סדנא ב- C++ – 150018

תרגיל בית מספר 7

פולימורפיזם

<u>שים/י לב:</u>

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 בזוגות.

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

<u>שאלה מס' 1:</u>

בשאלה זו נגדיר מחלקה בסיסית אבסטרקטית Shape עבור ייצוג צורה מופשטת. בנוסף, נגדיר מחלקות יורשות עבור ייצוג מלבן, משולש ומעגל.

א. המחלקה תשתמש במערך של נקודות. לצורך כך נגדיר מחלקה עבור נקודה.
(אפשר להשתמש במחלקה של Point שהגדרתם בתרגיל בית 2 שאלה 1.)

- תזכורת במחלקה שהגדרתם בתרגיל בית 2 היו תכונות/הבנאים/מתודות הבאים:
  - x – מספר עשרוני (!) המייצג את המיקום על ציר x
  - y – מספר עשרוני (!) המייצג את המיקום על ציר y
  - **empty constructor** (בנאי ריק) - מאתחל את השדות x ו- y באפס (כך שנגדיר נקודה בראשית הצירים).
  - **constructor** – המקבל שני פרמטרים ומאתחל את שדות x ו-y לפי הפרמטרים שקבל.
  - **copy constructor** (בנאי העתקה)
  - הצבה ואחזור (get/set) לכל שדה.
  - מתודה המחשבת ומחזירה את המרחק בין הנקודה שקראה למתודה לבין הנקודה שהעבירו כפרמטר).
- יש להוסיף למחלקה את המתודה operator<< עבור קליטת נקודות
  (שים/י לב, הקלט יהיה מהצורה (x,y))

ב. הגדר/י מחלקה Shape עבור ייצוג צורה מופשטת.
המחלקה תכלול את התכונות הבאות:

- numOfPoints עבור מס' הנקודות(במישור) להגדרת הצורה (int)
- מערך נקודות Points עבור שמירת ערכי הנקודות המגדירות את הצורה במישור (Point*) – היעזר/י במחלקה Point המוגדרת בחלק א.

- שימו לב: במקרה של מעגל – numOfPoints יהיה 1 ומערך של נקודות יכיל מרכז המעגל (נקודה אחת בלבד)

בנוסף, המחלקה תכלול את הבנאים/מתודות הבאים:

- **empty constructor** (בנאי ריק) – המאתחל את גודל המערך ל-0 (מציבה 0 בתכונה numOfPoints ומציבה null במערך של נקודות).
- **constructor** – המקבל כפרמטר את מס' הנקודות במישור המגדירות את הצורה ומגדירה מערך בגודל מתאים. בנוסף, עליה לקלוט מהמשתמש את ערכי הנקודות בהתאמה ע"י הדפסת ההודעה: Enter values of # points: כאשר # מהווה את מספר הנקודות שעל המשתמש להכניס.
  (תזכורת: במחלקה Point כבר מוגדרת המתודה operator<< עבור קליטת נקודה בפורמט (x,y)
- **copy-constructor** – המבצע העתקה עמוקה
- **move-constructor**
- **virtual destructor**
- **אופרטור <<** - מתודה עבור פלט ערכי כל הנקודות מהצורה:
  points: (x1,y1) (x2, y2)... (xn,yn)
- מתודה const ()**area** לחישוב שטח הצורה. שטח צורה יהיה מספר עשרוני.
- מתודה בוליאנית const ()**isSpecial** הבודקת האם לצורה תכונה מיוחדת
- מתודה const ()**printSpecial** להדפסת התכונה המיוחדת

שים/י לב: ל-Shape אין תכונה מיוחדת וגם כן לא מייצגת צורה אמיתית ולכן אי אפשר לחשב לה את השטח – לכן חובה להצהיר על שלשת המתודות שלעיל להיות וירטואליות טהורות.

ג. הגדר/י מחלקה Circle עבור ייצוג מעגל (נקודת מרכז מעגל ורדיוס), היורשת ממחלקת Shape.

