

מס' תלמיד:
קמפוס:
למילוי ע"י הסטודנט



**שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב ++C  
מספר קורס: 150018**

- **שם המרצה:** גרינברג נורית, ווגל ג'ויס, ויזן אריה, חגיז מורדי מרדכי, חדד אורטל, חייט אלכסנדר, טרנר אורי, יטיב אריה, ליטמן מאיר, מילסטון עדינה, נדלר חני, עמר אפרת, קדרון דוד, רוזנבליט אורית, רז שרה, שנדוביץ חיים
- **תאריך הבחינה:** כ"ח בסיוון תשפ"ב 27 / 07 / 2022
- **משך הבחינה (בדקות):** 180
- **חומר עזר מותר לשימוש:** אין
- **מחשבון:** אסור

המבחן כולל שלושה חלקים עפ"י הפירוט הבא:

- חלק א':** 7 שאלות סגורות. יש לענות על 6 מתוכן. כל שאלה 7 נקודות, סה"כ 42 נקודות.
  - חלק ב':** 3 שאלות פתוחות קצרות. יש לענות על כולן. כל שאלה 10 נקודות, סה"כ 30 נקודות.
  - חלק ג':** 2 שאלות פתוחות. יש לענות על כולן. כל שאלה 16 נקודות, סה"כ 32 נקודות.
- הציון המקסימלי: 100**

- את התשובות יש לכתוב **בגוף המבחן** בלבד (תשובות במחברת או בדפי טיוטא לא ייבדקו).

**תלמיד יקר,**

1. **נוהל הבחינות של המרכז האקדמי לב מחייב אותך**, באחריותך לקוראו ולהכירו - בחינה עלולה להיפסל על כל חריגה מהנוהל.
2. אם אינך מבין את כוונת המרצה בשאלה כלשהי, עליך לכתוב בראש התשובה כיצד הינך מבין את השאלה ולפתור בהתאם. המרצה ישקול האם יש מקום להבנה זו ואז ינקד בהתאם.
3. **חובה להחזיר את השאלון**. מחברת שלא יצורף לה השאלון, לא תיבדק!
4. לידיעתך, תורדנה נקודות לא רק על שגיאות, אלא גם על תוספות לא רלוונטיות, העדר נימוק הולם לתשובה, חוסר סדר ותשובה דו-משמעית, כאשר נדרשת תשובה חד משמעית.

**בהצלחה רבה !**

לשימוש הבודקים בלבד:

חלק א	חלק ב			חלק ג		סה"כ
1 - 7	8	9	10	11	12	

שנה"ל תשפ"ב, סמסטר ב', מועד ב'

שאלון בחינה בקורס: סדנא ב C++

מספר קורס: 150018

שאלון בקורס סדנא ב- C++:

### חלק א'

לפניך 7 שאלות קצרות. יש לענות על 6 מתוכן, 7 נקודות לכל שאלה (סה"כ 42 נקודות).  
סמן בצורה ברורה איזו שאלה לא לבדוק ע"י מחיקה ב X של כל התשובות לשאלה זו.  
במידה ותענה על כל השאלות, ייבדקו ויחושבו לציון הסופי 6 מתוכן באופן אקראי.

### שאלה מספר 1:

לפניך הגדרה של המחלקה Cat :

```
class Cat
{
public:
    Cat() { age = 0; }
    int getAge()const { return age; }

private:
    int age;
};
```

נתונה תוכנית ראשית המשתמשת במחלקה Cat. מטרת התוכנית היא להדפיס את כל המספרים מ 0 עד 6 (כולל 0 וכולל 6) :

```
int main() {
    Cat a;
    for (Cat c=a; c.getAge() < 7; c++) {
        cout << c.getAge() << endl;
    }
}
```

איזו מתודה חובה להוסיף למחלקת Cat כדי שהתוכנית תעבור קומפילציה ותבצע את הנדרש?

א. bool operator < (const int &) const

ב. void operator++ ( int )

ג. Cat & operator= ( const Cat &)

ד. בנאי העתקה

ה. יש יותר מתשובה אחת נכונה.

שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב C++  
מספר קורס: 150018

שאלה מספר 2:

לפניך תוכנית:

```
#include <iostream>
using namespace std;

class Flower {
public:
    Flower() { cout << "Empty Flower " << endl; }
    Flower(const Flower& other) { cout << "CC Flower" << endl; }
    Flower(Flower&& other) { cout << "Move Flower" << endl; }
    virtual ~Flower() { cout << "Dest Flower" << endl; }
};

class Petal_Flower : public Flower {
public:
    Petal_Flower() { cout << "Empty Petal " << endl; }
    Petal_Flower(const Petal_Flower& other) { cout << "CC Petal" << endl; }
    Petal_Flower(Petal_Flower&& other) { cout << "Move Petal" << endl; }
    virtual ~Petal_Flower() { cout << "Dest Petal" << endl; }
};

Flower func(Flower a)
{
    cout << "func\n";
    return a;
}

int main()
{
    Petal_Flower pf;
    func(pf);
    cout << "main\n";
    return 0;
}
```

מה יהיה פלט התוכנית לאחר הרצת התוכנית? (בעמוד הבא)

שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב C++  
מספר קורס: 150018

ה.	ד.	ג.	ב.	א.
Empty Flower Empty Petal CC Flower func CC Flower Dest Flower main Dest Petal Dest Flower	Empty Flower Empty Petal func Move Flower main Dest Flower Dest Petal	Empty Flower Empty Petal CC Flower func CC Flower Dest Flower Dest Flower main Dest Petal Dest Flower	Empty Petal Empty Flower CC Flower func Move Flower Dest Flower Dest Flower main Dest Flower Dest Petal	Empty Flower Empty Petal CC Flower func Move Flower Dest Flower Dest Flower main Dest Petal Dest Flower

שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב C++  
מספר קורס: 150018

שאלה מספר 3:

לפניך רשימת טיעונים. איזה מהטיעונים אינו נכון?

- א. בשפת C++, ההבדל בין הגדרה של פונקציה וירטואלית טהורה (abstract - abstract) במחלקה, לבין הגדרה של פונקציה וירטואלית רגילה במחלקה, הוא שלפונקציה הווירטואלית הטהורה נכתב הערך 0 = לאחר ההצהרה, ולפונקציה הווירטואלית הרגילה לא.
- ב. מחלקה מופשטת (אבסטרקטית) היא מחלקה שלא ניתן לייצר אובייקטים מהטיפוס שלה.
- ג. בשפת C++, מספיקה הגדרה אחת של פונקציה וירטואלית טהורה יחידה כדי להפוך את המחלקה למחלקה מופשטת.
- ד. מחלקת האב מופשטת (אבסטרקטית), וממנה ירשה מחלקת הבן. אם רוצים ליצור מצביע מהסוג של מחלקת הבן, חובה לממש במחלקת הבן את כל הפונקציות הווירטואליות הטהורות שירשה.
- ה. בעזרת מחלקה מופשטת ניתן ליישם את רעיון הפולימורפיזם של מבנה נתונים.

שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב C++  
מספר קורס: 150018

שאלה מספר 4:

נתונה התוכנית הבאה:

```
class A {
public:
    static int num;

    A() { num++; }
};
int A::num = 0;

class B {
public:
    static int num;
    B() { num++; }
};
int B::num = 0;

int main() {
    A a[5];
    B* b[5];
    cout << A::num << " " << B::num << endl;
}
```

מה יהיה פלט התוכנית לאחר הרצת התוכנית (הפלט משמאל לימין)?

א.

10 10

ב.

1 1

ג.

0 0

ד.

5 5

ה.

5 0

שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב C++  
מספר קורס: 150018

שאלה מספר 5:

נתונה התוכנית הבאה:

```
#include <iostream>
using namespace std;

class Father
{
protected:
    virtual void printVal()
    {
        cout << "ffff : " << endl;
    }
public:
    virtual void print()
    {
        printVal();
        cout << "I am a Father" << endl;
    }
};

class Son :public Father
{
    void printVal()
    {
        cout << "ssss : " << endl;
    }

    void print()
    {
        Father::print();
        cout << "I am a Son" << endl;
    }
};

int main()
{
    Father* family[2];
    family[0] = new Father();
    family[1] = new Son();
    for (int i = 0; i < 2; i++)
        family[i]->print();
    return 0;
}
```

שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב C++  
מספר קורס: 150018

סמן את התשובה הנכונה, לגבי הרצת התוכנית:

א. התוכנית לא תעבור קומפילציה.

