

שפת C++ – תרגיל 1

היכרות עם השפה, classes, const, references, dynamic allocation, operators overloading

תאריך הגשה של התרגיל והבוחן: **יום ראשון 03.09.17 עד שעה 23:55**¹

1. הנחיות חשובות:

1. בכל התרגילים יש לעמוד בהנחיות הגשת התרגילים וסגנון כתיבת הקוד. שני המסמכים נמצאים באתר הקורס – הניקוד יכלול גם עמידה בדרישות אלו.
2. בכל התרגילים עליכם לכתוב קוד ברור. בכל מקרה בו הקוד שלכם אינו ברור מספיק עליכם להוסיף הערות הסבר בגוף הקוד. יש להקפיד על תיעוד (documentation) הקוד ובפרט תיעוד של כל פונקציה.
3. במידה ואתם משתמשים בעיצוב מיוחד או משהו לא שגרתי, עליכם להוסיף הערות בקוד המסבירות את העיצוב שלכם ומדוע בחרתם בו.
4. עבור כל פונקציה בה אתם משתמשים, עליכם לוודא שאתם מבינים היטב מה הפונקציה עושה גם במקרי קצה (התייחסו לכך בתיעוד). ובפרט עליכם לוודא שהפונקציה הצליחה.
5. בכל התרגילים במידה ויש לכם הארכה, או שאתם מגישים באיחור. **חל איסור להגיש קובץ כלשהוא בלינק הרגיל (גם אם לינק overdue טרם נפתח).** **מי שיגיש קבצים בשני הלינקים מסתכן בהורדת ציון משמעותית.**
6. אין להגיש קבצים נוספים על אלו שתדרשו. ובפרט אין להגיש קובץ README אלא אם צוין במפורש שיש צורך בכך (לדוגמא, בתרגיל זה אין צורך להגיש).
7. עליכם לקמפל עם הדגלים -g -pthread -std=c++17 -Wall -Wextra -Wvla. ולוודא שהתוכנית מתקמפלת ללא אזהרות, **תכנית שמתקמפלת עם אזהרות תגרור הורדה משמעותית בציון התרגיל.** למשל, בכדי ליצור תוכנית מקובץ מקור בשם ex1.cpp יש להריץ את הפקודה:
g++ -o ex1 ex1.cpp -g -pthread -std=c++17 -Wall -Wextra -Wvla
8. עליכם לוודא שהתרגילים שלכם תקינים ועומדים בכל דרישות הקימפול והריצה במחשבי בית הספר מבוססי מעבדי bit-64 (מחשבי האקווריום, לוי, השרת river). **חובה להריץ את התרגיל במחשבי בית הספר לפני ההגשה.** (ניתן לוודא שהמחשב עליו אתם עובדים הנו בתצורת bit-64 באמצעות הפקודה "uname -a" ווידוא כי הארכיטקטורה היא 64, למשל אם כתוב x86_64)
9. לאחר ההגשה, בדקו את הפלט המתקבל בקובץ ה-PDF שנוצר מהpresubmission script בזמן ההגשה. באם ישנן שגיאות, תקנו אותן על מנת שלא לאבד נקודות.
10. **שימו לב! תרגיל שלא יעבור את הpresubmission script ציונו ירד משמעותית** (הציון יתחיל מ-50, ויוכל לרדת) **ולא יהיה ניתן לערער על כך.**
11. בדיקת הקוד לפני ההגשה, גם על ידי קריאתו וגם על ידי כתיבת בדיקות אוטומטיות (tests) עבורו היא אחריותכם. בדקו מקרי קצה. במידה וסיפקנו לכם קבצי בדיקה לדוגמא, השימוש בהם יהיה על אחריותכם. **במהלך הבדיקה הקוד שלכם ייבדק מול קלטים נוספים לשם מתן הציון.**
12. **הגשה מתוקנת -** לאחר מועד הגשת התרגיל ירצו הבדיקות האוטומטיות ותקבלו פירוט על הטסטים בהם נפלתם. לשם שיפור הציון יהיה ניתן להגיש שוב את התרגיל לאחר תיקוני קוד ולקבל בחזרה חלק מהנקודות - **פרטים מלאים יפורסמו בפורום ואתר הקורס.**

¹ ניתן להגיש באיחור של עד 24 שעות עם קנס של 10 נקודות.

