

תכנות מתקדם – 150024

תרגיל בית מספר 6

חריגות וסטטי

שים לב:

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

בשאלה זו עליך לכתוב תכנית לניהול כספומט. בצע את השלבים הבאים, עד לתכנית המלאה.

בכל מצבי השגיאה האפשריים יש לזרוק חריגה מתאימה ולהמשיך לפעולה הבאה. רשימת ההודעות לחריגה מפורטת בהמשך לגבי כל אחד מחלקי התכנית. הנחיה כללית: כתוב כל קטע קוד במלואו עבור מצב תקין, ורק לאחר מכן הוסף את זריקת החריגות במקומות המתאימים. ככלל – זריקת החריגות מתבצעת ע"י הפונקציות, ותפיסתן רק ע"י התכנית הראשית. שים לב שעליך לשנות בהתאם (כלומר – להוסיף תפיסת חריגות וכו') גם את התכנית הראשית.

תזכורת: אם זורקים מחרוזת מסוג rvalue (ערך הניתן להשמה במשתנה אחר, אך שלא ניתן לבצע אליו השמה - לדוגמא: "Wrong time format") אז חובה לתפוס אותה במשתנה מסוג `const char *`

חלק א

הגדר מחלקה **Clock** עבור ייצוג שעה.

א. המחלקה תכלול את התכונות הבאות:

- hour** - השעה (0 עד 23)
- minute** - מספר הדקות (0 עד 59)
- second** - מספר השניות (0 עד 59)

יש להגדיר את תכונות המחלקה בזמן הגדרתן עם ערכי ברירת מחדל 0

ב. הוסף למחלקה לפחות את הבנאים הבאים :

- **empty constructor** (בנאי ריק) – מאתחל את כל השדות ב-0
- **assignment constructor** (בנאי עם פרמטרים) - מקבל ערכים ומציב בתכונות.
- במידה שאפילו אחד מהערכים שגויים אז
- **כל התכונות** של המחלקה יישארו עם ערכי ברירת מחדל.
- תיזרק הודעה מתאימה (הבנאי יזרוק את החריגה, שתתפס בתוכנית הראשית). במקרה זה תודפס רק הודעה אחת אפילו אם יש יותר משגיאה אחת.
- הבנאי יבדוק את השגיאות ויזרוק שגיאות בסדר הבא :

Invalid time - number of seconds is not valid
Invalid time - number of minutes is not valid
Invalid time - number of hours is not valid

- כלומר קודם יבדוק האם יש שגיאה בשניות ואם כן תודפס הודעה מתאימה על השגיאה בשניות
- ואם אין שגיאה בשניות אז יבדוק האם יש שגיאה בדקות ואם כן תודפס הודעה מתאימה על השגיאה בדקות
- ואם אין שגיאה בשניות ודקות אז יבדוק האם יש שגיאות בשעות ואם כן תודפס הודעה מתאימה על השגיאה בשעות.

• **שאלה למחשבה – למה אין פה copy constructor (בנאי העתקה)?**

ג. הוסף למחלקה את המתודות הבאות :

- הצבה ואחזור (get/set) לכל תכונה. עבור כל set, במידה שהתקבל ערך שגוי יש להשאיר את התכונה בלי לשנות אותה – אבל כן יש צורך לזרוק שגיאה כך שהתוכנית הראשית תדפיס הודעה מתאימה.
- **operator+=** המקבל כפרמטר מספר שניות מקדם את הזמן בהתאם
- **operator<<** להדפסת הזמן בפורמט hh:mm:ss (שים לב שגם כאשר הערכים קטנים מ-10 יש להדפיס בפורמט של 2 ספרות).
- **operator>>** עבור קלט בפורמט המתאים. במידה והקלט אינו תקין תשלח הודעת חריגה ובאובייקט יוצב הערך 00:00:00.

בכל מקרה של שגיאה תיזרק חריגה מתאימה לפי הניסוח בפירוט הבא :

Invalid time - number of seconds is not valid
Invalid time - number of minutes is not valid
Invalid time - number of hours is not valid

שימו לב : במקרה של זמן – קולטים קודם את כל 3 הערכים (שעות, דקות ושניות) ורק אחר כך בודקים האם תקין או האם זורקים חריגה.