המחלקה תכלול בנוסף את התכונה הבאה:

- radius עבור רדיוס המעגל (float)
- (נקודת המרכז תישמר במערך של נקודות (Point))

בנוסף, המחלקה תכלול את הבנאים/מתודות הבאים:

- **constructor** – המקבל כקלט את רדיוס המעגל. בנוסף, בונה מערך עבור נקודת מרכז המעגל (ע"י זימון בנאי האב)
- **copy constructor – המעתיק את הרדיוס וקורא לבנאי העתקה עבור העתקה של שאר תכונות המחלקה**
- אין צורך באף בנאי אחר
- מתודה בוליאנית const ()isSpecial - הבודקת אם המעגל הוא מעגל קנוני.
  מעגל נחשב מעגל קנוני אם מרכז המעגל נמצא בראשית הצירים (0,0). על המתודה לבדוק האם המעגל הוא מעגל קנוני ולהחזיר ערך בוליאני מתאים
- מתודה const ()printSpecialלהדפסת התכונה המיוחדת של מעגל, כלומר - שהוא מעגל קנוני (במידה זה אכן כך) בפורמט הבא:
  A canonical circle with a radius #
  כאשר # מהווה את רדיוס המעגל.
  על התוכנית לבדוק האם קיימת התכונה המיוחדת (על ידי קריאה למתודה המתאימה) ובמידה שכן לקרוא למתודה של printSpecial להדפסת התכונה המיוחדת (שבמקרה שלנו היא שהמעגל הוא קנוני)
- מתודה ()area המחשבת את שטח המעגל.

בחישוב השטח – יש להשתמש **בקבוע** PI=3.14

ד. הגדר/י מחלקה Triangle עבור ייצוג משולש (שלוש נקודות המהוות את קודקודי המשולש), היורשת ממחלקת Shape.
המחלקה תכלול בנוסף את המתודות הבאות:

- empty-constructor – הבונה מערך עבור קודקודי המשולש (ע"י זימון בנאי האב)
- אין צורך באף בנאי אחר
- מתודה בוליאנית isSpecial() const - הבודקת אם המשולש הוא שווה צלעות.
  משולש נחשב למשולש שווה צלעות אם שלוש צלעות המשולש שוות. על המתודה לבדוק האם המשולש הוא משולש שלש צעלות ולהחזיר ערך בוליאני מתאים.
- מתודה printSpecial() const להדפסת תכונה מיוחדת זו (במידה שהיא קיימת) בפורמט הבא:

An equilateral triangle with a side length #

כאשר # מהווה את צלע המשולש.
על התוכנית לבדוק אם קיימת התכונה המיוחדת (על ידי קריאה למתודה המתאימה) ובמידה שכן לקרוא למתודה של printSpecial להדפסת התכונה המיוחדת (שבמקרה שלנו היא משולש שווה צלעות).

- מתודה area() המחשבת את שטח המשולש.

הערה: ניתן לחשב שטח משולש על ידי שימוש בנוסחה של הרון: :

$$s = \frac{a+b+c}{2}$$

$$area = \sqrt{s(s-a)(s-b)(s-c)}$$

ה. הגדר/י מחלקה Rectangle עבור ייצוג מלבן (ארבע נקודות המהוות את קודקודי המלבן), היורשת ממחלקת Shape.
המחלקה תכלול בנוסף את המתודות הבאות:

- empty-constructor – הבונה מערך עבור קודקודי המלבן (ע"י זימון בנאי האב)
- אין צורך באף בנאי אחר
- מתודה בוליאנית isSpecial() const - הבודקת אם המלבן הוא ריבוע.
  מלבן נחשב לריבוע אם כל צלעותיו שוות. על המתודה לבדוק האם המלבן הוא מלבן שווה צלעות ולהחזיר ערך בוליאני מתאים.
- מתודה printSpecial() const להדפסת תכונה מיוחדת זו (במידה שהיא קיימת) בפורמט הבא:

Square with side length #

כאשר # מהווה את צלע המלבן.
על התוכנית לוודא שהתכונה המיוחדת קיימת (על ידי קריאה למתודה המתאימה) ובמידה שכן לקרוא למתודה של printSpecial להדפסת התכונה המיוחדת (שבמקרה שלנו היא מלבן שהוא ריבוע).

- מתודה ()area המחשבת את שטח המלבן.

בכל אחת מן המחלקות שהגדרת יש לממש את כל המתודות הרלוונטיות.
שים/י לב במידה וישנן מתודות הנדרשות להיות מוגדרות כ- virtual או כ- virtual טהורות.
בנוסף, **יש להימנע משכפול קוד ואין לכתוב במחלקות היורשות קוד שאינו נצרך**. כלומר, במידה וקיימת מתודה או חלק מהקוד שבה אצל האב – אין לכתוב קטע קוד נוסף המבצע את אותה פעולה אצל הבן!!