ב. התוכנית תעבור קומפילציה ותדפיס בזמן הריצה:

```
ffff:  
I am a Father  
ssss:  
I am a Father  
I am a Son
```

ג. התוכנית תעבור קומפילציה ותדפיס בזמן הריצה:

```
ffff:  
I am a Father  
ffff:  
I am a Father
```

ד. התוכנית תעבור קומפילציה ותדפיס בזמן הריצה:

```
ffff:  
I am a Father  
ffff:  
I am a Father  
I am a Son
```

ה. התוכנית תעבור קומפילציה ותדפיס בזמן הריצה:

```
ffff:  
I am a Father
```

ולאחר מכן התוכנית תעוף בזמן הריצה.



שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב C++  
מספר קורס: 150018

שאלה מספר 6:

נתונה התוכנית הבאה – תוכנית ראשונה, הבונה ומשתמשת בקובץ :fout.bin.  
הריצו את התוכנית, והתוכנית רצה בלי שגיאות בזמן ריצה.

```
//תוכנית ראשונה
#define _CRT_SECURE_NO_WARNINGS

#include<iostream>
#include<fstream>
using namespace std;

struct Restaurant
{
    char * name;
    int rating;
};

int main()
{
    ofstream fout("fout.bin");
    Restaurant restaurant;
    restaurant.rating = 10;
    restaurant.name = new char[20];
    strcpy(restaurant.name, "hapisga");
    fout.write((char*)&restaurant, sizeof(Restaurant));
    fout.close();
    return 0;
}
```

לאחר מכן - הריצו את התוכנית הבאה – תוכנית שנייה, המשתמשת גם היא בקובץ :fout.bin :

```
//תוכנית שנייה
#define _CRT_SECURE_NO_WARNINGS

#include<iostream>
#include<fstream>
using namespace std;

struct Restaurant
{
    char* name;
    int rating;
};

int main()
{
    ifstream fin("fout.bin");
    Restaurant rest;
    fin.read((char*)&rest, sizeof(Restaurant));
    cout << rest.name << endl;

    fin.close();

    return 0;
}
```

שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב C++  
מספר קורס: 150018

סמן את התשובה הנכונה, לגבי הרצת התוכנית השניה:

א. הקוד לא יעבור קומפילציה, כי לאחר שייצרנו קובץ לצורך כתיבה (בתוכנית הראשונה), לא ניתן לפתוח את אותו הקובץ לקריאה.

ב. הקוד תקין. הוא יעבור קומפילציה וירוך, וידפיס על המסך hapisga

ג. הקוד יעבור קומפילציה, אך לא ידפיס על המסך hapisga (ואולי אף יעוף בזמן הריצה).

ד. הקוד יעבור קומפילציה ויציא hapisga על המסך, אך יעוף לאחר מכן.

ה. הקוד לא יעבור קומפילציה, כי כשקראנו מהקובץ אסור היה לכתוב sizeof (Restaurant) היה צריך לכתוב sizeof (rest).

שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב C++  
מספר קורס: 150018

שאלה מספר 7:

נתונה הפונקציה  $f$  הבאה, המקבלת מספר כפרמטר ומחזירה ערך מספרי:  
(הבהרה: הערך  $0x00000001$  הוא ביטוי בבסיס הקסדצימלי של הערך 1 העשרוני. המשמעות זהה – כלומר  
היה ניתן לכתוב כאן:  $1 \mid x$ , ולקבל תוכנית זהה לחלוטין).

```
int f(int x)
{
    unsigned int y = x | 0x00000001;
    y <= 31;
    return y;
}
```

סמן את התשובה הנכונה:

א. הפונקציה תחזיר תמיד את הערך  $0x80000000$  (שיתורגם ל  $\text{int}$ , כמובן), עבור כל מספר שהיא מקבלת כפרמטר.

ב. הפונקציה מחזירה ערך שונה מ-0 כאשר הפרמטר  $x$  הוא מספר אי זוגי, אחרת מחזירה 0.

ג. לפונקציה שגיאת קומפילציה (הידור).