חלק ב

הגדר מחלקה **Account** עבור ייצוג חשבון בנק.

א. המחלקה תכלול את התכונות הבאות:

- **accountNumber** - מספר החשבון - מספר חיובי
- **code** - קוד סודי בן 4 ספרות (הספרה השמאלית ביותר אינה 0).
- **balance** - היתרה בחשבון.
- **email** - כתובת מייל של בעל החשבון - מטיפוס `string` – כתובת תקינה מכילה `@` ולפניו רצף של תווים ללא רווחים ולאחריו רצף תווים ללא רווחים עם אחת מהסיומות הבאות: `.com` או `.co.il` (יש להשתמש במחלקת `string` המוגדרת ב-STL)
(<https://www.cplusplus.com/reference/string/string>)

ב. הוסף/י למחלקה לפחות את הבנאים הבאים:

- **empty constructor** (בנאי ריק) – המציב 0 בכל התכונות ומחרוזת ריקה בתכונת המייל.
- **assignment constructor** (בנאי עם פרמטרים) - מקבל ערכים ומציב בתכונות, במידה שאפילו אחד מהערכים שגויים אז
- **כל התכונות** של המחלקה יישארו עם ערכי ברירת מחדל.
- תיזרק הודעה מתאימה (הבנאי יזרוק חריגה שתתפס בתוכנית הראשית).
- במידה ויש כמה שגיאות תיזרק רק החריגה הראשונה שתתגלה לפי הסדר הבא:

1. אם יש שגיאה במספר חשבון אז יודפס:
ERROR: account number must be positive
2. אם יש שגיאה בקוד סודי אז יודפס:
ERROR: code must be of 4 digits
3. ואם אין שגיאה בקוד סודי אבל יש שגיאה במייל שחסר סימן `@` אז יודפס:
ERROR: email must contain @
4. ואם אין שגיאה בקוד סודי וגם אין שגיאה עם הסימן `@` אבל יש שגיאה במייל כי סיומת שגויה אז יודפס:
ERROR: email must end with .com or .co.il

ג. הוסף למחלקה את המתודות הבאות:

- i. פונקציות אחזור (`get`) לכל תכונות המחלקה. אין דרך לשנות את תכונות המחלקה אחרי שהתקבלו ערכים על ידי העברת נתונים לבנאים, (אלא ע"י אופרטור הקלט) לכן אין צורך בפונקציות של `set`.
- ii. **>> operator** - עבור קלט של נתוני החשבון: מספר חשבון, קוד סודי, וכתובת מייל. בשלב זה אין לקלוט את היתרה אלא להציב יתרה הראשונית 0.
במידה ויקלטו ערכים שגויים (אחד או יותר), יש לטפל לפי ההנחיות שמפורטות ב-
`assignment constructor`
שימו לב: במקרה של הכנסת פרטי חשבון –
קולטים קודם את כל 3 הערכים (מספר חשבון, קוד סודי וכתובת מייל)
ורק אחר כך בודקים האם תקין או האם זורקים חריגה.

- **withdraw(int nis)** – מתודה למשיכת מזומנים בסכום של `NIS` ש"ח (ש). בפעולה אחת ניתן למשוך מזומנים עד לסכום של 2500 ₪, עם מסגרת אשראי של 6000 ₪ (כלומר תאפשר הגעה למינוס של עד 6000 ₪)

קודם יש לבדוק האם יש מינוס של יותר מ 6000 ₪ ובמידה שכן זורק חריגה, אחר כך יש לבדוק האם הסכום מעל 2500 ₪ במידה שכן זורק חריגה. אם הכל תקין אז ממשיך עם הפעולה (במידה ויש חריגה - לא מתבצעת משיכה, כמובן).

- **deposit(int)** – מתודה המאפשרת להפקיד צ'קים בחשבון בסכום שאינו עולה על 10000 ₪. במידה שסכום הצ'ק מעל 10000 ₪ אז נזרקת חריגה וההפקדה לא מתבצעת.

