

תרגיל 5 – תבניות, קבצים וטיפול בחריגות להגשה עד 6.06.15

הוראות ההגשה ודגשים מיוחדים מופיעים בסוף התרגיל! חובה לקרוא ולפעול לפיהן.

יש לקרוא את כל העבודה לפני שמתחילים!

שאלות יש להפנות אך ורק למרצה האחראית סבטלנה רוסין למייל assignc2014@gmail.com

בשליחת מיילים בנוגע לעבודה חובה לפרט בכותרת ההודעה על איזה קורס מדובר – "מבוא למדעי

מחשב מתקדם".

חלק א' – תאורטי (מענה בקובץ טקסט – וורד): משימה 1:

- (1) מהו התהליך המתבצע בהפעלת פונקציית תבנית, יש להסביר בקצרה על הפעולות של המהדר במקרה זה.
- (2) מימושן של כלל פונקציות מחלקת ה-template class מבוצע כולו בקובץ ההכרזה, מדוע? יש לתת הסבר קצר.
- (3) מהו עידון (Specialization)? יש לתת הסבר קצר.
- (4) מה הן התופעות הנובעות משימוש ב- templates?
- (5) מה מתרחש במקרה בו פונקציה מסוימת (function1) אינה מטפלת בחריגה הנזרקת מפונקציה שנקראה מתוכה (function2), כלומר

```
#include<iostream>
using namespace std;
void function2(){
    throw string("some exception");
}
void function1(){
    function2();
}
int main(){
    try{
        function1();
    }
    catch (string e){
        .....
    }
}
```

- (6) יש לתת 2 דוגמאות למחלקות חריגה תקניות (סטנדרטיות).
- (7) מה מונע תהליך ה- stack unwinding (שחרור מחסנית) ? - יש להסביר בקצרה.
- (8) בהינתן מחלקה X המכריזה על זריקת חריגה

```
class X{
    //...
public:
    X() throw (runtime_error){}
};
```

וקוד המשתמש הבא:

```
X* ptr=new X[1000000];
```

תיתכן זריקת חריגה בכל אחד מהשלבים. מה יהיה אופן הטיפול הראוי בחריגות (יש לכתוב את חלק הקוד הרלוונטי).

משימה 2:

(1) בדוגמה הבאה קיימת שגיאה.

מאיזה סוג היא .

וכיצד יש לתקן אותה.

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

class Matherr {};
class OverflowException : public Matherr {};
class UnderflowException : public Matherr {};
class ZeroDivideException : public Matherr {};

double divide(int numerator, int denominator) {
    if (denominator == 0)
        throw ZeroDivideException();
    double d = (double)numerator / denominator;
    return d;
}

double calculate(int number){
    if (number < 0)
        throw Matherr();
    return sqrt(number);
}

int main() {
    try {
        cout << calculate(4) << endl;
        cout << divide(6, 0) << endl;
        cout << calculate(-4) << endl;
    }
    catch (Matherr) { cerr << "Math Error" << endl; }
    catch (ZeroDivideException) { cerr << "Zero Divide Error" << endl; }
    catch (...) { cerr << "Unknown Error" << endl; }
}
```

(2) בדוגמה הבאה קיימת שגיאה.

מאיזה סוג היא .

וכיצד יש לתקן אותה.

```
class A{
public:
    ~A(){ throw exception(); }
};

void Function(){
    A obj;
    throw string("Some exception");
}

void AnotherFunction(){
    try{
        Function();
    }
    catch (string& e){
        cerr << "Exception:" << e << endl;;
    }
}
```