הערה: ניתן להיעזר בספריית cmath עבור פונקציות מתמטיות

ו. לפנייך תוכנית ראשית חלקית המגדירה מערך המכיל נתונים של צורות.
שים/לב: כדי להגדיר מערך (המכיל גם אובייקטים שהם מופעים של Circle וגם של Triangle וגם של Rectangle היה צריך להגדיר מערך המכיל מצביע ל-Shape)
כלומר חובה להשתמש בהצהרה והקצאה הבאה:

```
Shape** shapes = new Shape * [numShapes];
```

שאלה למחשבה : מה עשינו (איך כתבנו את המחלקות) כך שהמתודות area, isSpecial ו- printSpecial יבצעו את הנדרש עבור כל סוגי האובייקטים?

התוכנית קולטת את מספר הצורות ואחר כך מאתחלת ומקצה מקום בזכרון עבור המערך הדינאמי.
לאחר מכן, התכנית קולטת נתונים עבור הצורות (מדפיס שאלה איזו צורה המשתמש רוצה ואחר כך קולט הנתונים המתאימים עבור הצורה) ושומר במקום המתאים במערך. במידה והתקבל קלט לא תקין עבור צורה יש להוציא הודעת שגיאה ואז לקלוט את כל הנתונים עבור הצורה מחדש.

לאחר מכן, יש לעבור על כל הצורות בלולאה אחת ולהדפיס את הנקודות המגדירות את הצורות השונות (ע"י אופרטור >>) ואת שטחי הצורות:

points: (x1,y1) (x2,y2)…(xn,yn) area is: #

לאחר מכן , במידה וישנה תכונה מיוחדת -מדפיס את תיאור התכונה כמו שהוגדר לעיל.

(הערה: תזכורת – הבדיקה לגבי התכונה נעשית ע"י מתודת ההדפסה printSpecial)

לפניך חלק מהתוכנית הראשית
חסרות הפקודות שיתבצעו

- בפקודת switch במידה שהמשתמש בחר ב3- (משולש)
- בפקודת switch במידה שהמשתמש בחר ב-4 (מלבן)
- בלולאה הסופית (שמדפיסה את כל הנקודות המגדירות את הצורות): הדפסת שטחי הצורות והדפסת תיאור תכונה עבור צורות בעלת תכונה מיוחדת

יש להשלים את תוכנית הראשית הבאה:

```
#include "Rectangle.h"
#include "Circle.h"
#include "Triangle.h"
enum SHAPES { CIRCLE = 1, TRIANGLE = 3, RECTANGLE };
int main() {


	int numShapes, choice;
	cout << "How many shapes you would like to define?\n";
	cin >> numShapes;
	Shape** shapes = new Shape * [numShapes];
	for (int i = 0; i < numShapes; i++) {
		cout << "Which shape will you choose? Circle - 1, Triangle - 3,
Rectangle - 4\n";
		cin >> choice;
		switch (choice) {
		case CIRCLE:
			float radius;
			cout << "Enter radius :\n";
			cin >> radius;
			shapes[i] = new Circle(radius);
			break;
		case TRIANGLE:

			break;
		case RECTANGLE:

			break;
		default:
			cout << "invalid input\n";
			i--;
		}

	}

	for (int i = 0; i < numShapes; i++) {



		}

	}


	return 0;
	}
```

דוגמא להרצת התכנית:

```
How many shapes you would like to define?
3
Which shape will you choose? Circle - 1, Triangle - 3, Rectangle – 4
3
Enter values of  3 points:
(0,0) (1,0) (1,1)
Which shape will you choose? Circle - 1, Triangle - 3, Rectangle – 4
1
Enter radius:
4
Enter values of  1 points:
(0,0)
Which shape will you choose? Circle - 1, Triangle - 3, Rectangle – 4
2
invalid input
Which shape will you choose? Circle - 1, Triangle - 3, Rectangle – 4
4
Enter values of  4 points:
(0,0) (1,0) (1,1) (0,1)

points: (0,0) (1,0) (1,1)  area is: 0.5

points: (0,0) area is: 50.24
A canonical circle with a radius 4


points: (0,0) (1,0) (1,1) (0,1)  area is: 1
Square with side length 1
```

בהצלחה רבה!!