ד. הפונקציה תחזיר תמיד את הערך 31.

ה. הפונקציה תחזיר 1 כאשר הפרמטר  $x$  הוא שלילי, אחרת מחזירה 0.

**שנה"ל תשפ"ב, סמסטר ב', מועד ב'**  
**שאלון בחינה בקורס: סדנא ב C++**  
**מספר קורס: 150018**

**חלק ב'**

לפניך 3 שאלות פתוחות קצרות. יש לענות על כולן, 10 נקודות לכל שאלה (סה"כ 30 נקודות)

**שאלה מספר 8:**

לפניך (בעמוד הבא) קטע קוד הכולל:

- הגדרה של מחלקה (class) לשמירת מידע על סטודנט
  - שני שדות: שם הסדונט וממוצע הציונים שלו.
  - בנאי המאתחל את ערכי השדות של הסטודנט
  - מתודה המחזירה את ערכו של שדה ממוצע הציונים של הסטודנט.
- פונקציה בוליאנית בשם honors המחזירה true אם ערכו של ממוצע הציונים של הסטודנט הוא 90 ומעלה.
- פונקציה בשם init המאתחלת vector של סטודנטים (המתודה init לא מוצגת כאן, אך יש להניח שהיא פועלת כראוי).

בתוכנית הראשית, מחזיקים וקטור של סטודנטים, ומאתחלים אותו באמצעות הפעולה init.

עליך להשלים את הקוד כך:

אם - יש לפחות סטודנט אחד שקבל ציון ממוצע מושלם (ציון של 100),  
אז - התוכנית תדפיס את מספר הסטודנטים שציון הממוצע שלהם שווה ל-90 או גבוה מ-90 (ע"י שימוש בפונקציה honors)

לצורך ההשלמה – יש להשתמש באחד או יותר מהאלגוריתמים של STL הבאים:

- all\_of
- any\_of
- count
- count\_if
- for\_each

שים לב: בקטע הקוד חסרות מספר שורות המסומנות בקו. עליך להשלימן לפי ההוראות בסוף השאלה לעיל.

יש להשלים רק את פקודת ה if (עם פקודות נוספות בתוכה, כמובן – לפי ההנחיות לעיל) – כפקודה יחידה, וללא בלוק (scope) פנימי.  
אין להתייחס לאורך השורה, השתמש במספר השורות הרצוי לך.

שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב C++  
מספר קורס: 150018

```
#include<string>
#include <algorithm>
#include <vector>
#include <iostream>
using namespace std;

class Student {
private:
    string name;
    float avg;
public:
    Student(string sName, float sAvg) :avg(sAvg) { name = sName; }
    float getAvg() { return avg; }
};

bool honors(Student s) {return s.getAvg() >= 90; }

void init(vector<Student> & vec); // הפעולה תקינה, ולא מוצגת כאן

int main() {
    vector<Student> students;
    init(students);

    if _____

    _____

    _____

    _____

    _____

    _____;

    return 0;
}
```

**שנה"ל תשפ"ב, סמסטר ב', מועד ב'**

## שאלון בחינה בקורס: סדנא ב C++

## מספר קורס: 150018

**שאלה מספר 9:**

לפניך תוכנית (בעמוד הבא).

בהנחה שהקלט עבור התוכנית הינו רצף של אפסים ( $\theta = 0$  וכי לפי הצורך).

## סעיף א:

כתוב מה התוכנית תדפיס במהלך ההרצה?

(במידה והתוכנית תעורר, כתוב במקום המתאים את המילה "abort").

השתמש במספר השורות הרצוי לך.

סעיף ב:

מה יהיה הערך של  $\text{months}[0] \rightarrow \text{days}$  לקראת סיומה של התוכנית? (כלומר – בזמן ביצוע השורה האחרונה שתבצע בתוכנית)

(במידה והערך לא מאותחל – כתוב את המילה "זבל")