```

    }
    catch (exception & e){
        cerr << "Exception:" << e.what() << endl;
    }
}
int main(){
    AnotherFunction();
}

```

חלק ב' – מעשי (ההגשה היא של קבצי ה-CPP ו-H בלבד):

חלק זה מורכב משתי משימות נפרדות.

משימה 1 Template functions :

במשימה זו עליכם לבנות תבניות לפונקציות הבאות:

1. המקבלת מערך מסוג כללי וממיינת אותו בסדר עולה (מקטן לגדול) לפי מיון המהיר (quick sort)
http://he.wikipedia.org/wiki/%D7%9E%D7%99%D7%95%D7%9F_%D7%9E%D7%94%D7%99%D7%A8

2. המקבלת מערך מסוג כללי וערך נוסף מסוג כללי (סוג כללי – אחיד). הפונקציה תבדוק האם הערך הנוסף מופיע במערך הנתון או לא ותחזיר את הערך הבוליאני.

לבסוף, כתבו תכנית ראשית אשר תשתמש בפונקציות הנ"ל עבור הסוגים הבאים:
 long, char, class Date (user – defined class ,
 (ניתן לקחת מפתרון של המעבדה ,
 אפשר להניח שהחיפוש יתבצע תמיד אחרי המיון ולכן יש לחשוב על תהליך החיפוש היעיל ביותר מבחינת זמן ריצה.

משימה 2 Class Template :

הגדרה: קבוצה הינה אוסף מוגדר היטב של איברים "יחודיים" (ללא חשיבות לסדר כלשהו).

אנו נייצג קבוצה על ידי שימוש במבנה נתונים מסוג מערך דינאמי .
 בתרגיל זה עליכם לבנות את המחלקות הבאות:

א. מחלקה של קבוצת איברים מסוג כללי

יש ליצור מחלקה בשם CSet, המייצגת קבוצת ערכים מסוג כללי תוך שימוש בשדה שהוא מערך דינאמי.

יש ליצור עבור המחלקה:

- בנאי ברירת מחדל (היוצר קבוצה ריקה).
- בנאי מעתיק (יוצר קבוצה חדשה כעותק של קבוצת-פרמטר).
- הורס שמשחרר את הקבוצה.
- אופרטור השמה = המעדכן את האובייקט מסוג הנ"ל הנוכחי (LValue) ע"י אובייקט הפרמטר (RValue).

יש ליצור את הפונקציות והאופרטורים הבאים עבור המחלקה לטיפול בקבוצות:

1. אופרטור $+$, אשר מקבל ערך חדש ומוסיף אותו לקבוצה (**דגש** – אין חשיבות לסדר) רק במידה והערך אינו קיים בקבוצה, אחרת לא תתבצע אף פעולה נוספת. לדוג' נניח כי ישנה קבוצת מספרים שלמים מסוימת $A=(12,14,16,17)$ (ללא חשיבות לסדר כלשהו). בעת הפעלת אופרטור זה על קבוצה A עם המספר 8, איברי הקבוצה יעודכנו ל-: $(12,14,16,17,8)$.

2. אופרטור $-$, אשר מקבל ערך, במידה וערך זה מופיע בקבוצה, אזי הוא יוציא אותו מקבוצה זו, אחרת לא תתבצע אף פעולה נוספת. בהינתן הפעלת אופרטור זה על קבוצה A $(12,14,16,17,8)$ וערך נוסף 16, איברי הקבוצה יעודכנו ל-: $(12,14, 17,8)$.

3. אופרטור $|$ (or) הפועל על קבוצה ומקבל כפרמטר קבוצה נוספת, תוצאת האופרטור הינה קבוצה חדשה המתקבלת כאיחוד שתי הקבוצות (כלל האיברים הייחודיים משניהן). לדוג', איחוד של $(18,3)$ ו- $(3,1)$ היא הקבוצה $(3,1,18)$.

4. אופרטור $\&$ (and) הפועל על קבוצה ומקבל כפרמטר קבוצה נוספת, תוצאת האופרטור הינה קבוצה חדשה המתקבלת כחיתוך שתי הקבוצות (כלל האיברים המשותפים משניהן). לדוג', חיתוך של $(18,3)$ ו- $(3,1)$ היא הקבוצה (3) (כמובן שתוצאת החיתוך יכולה להיות קבוצה ריקה).

5. אופרטור $\>$, אופרטור הפועל על קבוצה ומקבל כפרמטר קבוצה נוספת, תוצאת האופרטור הינה קבוצה חדשה המורכבת מכלל האיברים שנמצאים בקבוצת הבסיס ואינם כלולים בקבוצת הפרמטר. לדוג', קבוצת הבסיס הינה $(10,7,8,1)$, וקבוצת הפרמטר $(8,7,3)$ כך שהאופרטור יחזיר את הקבוצה $(10,1)$.

6. אופרטור $=$, המשווה בין 2 קבוצות, ומחזיר אמת במידה והקבוצות מכילות את אותם האיברים ללא חשיבות לסדר שלהם בתוך הקבוצות. למשל, הקבוצות הבאות הינן זהות: $(18,3,7)$ ו- $(3,18,7)$

7. אופרטור $\>$, המהווה את הפעולה "A מכיל B", כלומר קבוצה A מכילה לפחות את כל איברי B, למשל:

$A: (2,1,4,3,6,5,8,9,7)$
 $B: (3,1,9,5,7)$ מחזיר אמת.

8. אופרטור \wedge - , המהווה את פעולת XOR ("אֶקְסוֹר"), $A|B-A\&B$, למשל $(2,1,5,4,8,7) \wedge (3,1,6,5,9,8) \Rightarrow (2,4,7,3,6,9)$

9. פונקציה **DifWrite** המקבלת אובייקט-פרמטר שבו נמצאת קבוצה כלשהיא (קבוצה הראשונה שמורה בתוך האובייקט הנוכחי), ומפעילה את האופרטור $-$ ביחס לאובייקט הנוכחי (האובייקט הנוכחי "מינוס" אובייקט-פרמטר). הקבוצה החדשה שמתקבלת, תיכתב לקובץ (טקסט או בינארי לפי בחירתכם).

10. אופרטור פלט \ll למסך אשר מדפיס את הקבוצה באופן מסודר, בצורה הבא:

$(a_1,a_2,...,a_n)$

ניתן להוסיף מתודות עזר כרצונכם, אך אסור להוסיף שדות.

ב. מחלקת תפריט

מחלקת תפריט תנהל עבורנו את כלל הקבוצות. בהתחלה המחלקה תבנה 5 אובייקטים מהתבנית הנ"ל:
2 קבוצות של מספרים שלמים (int), קבוצה אחת של תווים (char) ו-2 קבוצות של תאריכים (Date).
לאחר-מכן, יופעל עבור המשתמש התפריט הבא (בלולאה עד לבחירת יציאה):

1. הוספת איבר חדש לקבוצה:

לאחר בחירה באפשרות זו, המשתמש יידרש לבחור לאיזו קבוצה הוא רוצה להוסיף איבר וכמו-כן יידרש להזין את האיבר החדש.

2. מחיקת איבר מהקבוצה הקיימת:

לאחר בחירה באפשרות זו, על המשתמש יהיה לבחור מאיזו קבוצה הוא מבקש להוציא את האיבר וכמו-כן עליו יהיה להזין את האיבר שברצונו להוציא.

3. איחוד 2 הקבוצות המכילות קבוצות המספרים השלמים:

לאחר בחירה באפשרות זו, המערכת תדפיס על-גבי המסך את תוצאת פעולת האיחוד בין קבוצות המספרים השלמים.

4. חיתוך 2 הקבוצות המכילות תאריכים:

לאחר בחירה באפשרות זו, המערכת תדפיס על-גבי המסך את התוצאה של פעולת החיתוך בין קבוצות התאריכים.

5. הפרש 2 הקבוצות המכילות קבוצות המספרים השלמים:

לאחר בחירה באפשרות זו, על המשתמש יהיה לבחור את 2 הקבוצות לפי הסדר הרצוי, ראשית עליו לבחור את קבוצת הבסיס ולאחר-מכן את קבוצת הפרמטר, לבסוף המערכת תדפיס על-גבי המסך את תוצאת פעולת ההפרש בין קבוצות המספרים השלמים, כלומר את המספרים השלמים שמופיעים בקבוצת הבסיס אך לא בקבוצת הפרמטר.

6. הפעולה "מכיל" עבור 2 קבוצות תאריכים:

לאחר בחירה באפשרות זו, על המשתמש יהיה לבחור את 2 הקבוצות לפי הסדר הרצוי, ראשית עליו לבחור את קבוצת הבסיס ולאחר-מכן את קבוצת הפרמטר, לבסוף המערכת תדפיס על-גבי המסך את התוצאה של פעולת "מכיל" עבור 2 קבוצות תאריכים.
לדוגמא, במידה ו-A מכיל את B, תוצג ההודעה הבאה:

The base Group contains the parameter Group

לעומת זאת, במידה ו-A אינו מכיל את B, תוצג ההודעה:

The parameter Group is not contained in the base Group

7. הפעולה "אקסור" עבור 2 קבוצות המספרים השלמים:

לאחר בחירה באפשרות זו, על המשתמש לבחור את 2 הקבוצות, המערכת תדפיס על-גבי המסך את התוצאה של פעולת "אקסור" עבור 2 קבוצות המספרים השלמים.

8. הפעולה "שוויון" עבור 2 קבוצות תאריכים:

לאחר בחירה באפשרות זו, המערכת תדפיס על-גבי המסך את תוצאת פעולת השוויון בין קבוצות התאריכים.

לדוגמא, במידה ו-A שווה ל-B, תוצג ההודעה הבאה:

The two Groups are equal

לעומת זאת, במידה ו-A אינו שווה ל-B, תוצג ההודעה:

The two Groups are not equal

9. פעולת הפרש בין 2 קבוצות תאריכים וכתיבתו לקובץ:

לאחר בחירה באפשרות זו, על המשתמש יהיה לבחור את 2 הקבוצות לפי הסדר הרצוי, ראשית עליו לבחור את קבוצת הבסיס ולאחר-מכן את קבוצת הפרמטר, לבסוף המערכת תיצור אובייקט

חדש ותרשום בקובץ (טקסט או בינארי לפי בחירתכם) את הקבוצה השמורה בתוך האובייקט החדש (תוצאת פעולת הפרש בין קבוצות התאריכים, כפי שהוסבר בפירוט מקודם).