בנוסף, המחלקה תכלול תכונות ומתודות סטטיים:

- **sumWithdraw** – תכונה: מונה של סכום כל המשיכות מהבנק (מכל החשבונות).
- **sumDeposit** – תכונה: מונה של סכום כל ההפקדות בהמחאה (מכל החשבונות).
- **getSumWithdraw()** – מתודה המחזירה את הסכום כל המשיכות מכל החשבונות יחד.
- **getSumDeposit()** – מתודה המחזירה את הסכום כל הצ'קים שהופקדו עד עכשיו.

בכל פעם שמתעוררת בעיה יש לזרוק הודעת שגיאה מתאימה.
להלן נוסח ההודעות:

אסור להפקיד יותר מ 10000 NIS //1000
אסור למשוך יותר מ 2500 NIS //2500
אסור מינוס יותר מ 6000 NIS //6000

חלק ג

בסעיף זה עליך להשלים את הקוד החסר כך שהתוכנית תבצע פעולות של כספומט, ניתן להניח שבבנק יש סה"כ 10 חשבונות.

התוכנית ראשית החלקית שלפניך אמורה להשתמש במחלקות שכתבת בסעיפים א' וב' ולבצע את הפעולות הבאות:

- לקלוט את הזמן הנוכחי ולהדפיס אותו
- לקלוט מהמשתמש את פרטי 10 חשבונות הבנק – במידה והתקבל קלט שגוי יש להדפיס הודעת שגיאה מתאימה ולקלוט מחדש את פרטי החשבון.
- לאחר מכן התוכנית תציג למשתמש תפריט אפשרויות ותבצע את הנדרש לפי בחירות המשתמש.
- בסיום כל פעולה יש לקדם את הזמן ב – 40 שניות.

עליך להשלים את החלקים החסרים, שים לב שכל החריגות נתפסות רק בתוכנית הראשית ולכן הקפד להוסיף try-catch במקומות הנכונים.
שימו לב:

- לפני קליטת מספר חשבון יש להדפיס **please enter account number**
- לפני קליטת קוד יש להדפיס **please enter the code**
- לפני קליטת סכום להפקדה יש להדפיס **enter the amount of the deposit**
- לפני קליטת סכום למשיכה יש להדפיס **enter the amount of money to withdraw**
- במידה שיש שגיאה במספר חשבון יש לזרוק את חריגה **ERROR: no such account number**
- במידה שיש שגיאה בקוד יש לזרוק את חריגה **ERROR: wrong code**

שימו לב: במקרה של קלט בתוכנית הראשית קולטים כל נתון בנפרד. אם יש שגיאה אז זורקים חריגה ומתחילים מחדש.

```

#include <iostream>
#include "Clock.h"
#include "Account.h"

using namespace std;

enum ACTION {
    EXIT,
    BALANCE,
    DEPOSIT,
    WITHDRAW,
    SUM_DEPOSIT,
    SUM_WITHDRAW
};

ACTION menu() {
    cout << "enter 1 to get account balance" << endl;
    cout << "enter 2 to deposit money" << endl;
    cout << "enter 3 to withdraw money" << endl;
    cout << "enter 4 to see the sum of all deposits" << endl;
    cout << "enter 5 to see the sum of all withdrawals" << endl;
    cout << "enter 0 to exit" << endl;
    int x;
    cin >> x;
    return (ACTION)x;
}

void printTransaction(Account a, ACTION ac, Clock& c) {
    cout << c << "\t";
    switch (ac) {
        case BALANCE:
        case DEPOSIT:
        case WITHDRAW: cout << "account #: " << a.getAccountNumber() << "\t";
            cout << "balance: " << a.getBalance() << endl;
            break;
        case SUM_DEPOSIT:
            cout << "sum of all deposits: " << Account::getSumDeposit() << endl;
            break;
        case SUM_WITHDRAW:
            cout << "sum of all withdrawals: " << Account::getSumWithdraw() << endl;
            break;
    }
}

void getBalance(Account* bank, int size, Clock& c) {
    //TODO:
    הפונקציה קולטת מספר חשבון וקוד. במידה שיש שגיאה אז הפונקציה תזרוק שגיאה
    מדפיסה את היתרה בחשבון עם הזמן הנוכחי על ידי קריאה לפונקציה printTransaction
}

void cashDeposit(Account* bank, int size, Clock& c) {
    //TODO:
    הפונקציה קולטת מספר חשבון, קוד וסכום להפקדה ומעדכנת את היתרה בחשבון המבוקש.
    במידה שיש שגיאה אז הפונקציה תזרוק שגיאה
    בסיום הפעולה יש לקרוא לפונקציה printTransaction
}

void cashWithdraw(Account* bank, int size, Clock& c) {
    //TODO:
    הפונקציה קולטת מספר חשבון, קוד וסכום למשיכה ומעדכנת את היתרה בחשבון המבוקש.
    במידה שיש שגיאה אז הפונקציה תזרוק שגיאה

