



פקולטה: מדעי הטבע. מחלקה: מדעי המחשב
שם הקורס: תכנות מערכות ב .
קוד הקורס: 7020411 כל הקבוצות
מועד: א סמסטר: ב שנה: ה'תשפ"ד
תאריך הבחינה: 21/7/24
משך הבחינה: שעתיים וחצי – 150 דקות

מרצה אחראי: ד"ר אראל סגל-הלוי
מרצים: מר מור בסן, ד"ר אלינה (בננסון) אופלינסקי
מתרגלים: מר ליאור ברייטמן, מר עמית רזינובסקי, מר בנימין סאלדמן, מר יהונתן עמוסי

חל איסור להשתמש בחומר עזר \ מחשבון \ מחשב.
יש לענות על כל השאלות במחברת הבחינה בלבד, בכתב ברור וקריא!
במבחן 5 שאלות, על 6 עמודים, יש לענות רק במחברת ולא על הטופס, יש לענות על כל השאלות

נא קראו היטב את כלל ההנחיות והשאלות לפני כתיבת התשובות

ייתן מענק של 2 נקודות על כתיבה מסודרת בהתאם לקיום כלל הסעיפים הבאים:

- כלל השאלות פתורות במחברת הבחינה לפי הסדר שבשאלון;
- 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!!!

שאלה 0 [50 נק']

[ציון על מטלות והצגות\נוכחות במעבדות]

כפי שנכתב בסילבוס : ציון זה יתווסף רק בתנאי של קבלת 30 נקודות או יותר במבחן

שאלה 1 [10 נק'] – בקיאות (2 נק' לסעיף)

בסעיפים אלו יש לענות בצורה ברורה ולתת דוגמאות קוד.

1. לגבי העמסת פונקציות (Function Overloading)

- מהי העמסת פונקציות?
- כתבו דוגמא לקוד בה מוצגים מס אפשרויות להעמסת פונקציה, יש להציג לפחות 3 אפשרויות להעמסת פונקציית כלשהי, בהתאם לעקרונות שנלמדו בקורס (אין צורך במימוש, חתימה מספיקה)
- האם שינוי הטיפוס המוחזר מהפונקציה מהווה גם העמסת פונקציה?

2. הסבירו בקצרה מהם המנגנונים השונים לניהול (הקצאה ושחרור) זיכרון בשפת ++C. מי אחראי לשחרור את הזיכרון בכל מנגנון? האם יש הבדל במיקום האחסון של האובייקטים בין שפת JAVA לשפת ++C? ציינו והסבירו את תשובתכם (דוגמות קוד)

3. האם יש הבדל או השפעה כלשהי למיקום כתיבת ה const, , הסבירו ופרטו לאור שורות הקוד הבאות, בהנחה ומעוניינים להשתמש ב p2, p1 בתוכנית

```
int num1 =5, num2 =10;
const int * p1 = &num1;
int * const p2 = &num2;
```

4. הסבירו את המושגים implicit / explicit casting. הוסיפו דוגמאות קוד מתאימות.

5. במהלך הקורס למדנו על כתיבת קוד תבניות (Templates) וראינו שיש מקרים בהם נדרש או נעדיף לממש קוד עבור מקרה פרטני (Template Specialization) בנוסף לכתיבת קוד תבנית כללית.

- הסבירו מתי נשתמש במימוש פרטני לתבניות (Template Specialization) ולא רק בתבנית הכללית.
- מהם היתרונות של שימוש במימוש פרטני לתבניות?

שאלה 2 [10 נק']

נתון מימוש של מחלקה

```
template<typename T, int ROWS, int COLS>
class Matrix {
private:
    T data[ROWS][COLS];

public:
    Matrix() {
        for (int i = 0; i < ROWS; ++i)
            for (int j = 0; j < COLS; ++j)
                data[i][j] = T();
    }
    Matrix(initializer_list<initializer_list<T>> init) {
        int i = 0;
        for (auto row : init) {
            int j = 0;
            for (auto elem : row) {
                if (i < ROWS && j < COLS) {
                    data[i][j] = elem;
                }
                ++j;
            }
            ++i;
        }
    }
    T get(int row, int col) const { return data[row][col]; }
    void set(int row, int col, T value) { data[row][col] = value; }
    T* operator[](int row) { return data[row]; }
    const T* operator[](int row) const { return data[row]; }
}
```

הוסיפו למחלקה שני איטרטורים (ראשון שסורק את המטריצה לפי שורות ושני לפי עמודות) (ואופרטורים הדרושים ++ pre/post-increment וכן אופרטור השוואה\אי-שוויון) ובנוסף פונקציות מתאימות ב Matrix כדי לתמוך בקוד הבא:

```
int main() {
    Matrix<int, 3, 4> matrix = {
        {9, 8, 7, 0},
        {6, 5, 4, 5},
        {3, 2, 1, 9}
    };
    cout << endl << " Matrix elements by rows: ";
    for (auto it = matrix.begin(); it != matrix.end(); ++it) {
        cout << *it << " ";
    }
    cout << endl << " Matrix elements by columns : ";
    for (auto it = matrix.colsBegin(); it != matrix.colsEnd(); ++it) {
        cout << *it << " ";
    }
    return 0;
}
```

