



פקולטה: מדעי הטבע, מחלקה: מדעי המחשב

שם הקורס: תכנות מערכות ב'
קוד הקורס: 2-7020411 (כל הקבוצות)
מועד א_סמסטר: ב_שנה: ה'תשפ"ב
תאריך הבחינה: כא' סיוון תשפ"ב, 20/06/2022
משך הבחינה: שעה וחצי – 150 דקות

מרצה אחראי: ד"ר אראל סגל
מרצים: מור בסן, חיים שפיר
מתרגלים: טל זיכלינסקי, יבגני נייטרמן, נופר טאוב, לירון טאוב, אביטל אור

- חל איסור להשתמש בחומר עזר \ מחשבון \ מחשב.
- יש לענות על כל השאלות במחברת הבחינה בלבד, בכתב ברור וקריא!

נא קראו היטב את כלל ההנחיות והשאלות לפני כתיבת התשובות

- יינתן מענק של 2 נקודות לסטודנטים שיכתבו את הפתרון באופן ברור קריא וקל לבדיקה, בפרט: השאלות פתורות במחברת הבחינה לפי הסדר שבשאלות.
כל שאלה מתחילה בעמוד נפרד. הכתב ברור וקריא, ללא מחיקות קשקושים חיצים וטקסט מיותר.

יש לענות תשובות מלאות, להסביר כל תשובה בפירוט, ולכתוב תיעוד לקוד ושמות משמעותיים.
יש לענות תשובות ממוקדות - לא יינתנו נקודות על תשובות עם טקסט מיותר שאינו קשור לנושא.

אם אתם לא זוכרים, לא בטוחים או לא מבינים משהו בשאלה כלשהי - נסו לפתור את השאלה כמיטב יכולתכם, ציינו והסבירו מה הבנתם ולמה התכוונתם, והמשיכו לשאלה הבאה. אל "תיתקעו" בשאלה אחת.

בהצלחה!!!

ת"ז :

שאלה 0 [עד 60 נק']

נקודות על המטלות, הגשות וכו'

שאלה 1 [סה"כ 10 נק'] - בקיאות כללי (2 נק' לכל סעיף)

- א. ציינו את היתרונות/חסרונות בשימוש בתבניות (Templates) לעומת שימוש בפולימורפיזם.
- ב. מה ההבדל בין פונקציה וירטואלית לפונקציה וירטואלית טהורה? הסבירו ותנו דוגמא קצרה.
- ג. הסבירו כיצד פועל מנגנון ניהול פונקציות וירטואליות, פרטו את המשמעויות של המנגנון.
- ד. הסבירו את המושגים overloading , overriding , תנו דוגמא לכל מושג.
- ה. ציינו לפחות 3 הבדלים מהותיים בין שפת C++ לשפת JAVA לא קוד תחבירי (syntax—

שאלה 2 - 10 נקודות

נתונים שלושת קטעי הקוד הבאים:

קטע 1:

נתון קטע הקוד הבא אשר מתוארת בו פעולת מיון של מערך שלמים:

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

int Biggest(int* array, int size)
{
    int max=arr[0];
    for(int i=0; i<size; ++i)
        if( max<array[i])
            max=array[i];
    return max;
}

void Print(const int* array, int size)
{
    for(int i=0; i<size; ++i)
        cout<<array[i];
    cout<<endl;
}
```

קטע 2: נתונה במחלקה Product המתוארת באופן חלקי:

```
class Product
{
    private:

    int m_id; //קוד מוצר/
    string m_Name; /שם מוצר/

    public:
    //...

};
```

(המשך בעמוד הבא)

קטע 3: נתון גם הקטע הבא:

```
void main()
{
    const int size=3;
    double* ArrayDouble = new double[size];
    Product * ArrayProduct = new Product[size];
    Product *pa = new Product(12345,"Milk");
    Product *pb = new Product(45643,"Fruit");
    Product * pc = new Product(45878,"Vegetable");

    ArrayProduct[0]=*pa;
    ArrayProduct[1]=*pb;
    ArrayProduct[2]=*pc;

    for(int i=0; i<size;i++)
        ArrayDouble[i]= i*0.8;

    double max=Biggest(ArrayDouble,size);
    Print(ArrayDouble,size);

    Product Pmax=Biggest(ArrayProduct,size); // הגדול ביותר לקסיקוגרפית
    Print(ArrayProduct,size);

    delete[] ArrayDouble;
    delete[] ArrayProduct;
}
```

שנו את קטע הקוד הראשון ואת קטע הקוד השני כך שיוכל לעבוד גם עם ה main הנתון (מבלי לשנות את ה-main) בצורה נכונה. שימו לב! אין לכתוב מתודות נוספות בקטע הקוד הראשון אך מותר לערוך את המתודות הקיימות. בקטע הקוד השני ניתן להוסיף מתודות, אופרטורים ושדות.

שאלה 3 – 10 נק'

בהמשך למטלה מס' 3 במהלך הקורס על מחשבון פעולות למטריצות (ראו נספח בסוף)

א. מעוניינים להגדיר מתודה חדשה במחלקת המטריצה, אשר תקבל כפרמטר מטריצה אחרת ותבדוק האם הם חופפות. **התנאי להגדרתם כחופפות הינו:**
1. גודל המטריצות שווה 2. *סכום כלל* איברי המטריצות זהה בשתיים.
כתבו מימוש לאופרטור השוואה בהתאם

ב. ממשו את האופרטור A (העלאה בחזקה) למטריצה כלומר בהינתן מטריצה A בחזקה כלשהי, תחושב המשוואה הבאה

$$A^3 = A * A * A$$

ג. זיהוי מטריצת יחידה (ערכי האלכסון הינם 1 והשאר 0)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ לדוגמא:}$$

כתבו מתודה במחלקה אשר בודקת האם המטריצה הינה יחידה

שימו לב: יש לממש את הסעיפים לקוד בצורה קצרה ולאור המימוש שלכם במטלה

שאלה 4 – 10 נק'

א. (4 נק') כתבו תבנית פונקציה (template) בשם `count_if`, אשר מקבלת וקטור מכל סוג משתנה שהוא, וגם פונקטור (functor) כלשהו. הפונקציה תחזירה את מספר האיברים בווקטור אשר מקיימים את התנאי הנ"ל (functor), שימו לב: אין להשתמש בפונקציה מהספריה התקנית.

ב. (2 נק'). כתבו פונקטור בשם `greater_than_n` הפונקטור מאותחל ע"י פרמטר `n`. הוא מקבל ערך אחד, ובודק האם הערך הזה גדול מ-`n` להלן דוגמה לשימוש:

```
greater_than_n g(100);
```

```
cout << g(50); // false
```

```
cout << g(150); // true
```

ג. (4 נק'). כתבו תוכנית, אשר תדפיס את כלל המספרים בין 100 ל 1 (בסדר יורד) מבלי להשתמש בלולאות!

שאלה 5 - לינוקס (2 נק' לכל סעיף בשאלה)

א. כתבו סקריפט בלינוקס אשר יכתוב לקובץ את כלל הקבצים בספרייה אשר מתחילים באות b לתוך קובץ bfiles.txt

ב. כתבו סקריפט אשר ייצר מאה קבצים חדשים בספרייה עם סיומת מס' סידורי של הקובץ. לדוגמא file1.txt , file2.txt ... וכן הלאה עד לקובץ file100.txt

שאלה 6 (2 נק' לכל סעיף בשאלה)

נתונה התוכנית:

```
01 #include <iostream>
02 #include <string>
03 using namespace std;
04
05 class string {
06     const char* p;
07 public:
08     string(const char* p) {this->p = p;}
09 };
10
11 int main() { // a demo program
12     string s = "abc";
13 }
```

כשמריצים make מתקבלת ההודעה הבאה:

```
main.cpp:12:2: error: reference to 'string' is ambiguous
string s = "abc";
^
```

- א. מהו סוג השגיאה (קומפילציה \ זמן ריצה \ לוגית \ דליפת זכרון \ אחר) ?
- ב. הסבירו את הסיבה שגורמת לשגיאה? כיצד ניתן לתקן זאת?
- ג. במחלקה הנתונה יש פוינטר p אבל אין מפרק (destructor). האם זה עלול לגרום לשגיאה כלשהי? אם כן מה סוג השגיאה שעלולה להיווצר? אם לא, מדוע לא?

בנוס [2 נק]

יינתן מענק של 2 נקודות על כתיבה מסודרת לפי הפירוט הבא:

- השאלות פתורות במחברת הבחינה לפי הסדר שבשאלון;
- כל שאלה מתחילה בעמוד נפרד;
- הכתב ברור וקריא, בלי מחיקות\קשקושים\חיצים\וטקסט מיותר.

מחשבון מטריצות - שלב א

במהלך תואר במדעי המחשב, נדרשים הסטודנטים לעבור קורס באלגברה לינארית. ככל הנראה, כבר בשנה הבאה תצטרכו לבנות מטריצות משלכם שיעזרו לכם לכתוב תכניות העוסקות בלמידת מכונה, ניווט רובוטים אוטונומיים, בינה מלאכותית, ועוד.

הגדירו מחלקה בשם `Matrix` עם הפעולות הבאות (ראו בקובץ המצורף `Demo.cpp`):

- שישה אופרטורים חשבוניים: חיבור (+) הוספה (=+), פלוס אונרי (+), ושלושת האופרטורים המקבילים לחיסור (-). כאמור, חיבור/חיסור של שתי מטריצות מוגדר רק על מטריצות מאותו סדר גודל $m \times n$. ניסיון לחבר/לחסר שתי מטריצות שלא מקיימות תנאי זה יגרום לזריקת שגיאה.
- שישה אופרטורי השוואה: גדול, גדול-או-שווה, קטן, קטן-או-שווה, שווה, לא-שווה. לשם מטלה זו כללי השוואת מטריצות הם כדלקמן:

i. מטריצות יקראו שוות אם ורק אם כל האיברים בהן שווים.

ii. מטריצה A גדולה ממטריצה B אם ורק אם סכום איברי A גדול מסכום איברי B.

יתר האופרטורים העוסקים בהשוואה מוגדרים באופן טבעי הנגזר משני החוקים הנ"ל. כאמור, גם פעולות ההשוואה מוגדרות רק עבור מטריצות מאותו סדר גודל וניסיון להשוות מטריצות שלא מקיימות תנאי זה יגרום לזריקת שגיאה.

- הגדלה ב-1 (+) והקטנה ב-1 (-) לפני ואחרי המספר. פעולה זו תגדיל או תקטין ב-1 את כל אברי המטריצה.
- הכפלה בסקלר ממשי (`double`) באופן הטבעי והמוכר לכם מאלגברה לינארית.
- הכפלת מטריצה במטריצה, באופן הטבעי והמוכר לכם מאלגברה לינארית. ניסיון לבצע את פעולת כפל מטריצות על מטריצות בינהן הכפל לא מוגדר תגרום לזריקת שגיאה.
- אופרטור קלט ואופרטור פלט.