10. הדפסת קבוצה על-גבי המסך :

לאחר בחירה באפשרות זו, על המשתמש יהיה לבחור איזו קבוצה ברצונו להדפיס ואז המערכת תדפיס אותה.

11. יציאה מסודרת :

לאחר בחירה באפשרות זו, המערכת תצא בצורה מסודרת תוך שחרור כל המשאבים.

הפונקציה הראשית (main):

תיצור אובייקט מסוג תפריט ותפעיל אותו כך שיפעיל את המערכת ויריץ את התפריט.

הערה חשובה: שימוש במחלקה בנויה או ב-STL יוביל לאיפוס ציון התרגיל כולו! ראו הוזהרתם!

בנוסף יש לטפל במקומות הרלוונטיים בחריגות האפשריות, למשל מספר דוגמאות

למצבים "בעייתיים":

- שגיאה בהקצאת זיכרון דינאמי
- בעיית פתיחת קובץ לכתיבה
- גלישה מגבולות המערך
- הכנסת ערכים לא תקינים (למשל עבור תאריך הערכים התקינים עבור ימים הם 1-28/29,30,31, בהתאם ללוח השנה; עבור חודשים 1-12; עבור שנים – לפחות 2000).

סעיף סייבר : יש לספק פתרון נוסף לתרגיל בעזרת רשימה מקושרת.

הערות כלליות (לכל חלקי התרגיל):

1. הקלטים יהיו מהטיפוסים החוקיים.
 2. אחרי כל הדפסה יש לבצע ירידת שורה.
 3. בתרגיל מותר להשתמש בספריות `fstream`, `iostream` וכל ספריה הקשורה לטיפול בחריגות
- בלבד!**
4. יש להקפיד על תכנות נכון:

- a. כל הערכים שהם קבועים, (מבחינה לוגית הם לא אמורים להשתנות), **חייבים** להיות מוגדרים כ: `enum` או `const`, `define`, בהתאם לצורך.
- b. יש לרשום הערות בשפע! **ובאנגלית בלבד** (לכל מחלקה ופונקציה מה התפקיד שלה, כל תוכנית מה היא מבצעת, כל פעולה לא טריוויאלית להסביר, וכל 2~3 שורות קוד – הערה, כל מתודה מה היא עושה).

c. יש להקפיד על **כימוס נכון** – כל השדות ומתודות השירות ב-private, והממשק ב-

public! כמו כן: חלוקה לקבצים! **כל מחלקה בקבצים נפרדים!**

d. יש להקפיד על הזחות!!! כיתוב נכון וקריא! ושמות משמעותיים!

e. יש לנסות ולייעל את הקוד והתוכנית ככל שניתן. הקפידו על **reuse** בקוד.

f. לפני בקשת קלט (cin) יש להדפיס למשתמש הוראה (cout) איזה קלט מבוקש.

g. יש להקפיד על מוסכמות התכנות הנכון (שמות כמו שצריך וכו') וכללי התכנות שנלמדו.

h. להזכירכם: הנכם נדרשים לתכנת בשפת C++. אי-לכך, כל שימוש בפונקציות וספריות

של שפת C **אסורה**.

5. בהצלחה ☺

הנחיות הגשה

- תרגילים הם ביחידים! **כל עבודה משותפת היא אסורה ותיענש בחומרה!**
- הקובץ של החלק הראשון יש להציב בתיקיה ששמה part1 את הקבצים של החלק השני יש לאגד בתיקיה אחת ששמה part2 ואת שתי התיקיות לאגד תחת קובץ zip ששמו הוא לפי ת"ז למשל 2345678.zip.

- **ההגשה היא של קובצי הקוד (קבצי ה- CPP ו-H) וקובץ וורד (אין להגיש סריקה של כתב**

יד!!) בלבד, שישלח אך ורק דרך המערכת! (לכל מחלקה קובץ cpp ו-h משלה!).

- **כל שאלה ופניה בנוגע לתרגיל יש להפנות אך ורק למרצה האחראית סבטלנה רוסיין**

assignc2014@gmail.com

- בתחילת הקובץ יש להוסיף את התיעוד הבא:

/* Assignment: 3

Campus: Ashdod / Beer Sheva (תבחרו את המתאים)

Author: Israel Israeli, ID: 01234567 */

כמובן שיש לעדכן את השמות ומספרי תעודות הזהות שלכם.

- **הארכות יינתנו אך ורק במקרים חריגים (מילואים, אבל על קרובים ומחלה חריפה!) ובצרוף**

אישורים מתאימים! כמו כן במקרה של ידע מוקדם חובה ליצור קשר עם המרצה האחראית

על התרגיל לפחות יומיים לפני חלוף הדד-ליין!

- **ההגשה היא עד התאריך האחרון לתרגיל: יום בי 6.06.15 בשעה 23:40. כל יום איחור**

בהגשת העבודה יוביל להורדה של 5 נקודות מציון העבודה. לאחר 4 ימי איחור לא תתאפשר

כלל הגשת העבודה.