2. הנחיות חשובות לכלל התרגילים בקורס C++

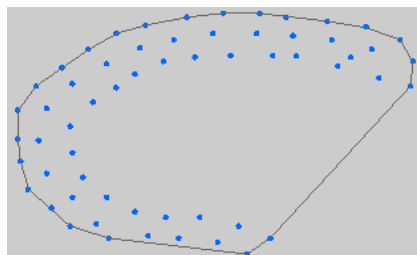
1. הקפידו להשתמש בפונקציות ואובייקטים של C++ (למשל new, delete, cout) על פני פונקציות של C (למשל malloc, free, printf).
2. בפרט השתמשו במחלקה string (ב-std::string) ולא במחרוזת של C (char *).
3. יש להשתמש בספריות סטנדרטיות של C++ ולא של C אלא אם כן הדבר הכרחי (וגם אז עליכם להוסיף הערה המסבירה את הסיבות לכך).
4. הקפידו על עקרונות Information Hiding – לדוגמא, הקפידו כי משתני המחלקות שלכם מוגדרים כמשתנים פרטיים (private).
5. הקפידו לא להעתיק by value משתנים כבדים, אלא להעבירם (היכן שניתן) by reference.
6. הקפידו מאוד על שימוש במילה השמורה const בהגדרות המתודות והפרמטרים שהן מקבלות: המתודות שאינן משנות פרמטר מסויים – הוסיפו const לפני הגדרת הפרמטר.
7. מתודות של מחלקה שאינן משנות את משתני המחלקה – הוסיפו const להגדרת המתודה.
8. שימו לב: הגדרת משתנים / מחלקות ב- C++ קבועים הוא אחד העקרונות החשובים בשפה.
9. הקפידו על השימוש ב-static, במקומות המתאימים (הן במשתנים והן במתודות).
10. הקפידו לשחרר את כל הזיכרון שאתם מקצים (השתמשו ב-valgrind כדי לבדוק שאין לכם דליפות זיכרון).
11. שימו לב שהאלגוריתמים שלכם צריכים להיות יעילים.
12. אתם רשאים (ולעיתים אף נדרשים) להגדיר פונקציות נוספות לשימושכם הפנימי.

3. הנחיות ספציפיות לתרגיל זה:

1. שימו לב שהאלגוריתמים שלכם צריכים להיות יעילים.
2. בתרגיל זה אין להגיש קבצים נוספים.
3. חל איסור להשתמש במבני נתונים מוכנים בתרגיל (כדוגמת STL) שימוש כזה יוביל לפסילת הסעיף הרלוונטי.
4. טיפול בשגיאות ב-C++ מבוצע ע"י מנגנון חריגות (exceptions). אתם תלמדו על הנושא בהמשך הקורס.
5. בתרגיל זה אנו מניחים שהקלט תקין וכי הקצאות זיכרון מצליחות תמיד.
6. מומלץ מאוד לקרוא ולהבין היטב את סעיפים 4-5 לפני שאתם מתחילים לממש את התרגיל. תכנון נכון עשוי לחסוך לכם זמן יקר.

4. קמור - ConvexHull:

1. הקמור של גוף (או אוסף של גופים) הוא הגוף הקמור המינימלי המכיל אותם.
2. קמור של קבוצת נקודות הוא הצורה הקמורה הקטנה ביותר שמכילה את הנקודות.
3. במקרה הפרטי שבו אוסף הגופים הוא אוסף של נקודות, קטעים ומצולעים במישור הדו-ממדי, הקמור הוא החיתוך של כל הקבוצות הקמורות שמכילות את קבוצת הנקודות.
4. לקמירות שימושים רבים מתחומי המתמטיקה והיא עומדת בבסיסם של אלגוריתמים רבים בגיאומטריה חישובית, זיהוי כתב ותבניות, עיבוד תמונה, תכנון לינארי ועוד.
5. מבחינה אינטואיטיבית ניתן להגדיר את הקמור באופן הבא:
 - a. תהי X קבוצת נקודות במרחב הדו מימדי.
 - b. נמתח גומייה דמיונית סביב קבוצת הנקודות, הגומיה "תיתפס" במספר נקודות בקצוות, נסמנם ב- $H(X)$.
 - c. אוסף נקודות זה מהווה פוליגון קמור המקיף את קבוצת הנקודות X.



d. שימו לב ש- $H(X)$ הינה תת קבוצה של קבוצת הנקודות X .

