בית הספר למדעי המחשב
שם הקורס: תכנות מערכות **ב** .
קוד הקורס: 7020411 **כל הקבוצות**
מועד: **א** סמסטר: **ב** שנה: **ה'תשפ"ו**
תאריך הבחינה: **24.06.2026**
משך הבחינה: שעתיים וחצי – 150 דקות

מרצה אחראי : פרופ' אראל סגל-הלוי
מרצים : ד"ר אלינה (בננסון) אופלינסקי , גב' חרות סטרמן
מתרגלים: מר שחר צרפתי

חל איסור להשתמש בחומר עזר \ מחשבון \ מחשב.

במבחן 6 שאלות, יש לענות על כולן במחברת הבחינה בלבד, בכתב ברור וקריא. אין לענות על גבי הטופס.

אנא קראו היטב את כלל ההנחיות והשאלות **לפני** כתיבת התשובות

יינתן 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 , תשובה ארוכה ללא הקשר לא תזכה אתכם בניקוד. 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 7 [40 נק']

[ציון על מטלות]

כפי שנכתב בסילבוס : ציון זה יתווסף רק בתנאי של קבלת 36 נקודות או יותר במבחן

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

בסעיפים אלו יש לענות בצורה עניינית, קצרה וברורה ולתת דוגמאות קוד.

1. איך C++ תומכת בתכנות גנרי (generic programming) ? תנו דוגמא קצרה.
2. מהי פונקציה וירטואלית? וירטואלית טהורה? הראו דוגמא והסבירו.
3. מהי פונקציה inline? איך מגדירים כזו? מה היתרון בשימוש בה?
4. מה המשמעות של const בהקשר למשתנים ולמתודות? רשמו דוגמה לכל אחד מהם והסבירו.
5. הסבירו את ההבדלים בין הקצאה סטטית static allocation לבין דינאמית - dynamic allocation של זיכרון. מהם היתרונות והחסרונות של כל שיטה?

שאלה 2 [10 נק'] נתון מימוש של מחלקה

```
#include <iostream>

template  <typename T>  class LinkList {
private:
    struct Node {
        T data;
        Node* next;
        Node(const T& value) : data(value), next(nullptr) {}
    };
    Node* head;
    size_t size;
public:
    LinkList() : head(nullptr), size(0) {}
    ~LinkList() {
        clear();
    }
    void push_back(const T& value) {
        Node* newNode = new Node(value);
        if (head == nullptr) {
            head = newNode;
        }
        else {
            Node* current = head;
            while (current->next != nullptr) {
                current = current->next;
            }
            current->next = newNode;
        }
        size++;
    }

    void pop_front() {
        if (head == nullptr)
            return;
        Node* temp = head;
        head = head->next;
        delete temp;
        size--;
    }
    void clear() {
        while (head != nullptr) {
            pop_front();
        }
    }
    size_t getSize() const { return size; }
    bool isEmpty() const { return size == 0; }
};
```

הוסיפו למחלקה איטרטור . להלן תוכנית ראשית שיש לתמוך בה. (עמוד הבא)

```
int main() {
    LinkList<int> list;
    list.push_back(10);
    list.push_back(20);
    list.push_back(30);
    list.push_back(40);
    std::cout << "List contents: ";

    for (auto it = begin(list) ; it != end(list); ++it) {
        std::cout << *it << " ";
    }
    std::cout << "\n"; // List contents: 10 20 30 40
}
```

שאלה 3 [10 נק']

קראו את הקוד הבא וענו על 2 סעיפים הבאים:

1 (3 נק)

הוסיפו למחלקת Point פונקציה בשם **DistFromOrigin** שמחזירה את **המרחק** שלה מהראשית (0,0).

