימה מעגלית RoundList, הינה רשימה לינארית שבה החוליה האחרונה מצביעה על החוליה הראשונה.
על רשימה מעגלית ניתן לבצע את כל המתודות המוגדרות עבור רשימה לינארית, ובנוסף ניתן לבצע את המתודות הבאות:

- הוספת איבר בסוף הרשימה addToEnd(int val). המתודה מקבלת כפרמטר מספר שלם val, ומוסיפה חוליה בסוף הרשימה עם הערך val.
- חיפוש ברשימה search(int n). המתודה מקבלת כפרמטר מספר שלם לא שלילי, n. המתודה מחזירה את ערכו של האיבר הנמצא במקום n ברשימה המעגלית. (שים/י לב: n יכול להיות מספר גדול יותר ממספר האיברים הקיימים ברשימה. מאחר והרשימה מעגלית יש להמשיך למנות את האיברים בשנית עד לאינדקס n. מספור הרשימה מתחיל מאינדקס 0). במידה והרשימה ריקה, המתודה מחזירה 1-.

הגדר/י את המחלקה RoundList כמחלקה היורשת מ-List (המחלקה List המגדירה רשימה לינארית שנלמדה בהרצאה). ממש/י את כל המתודות הנדרשות מרשימה מקושרת וגם את שתי המתודות הנוספות עבור רשימה מקושרת מעגלית.

הנחיות נוספות:

- אין להוסיף תכונות פרטיות חדשות למחלקה RoundList – התכונה היחידה במחלקה תהיה מצביע לראש הרשימה כפי שמוגדר במחלקה List הבסיסית.
- עליך להכריע אילו מתודות במחלקה List יש לדרוס במחלקה RoundList היורשת, ואילו מתודות אין צורך לדרוס.

נתונה התכנית הראשית הבאה, הבוחנת את נכונות המחלקה:

```
#include "RoundList.h"
#include <iostream>
using namespace std;

enum CHOICES{
	EXIT, ADD, ADD_TO_END, REMOVE_FIRST, SEARCH, CLEAR, EMPTY
};
```

```
int main(){

        RoundList ls1;
        int choice;
        cout << "Enter your choice: ";
        cin >> choice;
        while(choice != EXIT)
        {
                int num;
                switch(choice){
                        case ADD :      cout << "Enter 5 numbers: ";
                                        for(int i=0; i < 5; i++)
                                        {
                                                cin >> num;
                                                ls1.add(num);
                                        }
                                        break;

                        case ADD_TO_END :cout << "Enter 5 numbers: ";
                                                for(int i=0; i < 5; i++)
                                                {
                                                        cin >> num;
                                                        ls1.addToEnd(num);
                                                }
                                                break;

                        case REMOVE_FIRST : ls1.removeFirst();
                                                break;

                        case SEARCH: cout << "Enter a number: ";
                                        cin >> num;
                                        cout << ls1.search(num)<<endl
                                        break;

                        case CLEAR: ls1.clear();
                                        break;

                        case EMPTY: if(ls1.isEmpty())
                                                cout << "Empty"<<endl;
                                        else
                                                cout << "Not empty" << endl;
                                        break;

                        default: cout<< "ERROR!"<<endl;
```

```
            }
            cout << "Enter your choice: ";
            cin >> choice;
        }
        return 0;
}
```

דוגמא להרצת התכנית:

```
Enter your choice: 1
Enter 5 numbers: 1 2 3 4 5
Enter your choice: 2
Enter 5 numbers: 5 4 3 2 1
Enter your choice: 3
Enter your choice: 4
Enter a number: 2
2
Enter your choice: 3
Enter your choice: 4
Enter a number: 2
1
Enter your choice: 4
Enter a number: 6
2
Enter your choice: 6
Not empty
Enter your choice: 5
Enter your choice: 6
Empty
Enter your choice: 0
```