המחלקה להנדסת מערכות תקשורת
אוני' בן גוריון בנגב

הנחיות הגשה:

1. ההגשה מתבצעת בזוגות ועל כל אחד מבני הזוג **לבצע את ההגשה בנפרד במודל**.
2. יש להגיש את כל קבצי העבודה במערכת ה VPL ולוודא שאין שגיאות קומפילציה במערכת.
3. כל מחלקה תכלול קובץ header וקובץ CPP (מלבד מחלקות Array).
4. הקפידו על הוראות ברורות וקריאות למשתמש לפני כל פעולת קלט.
5. יש להקפיד של שמות המחלקות והשיטות כפי שמצוין בעבודה (הקפידו על אותיות גדולות וקטנות).
6. הקפידו על תכנות נכון, זהו חלק מן הציון:
a. • שמות משתנים בעלי משמעות.
b. • שימוש חוזר בקוד.
c. • הקפדה על הזחות.
d. • שימוש נרחב בהערות באנגלית בלבד. הקפידו לתעד כל שיטה וכל קטע קוד משמעותי.
7. בראש כל קובץ יש להוסיף בהערה את הטקסט הבא:

```
*/Assignment C++: 1
  Author: Israel Israeli, ID: 01234567
/*
```

8. עליכם להחליף את הפרטים בפרטים שלכם.
9. כל שאלה יש להפנות לפורום המתאים לשאלות במודל.
10. הארכות יינתנו רק במקרי מילואים, אבל ומחלה חריפה ובצירוף אישורים מתאימים.

תרגיל מס' 2

נושא התרגיל: הורשה, פולימורפיזם , exceptions

<u>הערה חשובה</u>: רשימת המתודות היא **<u>חלקית</u>** בלבד.
היא מהווה את המינימום הנדרש ממכם. ייתכן ואף סביר להניח שתצטרכו להוסיף מתודות נוספות עבור המחלקות המתוארות.
לכל המחלקות חובה להוסיף Constructor ו- Destructor דיפולטיבים (אפילו אם הוא ריק).
**** שימו לב - בעבודה זו עליכם לוודא כי ערכי הקלט נמצאים בטווח המתאים. הטיפול יעשה באמצעות exceptions(אין צורך לבדוק טיפוסים), במידה ובוצעה פעולה לא תקינה בפונקציה מסויימת יש לזרוק חריגה. יש לטפל בחריגות בקוד ע"י הדפסת הודעה מתאימה ולאפשר למשתמש להזין את הנתונים בשנית.**

1. <u>מחלקת template –Array<T></u>

- <u>שדות:</u>
  - int – כמות איברים במערך.
  - *T - מצביע לאיבר הראשון במערך.
- <u>מתודות :</u>
  - insert – מוסיפה איבר בסוף המערך.
  - remove – מקבלת אינדקס של איבר ומוחקת אותו, על מנת לאפשר שחרור זיכרון תקין (במידה והוא מכיל כתובות), הפונקציה תחזיר את הערך שנמחק T.
    ** בסוף הפעולה נותר מערך בגודל size-1.
  - getSize – מחזירה את גודל המערך.

  <u>בנאים:</u>
  - בנאי ברירת מחדל.

  <u>אופרטורים:</u>
  - >> - הדפסת כל האיברים במערך.
  - [ ] – array index operator.

2. <u>Banking System:</u>

במשימה זו עליכם לממש מערכת בנק פשוטה.

- **<u>Account</u>**
- <u>שדות:</u>
  - accountNumber (string) – מספר החשבון.
  - accountHolderName (string) – שם בעל החשבון.
  - Balance (double) – יתרת החשבון.
- <u>מתודות:</u>
  - getBalance – מחזירה את השדה Balance המעודכן (לאחר עדכון ריבית במחלקות הרלוונטיות).
- <u>מתודות אבסטרקטיות:</u>
  - virtual void deposit(double amount) – מפקיד סכום בחשבון.
  - virtual void withdraw(double amount)– משיכת סכום מהחשבון.
- <u>בנאים:</u>
  - Account(const string& number, const string& holder, double initialBalance) - בנאי המקבל את כל הפרמטרים ומאתחל בהתאם.
- <u>אופרטורים:</u>
  - >> - הדפסת פרטי החשבון.

SavingsAccount:

מחלקה זו יורשת מ Account. מטרת מחלקה זו היא לייצג חשבון חסכון אשר צובר ריבית.

- שדות:
  - annualInterestRate (double)– ריבית שנתית.
  - lastTransactionTime (std::chrono::steady_clock::time_point)– זמן הפעולה האחרונה (הפקדה או משיכה או בירור יתרה).
- מתודות:
  - deposit – מעדכנת את הריבית על החסכון הקיים ואז מפקידה את הסכום הנדרש.
  - withdraw – מעדכנת את הריבית על החסכון הקיים ואז מבצעת משיכה מסכום הנדרש.
- בנאים:
  - SavingsAccount(const std::string& number, const std::string& holder, double initialBalance, double annualRate) - בנאי המקבל את כל הפרמטרים ומאתחל בהתאם.