מטרתנו בתרגיל הינה למצוא את $H(X)$.

5. למציאת הקמור קיימים אלגוריתמים רבים ומגוונים בעלי מאפיינים שונים ומעניינים. בתרגיל עליכם לממש את האלגוריתם $Graham's scan$.

6. עיינו בקישורים הבאים לצורך הבנת האלגוריתם ואופן פעולתו:

(גם אם קיימים הבדלים מינוריים בין הגרסאות השונות של אלגוריתם הינכם רשאים לממש את הגרסה הנוחה לכם).

http://www.gadial.net/2011/10/26/graham_scan/

https://en.wikipedia.org/wiki/Graham_scan

<http://www.cs.princeton.edu/courses/archive/fall08/cos226/demo/ah/GrahamScan.html>

<http://www.personal.kent.edu/~rmuhamma/Compgeomtry/MyCG/ConvexHull/GrahamScan/grahamScan.htm>

<http://geomalgorithms.com/a10-hull-1.html>

7. שימו לב כי האלגוריתם כולל שני שלבים עיקריים:

a. מיון הנקודות לפי זווית.

b. איטרציה על הנקודות לשם בחירת נקודות הקמור מתוכן.

בקישור הבא קיימת הדגמת מציאת הקמור של קבוצת נקודות בצורה וויזואלית:

<https://www.youtube.com/watch?v=dw120Yclav8>

1. תכניתכם (שתמומש בקובץ `ConvexHull.cpp`) תקבל כקלט מהמשתמש רצף של קורדינטות (נקודות במרחב הדו-ממדי), ותחשב את הקמור של נקודות אלו.

2. פורמט הקלט הוא:

<x num>,<y num>

<x num>,<y num>

...

3. פלט התכנית יהיה: המחרוזת "result\n" ורשימת נקודות המרכיבות את הקמור. כאשר:

3.1. כל נקודה בשורה חדשה.

3.2. הרשימה ממויינת לפי ציר ה-x (מיון ראשי), וציר ה-y (מיון משני).

4. לדוגמא, עבור הקלט הבא:

>ConvexHull

1,100

12,50

1,200

14,30

-12,-45

...

פלט התוכנית יהיה:

result\n

(-12,-45)\n

(1,100)\n

(1,200)\n

...

5. תכנון ומבנה הנתונים:

5.1. ככלל, תכנון התכנית מוטל עליכם מלבד המגבלות הבאות:

5.1.1. על תכניתכם לכלול את המחלקות הבאות:

(i) Point - מייצגת נקודה במרחב.

(ii) PointSet - מייצגת קבוצת נקודות (ללא כפילויות).

(שימו לב! מחלקה זו מכילה גם את הפתרון לסעיף הבא בתרגיל)

5.1.2. המחלקה Point תמומש בקבצים Point.h ו-Point.cpp, ותכלול (לפחות) את השיטות הבאות:

(i) בנאי והורס.

(ii) פונקציית toString שמחזירה string בפורמט הבא:

"(<x num>,<y num>)"

(iii) פונקציית set שמקבלת שתי מספרים מטיפוס long int:

set(<x num>,<y num>)

(iv) בנאי שמקבל שני מספרים מטיפוס long int:

Point(<x num>,<y num>)

5.1.3. המחלקה PointSet, תתנהג כקבוצה מתמטית (ללא אברים כפולים), ותמומש בקבצים

PointSet.h ו-PointSet.cpp, ותכלול (לפחות) את השיטות הבאות:

(v) בנאי והורס.

(vi) פונקציית toString שמחזירה string בפורמט הבא:

"(<1st x num>,<1st y num>)\n...<((size)st x num>,<(size)st y num>)\n"

(קבוצה ריקה מחזירה "")

(vii) המתודה add המוסיפה נקודה לסוף הקבוצה. (מחזירה true אם האיבר

אכן נוסף לקבוצה)

(viii) המתודה remove המוציאה נקודה מהקבוצה (מחזירה true אם האיבר

היה קיים בקבוצה). המתודה הזו עשויה להשפיע על סדר האיברים בקבוצה.