```

בסיום הפעולה יש לקרוא לפונקציה printTransaction

```
}

int main() {
    const int SIZE = 10;

    Clock present;
    cout << "enter current time";
    cin >> present;
    cout << "current time is: " << present << endl;

    Account bank[SIZE];
    cout << "enter account number, code and email for " << SIZE << "
accounts:\n";

    /** חסר פה קליטה של 10 חשבונות בנק */

    ACTION ac = menu();
    while (ac) {
        switch (ac) {
            case BALANCE: getBalance(bank, SIZE, present);
                          break;
            case WITHDRAW: cashWithdraw(bank, SIZE, present);
                          break;
            case DEPOSIT: cashDeposit(bank, SIZE, present);
                          break;
            case SUM_DEPOSIT:
                          printTransaction(bank[0], SUM_DEPOSIT, present);
                          break;
            case SUM_WITHDRAW:
                          printTransaction(bank[0], SUM_WITHDRAW, present);
        }
        present += 40;
        ac = menu();
    }

    return 0;
}
```

דוגמאה להרצת התכנית (תוכנית עם רק 3 חשבונות ובלי שגיאות):

```
enter current time10:08:12
current time is: 10:08:12
enter account number, code and email for 3 accounts:
123 4444 me@gmail.com
234 5555 you@walla.co.il
345 6666 us@g.com
enter 1 to get account balance
enter 2 to deposit money
enter 3 to withdraw money
enter 4 to see the sum of all deposits
enter 5 to see the sum of all withdrawals
enter 0 to exit
2
please enter account number:
234
please enter the code:
5555
enter the amount of the deposit:
5000
10:08:12    account #: 234 new balance: 5000
enter 1 to get account balance
enter 2 to deposit money
enter 3 to withdraw money
enter 4 to see the sum of all deposits
enter 5 to see the sum of all withdrawals
enter 0 to exit
3
please enter account number:
234
please enter the code:
5555
enter the amount of money to withdraw:
1000
10:08:52    account #: 234 new balance: 4000
```

```
enter 1 to get account balance
enter 2 to deposit money
enter 3 to withdraw money
enter 4 to see the sum of all deposits
enter 5 to see the sum of all withdrawals
enter 0 to exit
1
please enter account number:
234
please enter the code:
5555
10:09:32    account #: 234  balance: 4000
enter 1 to get account balance
enter 2 to deposit money
enter 3 to withdraw money
enter 4 to see the sum of all deposits
enter 5 to see the sum of all withdrawals
enter 0 to exit
2
please enter account number:
345
please enter the code:
6666
enter the amount of the deposit:
2000
10:10:12    account #: 345  new balance: 2000
enter 1 to get account balance
enter 2 to deposit money
enter 3 to withdraw money
enter 4 to see the sum of all deposits
enter 5 to see the sum of all withdrawals
enter 0 to exit
```



```
3
please enter account number:
345
please enter the code:
6666
enter the amount of money to withdraw:
500
10:10:52    account #: 345 new balance: 1500
enter 1 to get account balance
enter 2 to deposit money
enter 3 to withdraw money
enter 4 to see the sum of all deposits
enter 5 to see the sum of all withdrawals
enter 0 to exit
4
10:11:32    sum of all deposits: 7000
enter 1 to get account balance
enter 2 to deposit money
enter 3 to withdraw money
enter 4 to see the sum of all deposits
enter 5 to see the sum of all withdrawals
enter 0 to exit
5
10:12:12    sum of all withdrawals: 1500
enter 1 to get account balance
enter 2 to deposit money
enter 3 to withdraw money
enter 4 to see the sum of all deposits
enter 5 to see the sum of all withdrawals
enter 0 to exit
0
```

בהצלחה!!