CheckingAccount

מחלקה זו יורשת מ Account. מטרת מחלקה זו היא לייצג חשבון עובר ושב.

- שדות:
  - overdraftLimit (double) – מגבלת חריגה מיתרה.
- מתודות:
  - deposit – מפקידה את הסכום הנדרש.
  - withdraw – בצעת משיכה מסכום הנדרש.
- בנאים:
  - CheckingAccount(const string& number, const string& holder, double initialBalance, double overdraft); - בנאי המקבל את כל הפרמטרים ומאתחל בהתאם.

דגשים:

1) במשימה זו אתם רשאים להשתמש בספריות אלו:

```
<thread>, <chrono>, <cstring>
```

2) בחשבון החסכון, יתבצע עדכון ריבית לפני כל פעולת משיכה, הפקדה או בירור יתרה.
חישוב הריבית שיש להוסיף לחשבון יתבצע לפי **שניות שלמות ועל ידי מנגנון ריבית דריבית** ויחושב כך:

$$A_t = A_0(1+r)^t$$

כאשר A_t הינו הסכום המצטבר לאחר t שניות.
A_0 הסכום לפני תוספת הריבית.
r הריבית בחשבון (לדוגמא: 0.05 הוא 5% ריבית ליחידת זמן).
t הפרש הזמנים בין הפעולה האחרונה לזמן הנוכחי(מעוגל כלפי מטה).

לדוגמא:

במקרה בו קיים בחשבון 100 דולר מזמן $t_0 = 0$ וביקשנו להפקיד 100 דולר נוספים לאחר 2.8 שניות.
הבנק יתייחס רק לחלק השלם של הזמן שעבר (2). לכן לאחר עדכון הכסף וההפקדה במקרה בו $r = 0.05$, היתרה בחשבון תהיה $A = 100(1+0.05)^2 + 100 = 210.25$.

Code Example:

```
#include <iostream>
#include <thread>
#include "Account.h"
#include "SavingsAccount.h"
#include "CheckingAccount.h"
#define SLEEPDUR 2
#define RATE 0.05
int main() {
          CheckingAccount checkings("CA1010", "John Doe", 100, OVERDRAFT);
          SavingsAccount savings("SA2020", "Yossi Yossi", 100, RATE);
          std::this_thread::sleep_for(std::chrono::seconds(SLEEPDUR));
          savings.deposit(100); // print the updated balance
          std::cout << '\n' << savings.getBalance() << std::endl;
          std::cout << '\n' << checkings.getBalance() << std::endl;
          return 0;
       }
```

```
Account Number: SA2020
Account Holder: Yossi Yossi
New Balance: $210.25

210.25

100
```

3. מחלקת תפריט – Menu

מחלקה זו תנהל את המערכת. על מחלקה זו לבצע פעולות קלט/פלט מהמשתמש. יש להציג את התפריט בלולאה עד שהמשתמש יבחר לסיים אותה.

שדות:

bankAccounts - מסוג Array< Account*>.

שיטות:

mainMenu() – שיטה זו תדפיס את התפריט הבא:
המתודה תציג את התפריט הבא למשתמש.

המשתמש מקיש	שם הפעולה	הערות
1	הוספת חשבון חדש למערך	
2	הפקדה לחשבון	לפי אינדקס וכמות
3	משיכה מחשבון	לפי אינדקס וכמות
4	מחיקת איבר במערך לפי אינדקס.	
5	הדפסת פרטי החשבונות במערך.	
6	יציאה מהתוכנית	יש לשים לב שמשחררים את כל ההקצאות הדינמיות שנעשו ב main

הערות:

1. הקלטים יהיו מהטיפוסים החוקיים. ז"א בכל מקום נכניס את הטיפוס המתאים. אנחנו לא מתחייבים שהוא יהיה בטווח מסוים – אלא אם כן נאמר אחרת בשאלה עצמה.
2. אחרי כל הדפסה יש לבצע ירידת שורה.

יש להקפיד על תכנות נכון:

a. כל הערכים שהם קבועים (מבחינה לוגית הם לא אמורים להשתנות), חייבים להיות מוגדרים כ: const, define או, enum בהתאם לצורך.
b. יש לרשום הערות בשפע, ובאנגלית בלבד (לכל מחלקה מה התפקיד שלה, התוכנית 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, הממשק ב-public, חלוקה לקבצים.
d. יש להקפיד על הזחות, כיתוב נכון וקריא, ושמות משמעותיים.
e. יש לנסות ולייעל את הקוד והתוכנית ככל שניתן. הקפידו על reuse בקוד . נקודות ירדו על קוד כפול !
f. לפני בקשת קלט, יש להדפיס למשתמש הוראה (איזה קלט מבוקש)
g. להזכירכם: הנכם נדרשים לתכנת בשפת ++C אי-לכך, כל שימוש בפונקציות וספריות של שפת C אסורה.
3. עבור הצלחת הבדיקות עליכם לוודא שהדפסות זהות לקובץ ההרצה שקיבלתם. (לא יורדו נקודות על שוני ברווחים ושורות).
4. בהצלחה!