(ix) המתודה size המחזירה את גודל הקבוצה.

5.2. מספר הנקודות איננו מוגבל, באפשרותכם לממש את קבוצת הנקודות באמצעות מערך (שיוגדל

במידת הצורך), רשימה מקושרת, או כל מבנה נתונים אחר.

5.3. הנכם רשאים להוסיף מחלקות נוספות לתכנית.

5.4. בשאלה זו אין להשתמש במבני נתונים מוכנים בתרגיל (כדוגמת STL). אם אתם זקוקים לאחד,

ממשו בעצמכם.

5.5. הסבירו בקובץ ה- README (באנגלית בלבד) במפורט את מבנה תכניתכם והמחלקות בהן אתם

משתמשים.

6. הדרכה והנחיות כלליות:

6.1. לחלק זה של התרגיל פורסם פתרון ב"ס. בכל מקרה של שאלה בדקו כיצד פתרון בית הספר

מתמודד איתה. על הפתרון שלכם לתאום את ההתנהגות של פתרון בית הספר.

6.2. אתם רשאים להניח כי כל המספרים הינם מספרים שלמים מסוג `long int`, וכי הזנת הקורדינטות תסתיים ב-EOF.

6.3. הקלט אינו ממויין, והוא עשוי להכיל כפילויות.

6.4. אתם רשאים להניח כי הקלט תקין.

6.5. עליכם להתמודד גם עם רשימת נקודות בגודל ≥ 2 (חשבו כיצד).

6.6. במקרה של נקודות הנמצאות על קו ישר ומשתייכות לקמור, עליכם להדפיס רק את הקיצוניות.

6.7. 0 נקודות זה קלט תקין.

6.8. עליכם לפרט בהערה לפני מימוש האלגוריתם:

6.8.1. כיצד פועל האלגוריתם שלכם.

6.8.2. הסבר על מידת הסיבוכיות שלו.

6.8.3. הפנייה למקור האלגוריתם (במידה ולקחתם אותו ממקום כלשהו).

5. מימוש אופרטורים עבור `PointSet`:

1. עליכם לממש את האופרטורים הבאים עבור המחלקה `PointSet`:

6.9. אופרטור השוואה:

6.9.1. `!=operator`

6.9.2. `==operator`

2 `PointSet` זהות אם הן מכילות את אותן נקודות (אין משמעות לסדר).

למשל:

$A = \{(1,2);(2,6);(9,-1)\}$

$B = \{(2,6);(9,-1);(1,2)\}$

$C = \{(1,2);(2,4);(9,-1)\}$

$A == B$

$A != C$

ערך ההחזרה של האופרטורים הוא `bool`.

6.10. `-operator`

הפונקציה תחזיר קבוצה חדשה המכילה את הנקודות מה-קבוצה השמאלית אשר אינם מופיעות

ב-קבוצה הימנית².

6.11. `& operator`

זהו אופרטור המבצע את פעולת ה-Intersection על `PointSets`. הפונקציה תחזיר קבוצה חדשה

המכילה את כל הנקודות שמופיעות גם בקבוצה השמאלית וגם בימנית³.

6.12. אופטורי השמה:

עליכם לממש פונקציות (בנאי העתקה) ואופרטורים בכדי לתמוך בפעולות השמה.

למשל - `"PointSet2=PointSet1"`.

עליכם לכתוב גם קובץ בשם `PointSetBinaryOperations.cpp` המכיל פונקציות `main` ויוצר קובץ

הרצה המדגים את השימוש בכל האופרטורים שהוגדרו כאן.

