

## סדנא ב- C++ – 150018

### תרגיל בית מספר 1

#### תכנות מונחה עצמים

#### שים/י לב:

- הקפד/י על קריאות התוכנית ועל עימוד (Indentation).
- הקפד/י לבצע בדיוק את הנדרש בכל שאלה.
- בכל אחת מהשאלות יש להגדיר פונקציות במידת הצורך עבור קריאות התוכנית.
- יש להגיש את התרגיל על פי ההנחיות להגשת תרגילים (המופיע באתר הקורס) וביניהם:  
השתמש/י בשמות משמעותיים עבור המשתנים.  
יש לתעד את התוכנית גם עבור פונקציות אותם הנך מגדיר/ה וכן על תנאים ולולאות וקטעי קוד מורכבים ובנוסף, **דוגמת הרצה לכל תוכנית בסוף הקובץ!**  
הגשה 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:

- הגדיר/י מחלקה עבור המספרים הרציונליים.  
תזכורת: ערך המכנה לא יהיה 0 או שלילי)  
המחלקה תכלול את השדות הבאים:
- מונה (numerator)
  - מכנה (denominator)
- וכן את המתודות הבאות:
- עבור כל שדה:
    - מתודת הצבה (set). במכנה אין להציב 0, לכן במידה והפרמטר שמתקבל הוא 0 המתודה (setDenominator), תציב 1. שימו לב, כמו שמוגדר למעלה, אסור ערך שלילי במכנה יש לאתחל את השדות בהתאם.
    - מתודה המחזירה את ערכו (get).
  - מתודה להדפסת השבר הרציונלי. ההדפסה תהיה בפורמט: מונה/מכנה (לדוגמא  $1/2$ ,  $3/4$ ,  $54/56$  וכו') עפ"י הערכים המקוריים לא לאחר צמצום.
  - מתודה בוליאנית equal המשווה בין שני מספרים רציונליים ובודקת האם הם שווים. שימו לב: הכוונה של שווה היא שיש להם בדיוק אותו מונה ומכנה. (כלומר  $1/2$  ו-  $3/6$  הם לא שווים.)
  - מתודת עזר לצמצום השבר
    - חתימת המתודה: void reduction();
    - על המתודה לצמצם את העצם שקרא למתודה.
    - כלומר עבור
- ```
Rational rat(2,4);  
rat.reduction();
```
- עכשיו המונה של rat הוא 1 ומכנה של rat הוא 2.
- אחרי צמצום, אם המונה היה 0, אז הפונקציה יהפוך את המכנה להיות 1.
- מתודה add המחברת בין שני מספרים רציונליים ומחזירה את סכום שני המספרים (בתור אובייקט שלישי). על הפונקציה לצמצם את הסכום.
- כלומר עבור
- ```
Rational r1, r2;
```

```
r1.setNumerator(1);  
r1.setDenominator(6);  
r2.setNumerator(1);  
r1.setDenominator(3);
```

```
Rational r3;  
r3 = r1.add(r2);
```

r1 נשאר  $1/6$

r2 נשאר  $1/3$

r3 הוא  $1/2$

השתמש/י בתוכנית הראשית (main) הבאה כדי לבדוק את נכונות המחלקה שכתבת. התוכנית בודקת, את מתודות set, get, add ו-equals.

דוגמה להרצת התוכנית:

```
part 1. set/get  
enter a rational number:  
2/12  
numerator: 2  
denominator: 12  
enter a rational number:  
4/12  
4/12  
  
part 2. equals  
The two numbers are different  
  
part 3. add  
2/12 + 4/12 = 1/2
```

יש להשתמש ב-main המצורף עבור הרצת התוכנית:

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

int main()
{
    int num1, num2, num3;
    char junk;

    cout << "part 1. set/get" << endl;
    cout << "enter a rational number:" << endl;
    cin >> num1 >> junk >> num2;
    Rational rat1;
    rat1.setNumerator(num1);
    rat1.setDenominator(num2);
    cout << "numerator: " << rat1.getNumerator()<<endl;
    cout << "denominator: " << rat1.getDenominator()<<endl;

    cout << "enter a rational number:" << endl;
    cin >> num1 >> junk >> num2;
    Rational rat2;
    rat2.setNumerator(num1);
    rat2.setDenominator(num2);
    rat2.print();
    cout << endl << endl;

    cout << "part 2. equals" << endl;
    if (rat1.equal(rat2))
        cout << "The two numbers are equal" << endl;
    else
        cout << "The two numbers are different" << endl;
    cout << endl << endl;

    cout << "part 3. add" << endl;
    Rational ans = rat1.add(rat2);
    rat1.print();
    cout << "+ ";
    rat2.print();
    cout << "= ";
    ans.print();
    cout << endl;