להזכירכם מרחק בין נקודות (x_1, y_1) ו- (x_2, y_2) מוגדר על ידי נוסחה הבאה:

$$dist = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

2 (7 נק)

הקוד לפניכם משתמש בפונקציה **sort** של הספרייה התקנית (STL) כדי למיין vector של נקודות. ממשו functor בשם **DistanceFromOriginCmp** כך שהמספרים יהיו ממוינים לפי **המרחק** שלהם מהראשית (0, 0) **בסדר עולה**, במקרה של **שוויון** מרחק – הנקודה עם ערך **ה־ x** הקטן יותר תופיע קודם. השתמשו בפונקציה אותה הגדרתם בסעיף הקודם.

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <cmath>

struct Point {
    int x;
    int y;
};

int main() {
    std::vector<Point> points = { {3,4}, {1,1}, {0,5}, {2,2}, {4,3} };
    std::sort(points.begin(), points.end(), DistanceFromOriginCmp());
    for (const auto& p : points) {
        std::cout << "(" << p.x << "," << p.y << ") ";
        // Example output: (1,1) (2,2) (3,4) (4,3) (0,5)
    }
    return 0;
}
```

שאלה 4 [10 נק']

לפניכם קוד מחלקה ותוכנית ראשית.
זהו את **כל** הפעולות החסרות כדי שתוכנית ראשית תתקמפל (אופרטורים חסרים) וממשו אותן.

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

class Fraction {
private:
    int num;
    int denom;
    void normalize() {
        if (denom == 0) denom = 1;
        if (denom < 0) {
            denom = -denom;
            num = -num;
        }
    }
public:
    Fraction(int num = 0, int den = 1) : num(num), denom(den) {
        normalize();
    }
};

int main() {
    Fraction f1(3, 4);   // 3/4
    Fraction f2(5, 6);   // 5/6

    bool isGreater = (f1 > f2);
    cout << "Is f1 > f2? " << (isGreater ? "Yes" : "No") << endl;

    Fraction sum = f1 + f2;
    cout << "f1 + f2 = " << sum << endl; // 19/12

    Fraction mul = 2 * f1;
    cout << "2 * f1 = " << mul << endl; // 6/4

    if (f1)     cout << "f1 is not zero!" << endl;

    return 0;
}
```

שאלה 5 – [10 נק']

נתונות המחלקות הבאות Image ו- Frame

- השלימו את ההגדרה של מחלקה Frame כדי שתנהל את הזיכרון בצורה תקינה.
- הסבירו את כל השינויים שביצעתם במחלקה – מדוע הם דרושים, ומה יקרה אם לא תוסיפו אותם?

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

// Class representing an Image
class Image {
private:
    string filename; /* probably some more code */
public:
    Image(string name) : filename(name) { /* */ }
    ~Image() { /* */ }
    string getFileName() const { return filename; }
};

// Class representing a Frame that holds an Image
class Frame {
private:
    Image* image;

public:
    Frame(string file) {
        image = new Image(file);
        cout << "Frame created with image: " << image->getFileName() <<
endl;
    }

    void printImage() const {
        cout << "Frame contains image: " << image->getFileName() << endl;
    }
};
```

שאלה 6 – [10 נק'] לפניכם קטע קוד, קראו בעיון וכתבו מה יודפס לאחר הרצת הקוד (יש להקפיד על הסימונים) ניתן להוסיף הסבר קצר.

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

class Animal {
public:
    Animal() {
        cout << "Animal ctor" << endl;
    }
    virtual void speak() {
        cout << "Animal speaks" << endl;
    }
    virtual ~Animal() {
        cout << "Animal dtor" << endl;
    }
};

class Cat : public Animal {
public:
    Cat() {
        cout << "Cat ctor" << endl;
    }

    void speak() override {
        cout << "Meow!" << endl;
    }

    ~Cat() {
        cout << "Cat dtor" << endl;
    }
};

int main() {
    cout << "/* Case 1 */" << endl;
    Animal* p = new Cat();
    p->speak();
    delete p;

    cout << "/* Case 2  */" << endl;
    {
        Cat kitty;
    }

    cout << "/* Case 3  */" << endl;
    {
        Cat c;
        c.speak();
        Animal a = c;
        a.speak();
    }

    return 0;
}
```