² מסומן בדרך כלל `"left\right"`

³ מסומן בדרך כלל `"left\right"`

2. הדרכה והנחיות כלליות:

- 6.13 יש לממש את האופרטורים במחלקה עצמה בקבצים PointSet.h ו-PointSet.cpp כפונקציות מחלקה. הבנת החתימות הנדרשות של פונקציות האופרטורים היא חלק מהדרישות.
- 6.14 אתם רשאים לשנות את PointSet.h.
- 6.15 הקפידו שהפונקציות תעבודנה ללא דליפות זיכרון.
- 6.16 הקפידו שלא לשכפל קוד בין האופרטורים השונים.
- 6.17 מסופק לכם קובץ בדיקה בסיסי TestPointSetBinFuncs.cpp. אתם רשאים לשנות אותו כרצונכם ואין להגיש אותו.

6. בונוס (3 נקודות):

1. כל סטודנט שימצא שגיאה חדשה בפתרון בית הספר יקבל בונוס של 3 נקודות לציון התרגיל.
2. בכדי לקבל את הבונוס עליכם לפתוח דיון בפורום התרגיל שכלול את:
 - a. קובץ קלט שגורם לשגיאה.
 - b. קובץ פלט שמכיל את פלט פתרון בית הספר שנוצר מריצתו עם קובץ הקלט.
 - c. הסבר מפורט מהי השגיאה.

7. חומר עזר:

1. את פתרון הבית ספר ניתן למצוא ב:

~plabcpp/www/ex1/schoolSol.tar

2. את הקבצים המסופקים לצורך התרגיל ניתן למצוא ב:

~plabcpp/www/ex1/ex1_files.tar

3. קבצי בדיקה לדוגמא ניתן למצוא ב:

~plabcpp/www/ex1/tests_examples.tar

4. ++C-include <iostream>overloading# :

http://www.cprogramming.com/tutorial/operator_overloading.html

http://en.wikipedia.org/wiki/Operators_in_C_and_C%2B%2B

<http://www.cplusplus.com/reference/set/set/operators/>

8. הגשה:

1. עליכם להגיש קובץ tar בשם ex1.tar המכיל רק את הקבצים הבאים:

- README
- Point.cpp ו-Point.h
- PointSet.cpp ו-PointSet.h
- ConvexHull.cpp
- PointSetBinaryOperations.cpp
- קובץ Makefile התומך (לפחות) בפקודות הבאות:

- o make tar - יצירת קובץ tar בשם ex1.tar, המכיל רק את הקבצים שצריך להגיש בתרגיל.
- o make clean - ניקוי כל הקבצים שנוצרו באמצעות פקודות ה-makefile.

- o make PointSet.o - קימפול, ויצירת PointSet.o.
- o make Point.o - קימפול, ויצירת Point.o.
- o make PointSetBinaryOperations - יצירת קובץ ההרצה PointSetBinaryOperation.⁴
- o make ConvexHull - יצירת קובץ ההרצה ConvexHull⁵
- o הרצת make ללא פרמטרים תבנה את קבצי ההרצה ConvexHull
- ו-PointSetBinaryOperations, ותריץ את PointSetBinaryOperations.⁶
- extension.pdf - בק במקרה שההגשה היא הגשה מאושרת באיחור (מכיל את האישורים הרלוונטים להארכה).
- 2. לפני ההגשה, פתחו את הקובץ ex1.tar בתיקה נפרדת וודאו שהקבצים מתקמפלים ללא שגיאות וללא אזהרות.
- 3. מומלץ מאוד גם להריץ בדיקות אוטומטיות וטסטרים שכתבתם על הקוד אותו אתם עומדים להגיש. בנוסף, אתם יכולים להריץ בעצמכם בדיקה אוטומטית עבור סגנון קידוד בעזרת הפקודה:
- ~plabcpp/www/codingStyleCheck <file or directory>
- כאשר <directory or file> מוחלף בשם הקובץ אותו אתם רוצים לבדוק או תיקייה שיבדקו כל הקבצים הנמצאים בה (שימו לב שבדיקה אוטומטית זו הינה רק חלק מבדיקות ה codingStyle)
- 4. דאגו לבדוק לאחר ההגשה את קובץ הפלט (submission.pdf) וודאו שההגשה שלכם עוברת את ה- presubmission script ללא שגיאות או אזהרות.
- ~plabcpp/www/ex1/presubmit_ex1

בהצלחה!

⁴ ללא בדיקות debug
⁵ ללא בדיקות debug
⁶ ללא בדיקות debug