הערך של days -> months[0] יהיה: \_\_\_\_\_

```
#include <algorithm>
#include <iostream>
using namespace std;

enum SEASON { WINTER=1, SPRING, SUMMER, FALL };

class Month {
    int days = 30;
    SEASON type = (SEASON)SUMMER;
public:
    class dayException {};
    class seasonException {};

    Month(int myday, int mytype)
    {
        if ((myday < 28) || (myday > 31))
            throw dayException();
        if ((mytype < 1) || (mytype > 4))
            throw seasonException();
        days = myday;
        type = (SEASON)mytype;
    }
};

void initMonth(Month** months)
{
    for_each(months, months + 12, [](Month* month)
    {
        int day, type;
        cin >> day >> type;
        month = new Month(day, type);
    });
    cout << " ok\n";
}

int main() {
    try {
        Month** months = new Month * [12];
        initMonth(months);
    }
    catch (Month)
    {
        cout << "exception in month\n";
    }
    catch (Month::seasonException)
    {
        cout << "exception in season\n";
    }
    catch (Month::dayException)
    {
        cout << "exception in day\n";
    }
    catch (...)
    {
        cout << "you have a problem\n";
    }
    cout << "---we finish\n";
    return 0;
}
```

שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב C++  
מספר קורס: 150018

שאלה מספר 10:

רוצים לבנות מחלקה Calculator לצורך ביצוע 4 פעולות חשבוניות עבור שני מספרים:

- חיבור (add), הפרש (subtract), כפל (multiply), חילוק (divide).

המחלקה היא גנרית - ופועלת על סוגים שונים של מספרים (למשל: int, float ו-double), כך ששני המספרים והתוצאה של הפעולה – הם מאותו הסוג.

למשל: אם שני המספרים הם שלמים (int) אז כל אחת מהפעולות תחזיר int (כלומר - במקרה של חילוק תוחזר התוצאה כמספר שלם, ללא שארית), ואם שני המספרים הם רציונליים (float) אז כל אחת מהפעולות תחזיר float.

סעיף א:

כתוב הצהרה על מחלקה תבניתית (template class) בשם Calculator לפי הנדרש לעיל.  
(ההצהרה כוללת את השדות של המחלקה (שני המספרים) ואת הכותרות של 4 המתודות של הפעולות החשבוניות. ללא מימוש של המתודות)

סעיף ב:

כתוב מימוש של המתודה add



שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב C++  
מספר קורס: 150018

חלק ג'

לפניך 2 שאלות פתוחות. יש לענות על כולן, 16 נקודות לכל שאלה (סה"כ 32 נקודות)

שאלה מספר 11:

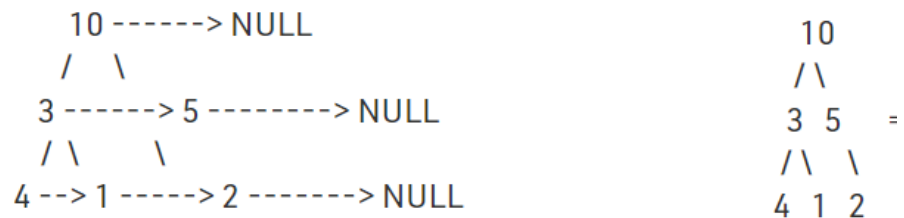
בהרצאה ראינו את מבנה הנתונים SearchTree – מבנה נתונים של עץ חיפוש בינארי המכיל נתונים גנריים. לצורך פישוט, נשתמש בשאלה זו במחלקה דומה – אך שתכיל מספרים שלמים בלבד (לא גנרית) וללא ירושה. (ההצהרה והמימוש המלאים נמצאים בנספח א של מבחן זה).  
כמו כן – מחלקת Node (צומת בעץ) תוגדר כמחלקה בפני עצמה (ולא כמחלקה פנימית של העץ).

אנו רוצים עכשיו לשנות את המחלקה כך שלכל צומת יתווסף מצביע (בנוסף למצביעים על שני הבנים), וכל הצמתים באותה רמה יצביעו אחד על השני, משמאל לימין, כשהאחרון ביניהם מצביע על nullptr. (תזכורת: רמה של צומת היא המרחק שלו מהשורש. הרמה של השורש היא 0).