    return 0;
}
```

**שאלה מס' 2:**

הגדר/י מחלקה אשר תייצג נתוני עובד בעמותת "לתת מהלב" לצורך חישוב משכורתו. המחלקה תכלול את השדות הבאים:

- מספר זהות (int).
- שם – עד 20 תווים (מערך סטטי); `char name[21];`
- שכר לשעה (float)
- מספר השעות שעבד (int).
- הסכום שהצליח להכניס לעמותה (float).

וכן את המתודות הבאות:

- עבור כל שדה:
  - מתודת הצבה (set).
  - מתודה המחזירה את ערכו (get).
- מתודה לחישוב המשכורת. חישוב המשכורת יבוצע בצורה הבאה:  
מספר שעות \* שכר לשעה + אחוזים מהסכום שהצליח להכניס לעמותה  
האחוזים יחושבו באופן הבא:

סכום ההכנסה :	אחוזים עבור המשכורת:
עד 1000 שקלים (כולל)	10%
מ-1000 שקלים עד 2000 שקלים (כולל)	15%
מ-2000 שקלים עד 4000 שקלים (כולל)	20%
מ-4000 שקלים עד 5000 שקלים (כולל)	30%
מ-5000 שקלים ומעלה	40%

לדוגמא: עובד שהכניס לעמותה 4500 שקלים, חלק האחוזים במשכורתו יחושב כך:  
 $1000 \cdot 0.1 + 1000 \cdot 0.15 + 2000 \cdot 0.2 + 500 \cdot 0.3 = 800$  כלומר, בנוסף למשכורתו הבסיסית של מספר השעות בתעריף לשעה ירוויח עובד זה עוד 800 שקלים

כתוב/י תוכנית ראשית אשר תקלוט פרטי עובדים עד להקשת ת.ז. 0. עבור כל עובד יקלטו הנתונים הבאים, לפי הסדר: מספר זהות, שם, שכר לשעה, שעות עבודה, הסכום שהצליח להכניס לעמותה. (ניתן להניח שקיים לפחות עובד אחד).

על התוכנית להדפיס:

- את מספר הזהות ושמו של העובד שהכניס הכי פחות כסף לעמותה. בנוסף יודפס סכום הכסף שהכניס לעמותה.
- את מספר הזהות ושמו של העובד שקיבל את המשכורת הגבוהה ביותר. בנוסף תודפס משכורתו של העובד.

שים/י לב:

- במקרה של קלט לא תקין, התוכנית תדפיס ERROR ותמשיך לקלוט את נתוני העובד הנוכחי

רמז: יש לכתוב הקוד כך שהתוכנית תקלוט תחילה את כל הנתונים של העובד ורק אחר כך תבדוק את תקינותם.

- להגבלות הרלוונטיות בכל שדה.
- בתוכנית הראשית אין להקצות מערך של עובדים.

דוגמאות להרצת התוכנית:

```
enter details, to end enter 0:  
123456789 moshe 50 40 2000  
135792468 rivka 120 55 3450  
97531246 sara 35 100 5632  
0  
minimum collected: 123456789 moshe 2000  
highest salary: 135792468 rivka 7140
```

```
enter details, to end enter 0:  
111111111 doron 35 120 6000  
222222222 tal 50 55 1400  
444444444 levi 45 -4 100  
ERROR  
333333333 naomi 30 120 800  
0  
minimum collected: 333333333 naomi 800  
highest salary: 111111111 doron 5550
```

### שאלה מס' 3:

הגדר/י מחלקה עבור ייצוג מעגל במישור (יש להגדיר מחלקת עזר Point עבור ייצוג נקודה).  
המחלקה (של המעגל) תכלול את השדות הבאים:

- נקודת המרכז
- הרדיוס.

וכן את המתודות הבאות:

- הצבה ואחזור (get/set) לכל שדה.
- חישוב שטח המעגל.
- היקף המעגל (חישוב היקף ושטח המעגל יהיה לפי  $PI=3.14$ ).
- מתודה המקבלת נקודה במישור ובודקת האם הנקודה נמצאת על המעגל (תחזיר 0), בתוך המעגל (תחזיר -1) או מחוץ למעגל (תחזיר 1).

כתוב/י תוכנית ראשית אשר תקלוט פרטים של שלושה מעגלים. התוכנית תדפיס עבור כל מעגל את ההיקף ואת השטח. (למעגל הראשון נקרא A, לשני B ולשלישי C).

לאחר מכן התוכנית תקלוט מספר לא ידוע מראש של נקודות במישור. סוף הקלט יסומן כראשית הצירים (0,0). עבור כל מעגל, התוכנית תחשב ותדפיס את מספר הנקודות הנמצאות על או בתוך המעגל (לא כולל ראשית הצירים).

דוגמא להרצת התוכנית:

```
enter the center point and radius of 3 circles:
(0,0) 3
(1,1) 2
(5,5) 2
perimeter:  A: 18.84    B: 12.56    C: 12.56
area:  A: 28.26    B: 12.56    C: 12.56
enter points until (0,0):
(0,1)
(1,0)
(0,4)
(0,0)
num of points in circle:  A:2    B:2    C:0
```

יש להשתמש בקבצי ה-header הבאים עבור הרצת התוכנית:  
קובץ Point.h:

```
#pragma once
class Point
{
private:
    int x;
    int y;
public:
    void setX(int myX);
    void setY(int myY);
    int getX();
    int getY();
};
```

קובץ Circle.h:

```
#pragma once
#include "Point.h"
class Circle
{
private:
    Point center;
    int radius;
public:
    void setCenter(int x, int y);
    void setRadius(int myRadius);
    Point getCenter();
    int getRadius();
    float area();
    float perimeter();
    int onInOut(Point p);
};
```

בהצלחה רבה!!