שאלה 3 [10 נק']

נתונות המחלקות הבאות :

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

class A {
public:
    virtual void func1() { cout << "A::func1" << endl; }
    void func2() { cout << "A::func2" << endl; }
    virtual void func3() { cout << "A::func3" << endl; }
};

class B : public A {
public:
    void func1() { cout << "B::func1" << endl; }
    void func2() { cout << "B::func2" << endl; }
    virtual void func3() { cout << "B::func3" << endl; }
    virtual void func4() { cout << "B::func4" << endl; }
};

class C : public B {
public:
    void func1() { cout << "C::func1" << endl; }
    void func3() { cout << "C::func3" << endl; }
};

class D : public C {
public:
    void func1() { cout << "D::func1" << endl; }
    void func2() { cout << "D::func2" << endl; }
    void func3() { cout << "D::func3" << endl; }
    void func4() { cout << "D::func4" << endl; }
};

int main() {
    D d;
    A* pa = &d;
    B* pb = &d;
    C* pc = &d;
    D* pd = &d;
    pa->func1();
    pb->func2();
    pc->func3();
    pb->func4();
    return 0;
}
```

א. [3 נק'] ציינו עבור כל מחלקה את טבלת vtable אשר נשמרת עבורה יש לציין מהי הפונקציה הרלוונטית (מימוש) לכל למחלקה – לדוגמא מימוש של פונקציית foo במחלקת base - יש לרשום base:foo()

ב. [2 נק'] כתבו מה יודפס בעת הרצת הקוד.

ג. [5 נק'] ענו על כל השאלות הבאות:

מהי מטרת הגדרת פונקציה כ"פונקציה וירטואלית"?

מה תהיה התוצאה אם נשכח להגדיר פונקציה כפונקציה וירטואלית כאשר נדרשת הגדרה זו? האם כדאי להוסיף ולהשתמש במנגנון וירטואליות לכלל הפונקציות במחלקה (גם כשאין הכרח או צורך) עבור כלל הפונקציות במחלקה?

שאלה 4 [10 נק']

נתונה מחלקה הבאה:

```
class DMatrix {
private:
    int rows, cols;
    int** data;
public:
    DMatrix(int r, int c) : rows(r), cols(c), data(new int*[r]) {
        for (int i = 0; i < r; ++i) {
            data[i] = new int[c];
        }
    }
    int getRows() const { return rows; }

    int getCols() const { return cols; }
    int** getData() const { return data; }
};
```

א. [5 נק'] המחלקה עלולה לגרום לזליגת זיכרון. הסבירו בקצרה מדוע, הוסיפו קוד למחלקה כדי למנוע זאת?
ב. [5 נק'] מה יש להוסיף למחלקה על מנת שהיא תעבוד נכון לפי "כלל חוק השלושה" (Rule of Three)? הוסיפו את הקוד הנדרש בהתאם.

שאלה 5 – [10 נק']

א. [3 נק']

כתבו פונקציה כללית (תבנית) אשר מקבלת מיכל איברים **מכל סוג שהוא** (באמצעות איטרטורים) ומחזירה את האיבר המינימלי במיכל. דוגמה לשימוש בפונקציה:

```
void main() {  
    int numbers[] = {5, 2, 9, 1, 5, 6};  
    cout << minElement(numbers, numbers + 6) << endl; // should print 1  
}
```

ב. [3 נק']

מעוניינים ליצור מיכל חדש המורכב מאיברי מחלקה חדשה בשם `SomeClass` כתבו את הקוד שיש להוסיף במחלקה זו על מנת שנוכל למצוא את האיבר המינימלי במיכל באמצעות הפונקציה שבניתם בסעיף א'.

ג. [4 נק']

בקוד הבא נפלו מספר שגיאות, והקוד לא עובר הידור (קימפול). עליכם לכתוב כיצד ניתן לתקן את הקוד (וללא הוספת שורות חדשות) כך שיוכל לעבור הידור ולרוץ. שימו לב: חובה לציין במחברת את מספר שורה בה הנכם מעוניינים להכניס תיקון

```
1. #include <iostream>  
2. #include <string>  
3.  
4. class Car {  
5. private:  
6.     std::string model;  
7.     int year;  
8. public:  
9.     Car(std::string b, int y) : model(b), year(y) {}  
10.    virtual void printDetails() const {  
11.        std::cout << "Model: " << model << ", Year: " << year << std::endl;  
12.    }  
13. };  
14.  
15. class ElectricCar : Car {  
16. private:  
17.     int batteryCapacity;  
18. public:  
19.     ElectricCar( string b, int y, int bc) : model(b), year(y),  
                                                batteryCapacity(bc) {}  
20.    virtual void printDetails() override {  
21.        Car::printDetails();  
22.        cout << "Battery Capacity: " << batteryCapacity << " kWh" << endl;  
23.    }  
24. };  
25.  
26. int main() {  
27.     ElectricCar hyundai("Ioniq 6", 2024, 100);  
28.     hyundai.printDetails();  
29.     return 0;  
30. }
```