לדוגמה:

יראה לאחר השינוי, כך:

עץ החיפוש הבינארי הבא:



לצורך כך נבצע במחלקה מספר שינויים:

ראשית - להגדרה של מחלקת Node נוסיף עוד 2 תכונות:

- הרמה של הצומת (מטיפוס מספר שלם) - level
  - מצביע לצומת הבא (מימין לו) באותה רמה - next
- כפי שמוגדר להלן:

```
class Node
{
public:
    Node* left;  \\ מצביע לבן שמאל
    Node* right; \\ מצביע לבן ימין
    Node* next;  \\ מצביע לצומת הבא באותה הרמה
    int value;   \\ ערך של הצומת
    int level;   \\ הרמה של הצומת

    Node(int val, int lv=0)
        : value(val), level(lv), left(nullptr), right(nullptr), next(nullptr) {}
    Node(int val, Node* l, Node* r, Node* n, int lv)
        : value(val), left(l), right(r), next(n), level(lv) {}
};
```

סעיף א:

במחלקה SearchTree כבר קיימות שתי מתודות בשם add להוספת צומת לעץ חיפוש בינארי. מתודה אחת ציבורית לתחילת הפעולה, ומתודה פרטית הפועלת בצורה רקורסיבית. יש להתאים את המתודות לשינוי שעשינו. (המשך בעמוד הבא)

עליך לשנות (באמצעות השלמת השורות המתאימות בקוד המסומנות בקו תחת) את המתודה הרקורסיבית הפרטית כך שתוסיף צומת מהסוג החדש שלעיל (הכולל רמה ומצביע לצומת הבא באותה רמה) לעץ החיפוש הבינארי. שים לב: המצביע next מאותחל לnullptr בשלב זה, ועדיין לא מצביע על הצומת הבא באותה רמה.

```
void SearchTree::add(int val)
{
    // add value to binary search tree
    if (!root)
    {
        root = new Node(val);
        return;
    }

    add(root, val, 1);
}

void SearchTree::add(Node* current, int val, int level)
{
    if (current->value < val)
    {
        if (!current->right)
        {
            current->right = new Node(val, _____);

            return;
        }
        else add(current->right, val, _____ );
    }
    else
    {
        if (!current->left)
        {
            current->left = new Node(val, _____);

            return;
        }
        else add(current->left, val, _____ );
    }
}
```

סעיף ב:

הוסף למחלקה SearchTree מתודה בשם connect. המתודה תעבור על העץ, ותחבר לרשימה מקושרת את כל הצמתים באותה רמה, עבור כל הרמות של העץ (כמו שהוסבר בתחילת השאלה). שים לב – המתודה אינה מקבלת ואינה מחזירה דבר. היא פועלת על העץ, ומעדכנת את הערכים המתאימים במצביע next שבכל צומת.

הכוונה: יש לכתוב את המתודה connect בדומה למתודה breadthScan שהתבקשת לבצע בתרגיל הבית. תזכורת: המתודה breadthScan סרקה לרוחב את צמתי העץ והדפיסה את ערכי הצמתים רמה אחר רמה (החל מהשורש) משמאל לימין (באמצעות הכנסת צמתי העץ לתור). המתודה תיעזר לשם כך במבנה נתונים תור של STL (הממשק נמצא בנספח ב של מבחן זה).

**מספר קורס: 150018**

**תשובה לשאלה 11, סעיף ב:**

כותרת ומימוש המתודה connect :

## שנה"ל תשפ"ב, סמסטר ב', מועד ב'

### שאלון בחינה בקורס: סדנא ב ++C

מספר קורס: 150018

#### שאלה מספר 12:

בשאלה זה נרצה לטפל בחשבונות בנק מסוגים שונים: חשבון חיסכון וחשבון אשראי. בין השאר – נרצה לבצע פעולת עדכון חודשית של היתרה (לפי אחוזי הריבית) עבור כל החשבונות בבנק. הערה בעניין הריבית: ריבית מציינת את האחוז שיש להוסיף לסכום. לדוגמא – עבור 5% ריבית יש לחשב  $sum * 1.05$ , כדי להוסיף את הריבית לסכום הנמצא ב `sum`. [כמובן שלבנק יש היתר עיסקא הלכתי 😊].

לשם כך, כתוב את המחלקות המתאימות לפי ההגדרות להלן:

- i. מחלקה בשם `Account` ובה התכונות הבאות:
  - `number` – מספר החשבון
  - `interest` – אחוז הריבית החודשית (כמספר עשרוני)
  - `balance` – היתרה בחשבוןערכי ברירת המחדל לכל התכונות המספריות הוא 0. כמו כן – למחלקה זו יש לכתוב:
  - בנאי המקבל כפרמטרים מספר חשבון ויתרה, ויוצר עצם שמאותחל בערכים אלו. (הריבית תקבע בהמשך לפי סוג החשבון).
  - מתודה מופשטת (`abstract`, וירטואלית טהורה) פרטית בשם `mayWithdraw` לבדיקה האם מותר לבעל החשבון למשוך כסף מהחשבון. המתודה מקבלת את הסכום שאותו בעל החשבון מבקש למשוך, ומחזירה `true` במידה והמשיכה מותרת, `false` אם לא.
  - מתודה בשם `withdrawal` למשיכה מהחשבון. המתודה מקבלת את הסכום שאותו בעל החשבון מבקש למשוך. המתודה מזמנת את המתודה הבוליאנית `mayWithdraw`, על מנת לבדוק אם ללקוח מותר למשוך את הסכום שביקש. במידה ומותר לו – המתודה תעדכן את היתרה בהתאם לסכום שנמשך (הפחתה של הסכום שבעל החשבון ביקש למשוך מהיתרה). במידה ולא ניתן למשוך כסף מחשבון זה – המתודה לא תבצע דבר.
  - מתודה בשם `update` לעדכון חודשי של היתרה. (המתודה תופעל מהתוכנית הראשית לפי בקשת המתכנת). במידה והיתרה בחשבון היא חיובית – יחושב סכום הריבית על היתרה – והוא יתווסף ליתרה הקיימת. במידה והיתרה שלילית המתודה לא תבצע דבר.
- ii. מחלקת `Savings` עבור חשבון חסכון היורשת ממחלקת `Account`. מחלקה זו תכיל:
  - בנאי המקבל ערכים של מספר החשבון והיתרה. הריבית תאותחל ל-5%.
  - מימוש של המתודה `mayWithdraw`. במידה והלקוח מבקש למשוך סכום הקטן או שווה ליתרה בחשבון – המתודה תחזיר `true`. אחרת `false`.
- iii. מחלקת `Credit` עבור חשבון אשראי היורשת ממחלקת `Account`. מחלקה זו תכיל בנוסף לתכונות הקיימות (מספר חשבון, יתרה וריבית):
  - תכונה מספרית בשם `overdraft` המייצג את גובה האשראי שיש ללקוח (כלומר: הסכום שהלקוח יכול למשוך, מעבר למה שיש לו בחשבון, כך שתהיה יתרה שלילית) כמו כן, מחלקה זו תכיל:
    - בנאי המקבל ערכים של מספר חשבון, היתרה וגובה האשראי. הריבית תאותחל ל-1%.
    - מימוש של המתודה `mayWithdraw`. במידה והלקוח מבקש למשוך סכום הקטן או שווה לגובה היתרה + האשראי בחשבון – המתודה תחזיר `true`. אחרת `false`.
      - לדוגמא: אם ללקוח יש `balance` של 100 ו `overdraft` של 200:
        - הוא יכול למשוך 50 ש"ח (כי היתרה שלו הוא גבוהה מסכום המשיכה)
        - הוא יכול למשוך 150 ש"ח (כי היתרה שלו היא 100 ויש לו אשראי של 200 והסכום שלהם גבוה מסכום המשיכה)
        - הוא לא יכול למשוך 350 ₪ (כי אז הסכום בחשבון יעבור את גובה האשראי שלו).
    - מימוש מעודכן של המתודה `update` כך שאם היתרה היא חיובית, היא פועלת כאמור לעיל, אך אם היתרה היא שלילית יש להוריד מהיתרה קנס של 25 ₪.

לאחר מכן כתוב פונקציה גלובלית `updateAll`, המקבלת וקטור של `STL` בשם `accounts` המכיל את פרטי חשבונות הבנק (הוקטור יכול להכיל חשבונות משני הסוגים: `saving`, `credit`). על הפונקציה לבצע `update` עבור כל החשבונות בבנק. כמובן שהפעולה `update` תתבצע בהתאם לסוג העצם

הערות:

- אין צורך לבדוק תקינות קלט.
- אין צורך לכתוב בנאי ברירת מחדל. אין לתת ערכי ברירת מחדל לערכי הפרמטרים של הבנאי.
- הימנע משכפול קוד (למשל בבנאים). כלומר – אם יש כבר קוד המבצע את הפעולה, השתמש בו במקום לכתוב אותו שנית).
- את המתודות של המחלקות המכילות קוד, יש לממש מחוץ לממשק המחלקה.

שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב C++  
מספר קורס: 150018

תשובה לשאלה 12:

i. ממשק של מחלקת Account :

---

---

---

---

---

---

---

---

---

---

מימוש של מחלקת Account :

---

---

---

---

---

---

---

---

---

---

ii. ממשק של מחלקת Savings :

---

---

---

---

---

---

---

---

---

---

**מספר קורס: 150018**

מימוש של מחלקת Savings:

iii. ממשק של מחלקת Credit :

מימוש של מחלקת Credit:

**שנה"ל תשפ"ב, סמסטר ב', מועד ב'**

## שאלון בחינה בקורס : סדנא ב C++

**מספר קורס: 150018**

מימוש של פונקציה גלובלית updateAll:

מקום פנוי במידת הצורך להשלמת תשובות.

אנא ציינ במפורש לאיזו שאלה התשובה שלהלן \_\_\_\_\_ ולאיזה סעיף \_\_\_\_\_

שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב C++  
מספר קורס: 150018

נספח א' – ממשק ומימוש המחלקה SearchTree:

```
class Node
{
public:
    Node* left;
    Node* right;
    int value;

    Node(int val) : value(val), left(nullptr), right(nullptr) {}
    Node(int val, Node* l, Node* r) : value(val), left(l), right(r) {}
};
    //end of Node class
```

```
class SearchTree
{
private:
    Node* root;

public:
    SearchTree() { root = nullptr; }    // initialize tree
    virtual ~SearchTree();
    int isEmpty() const;
    void clear() { clear(root); root = nullptr; }

    void inOrder() { inOrder(root); }

    virtual void process(int val) { cout << val << " " ; }

    virtual void add(int val);

private:
    //private function for not give acces to user
    void clear(Node* current);
    void inOrder(Node* current);

    virtual void add(Node * current, int val);
};
```

```
SearchTree::~~SearchTree() // deallocate SearchTree
{
    if (root != nullptr)
        clear(root);
}
```

```
void SearchTree::clear(Node* current)
{
    if (current)
    {
        // Release memory associated with children
        if (current->left)
            clear(current->left);
        if (current->right)
            clear(current->right);
        delete current;
    }
}
```



שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב C++  
מספר קורס: 150018

```
int SearchTree::isEmpty() const
{
    return root == nullptr;
}

// inOrder processing of SearchTree rooted at current

void SearchTree::inOrder(Node* current)
{
    // visit left child, Node, right child
    if (current)
    {
        inOrder(current->left);
        process(current->value);
        inOrder(current->right);
    }
}

void SearchTree::add(int val)
{
    // add value to binary search tree
    if (!root)
    {
        root = new Node(val);
        return;
    }

    add(root, val);
}

void SearchTree::add(Node* current, int val)
{
    if (current->value < val)
    {
        if (!current->right)
        {
            current->right = new Node(val);
            return;
        }
        else add(current->right, val);
    }
    else
    {
        if (!current->left)
        {
            current->left = new Node(val);
            return;
        }
        else add(current->left, val);
    }
}
```

שנה"ל תשפ"ב, סמסטר ב', מועד ב'  
שאלון בחינה בקורס: סדנא ב C++  
מספר קורס: 150018

נספח ב' – ממשק המחלקה תור של STL:

## C++ Queues

<b>(constructor)</b>	Construct queue	public member function
<b>empty</b>	Test whether the container is empty	public member function
<b>size</b>	Return size	public member function
<b>front</b>	Access next element	public member function
<b>back</b>	Access last element	
<b>push</b>	Insert element	public member function
<b>emplace</b> C++11	Construct and insert element	public member function
<b>pop</b>	Remove next element	public member function
<b>swap</b> C++11	Swap content	public member function
<b>Non-member function overloads</b>		
<b>relational operators</b>	Relational operators for queue	function
<b>swap(stack)</b> C++11	Exchanges the contents of queues	public member function