



פקולטה: מדעי הטבע. מחלקה: מדעי המחשב  
שם הקורס: תכנות מערכות ב .  
קוד הקורס: 7020411 כל הקבוצות  
מועד: א סמסטר: ב שנה: ה'תשפ"ג  
תאריך הבחינה: טו תמוז תשפ"ג, 2/7/23  
משך הבחינה: שעתיים וחצי – 150 דקות

מרצה אחראי : ד"ר אראל סגל-הלוי  
מרצים : מור בסן, אלינה (בננסון) אופלינסקי  
מתרגלים: חרות סטרמן, יבגני נייטרמן, יהונתן עמוסי

- 
- 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!!!**

## שאלה 0 [40 נק']

[ציון על מטלות + מענקים]

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

בסעיפים אלו יש לענות בצורה ברורה ולתת דוגמאות קוד.

1. ציינו 3 הבדלים (לא תחביר\syntax) שישנם בין pointers (מצביע) ל reference פוינטר יכול להיות null ולא תחל בכל שלב \ רפרנס חייב להיות מאותחל מהתחלה לאובייקט.

תקין – `int* p`

שגיאה – `int& r`

ניתן לשינוי הכתובת \ לא ניתן לשינוי

הפוינטר מצביע לעצם אחר. – `int* p; p = &b;`

הרפרנס מצביע לאותו מקום; העצם משתנה. - `int& r = b; r = c;`

פעולות אריתמטיות \ ללא

הפוינטר מצביע למקום אחר. – `int* p; p++;`

הרפרנס מצביע לאותו מקום; העצם משתנה. - `int& r = b; r++;`

מצביע = כתובת \ רפרנס = שם נוסף או כינוי לאובייקט והדגש על הערך האובייקט

הסבירו את המושגים LVALUE , RVALUE ותנו דוגמאות עבור כל אחד.

LVALUE - בעל מקום בזכרון, אפשר למקם בצד שמאל בהשמה.

`int a; a=2`

RVALUE - ביטוי \ חישוב או ללא מיקום בזכרון (הפכי ל LVALUE)

`2*a;`

כיצד ניתן להגדיר מחלקה אבסטרקטית (כמו JAVA) בשפת C++? תנו דוגמא קצרה.

מחלקה עם פונקציה וירטואלית טהורה (לפחות אחת)

```
class my_abs {
    virtual void draw () = 0;
    ...
}
```

}

הסבירו מה העקרון שעומד מאחורי SMART POINTER ? מה היתרון בשימוש בהם לעומת POINTER (מצביע) רגיל?

למדנו על מס' סוגים שיש ב-STL, מדובר בדגם עיצוב מחלקה שעוטפת מצביע וכוללת מפרק (שם מתבצעת מחיקה ההקצאה) אופרטורים תומכים כגון \* -> היתרון א. המתכנת לא נדרש יותר לדאוג לשחרור הקצאות באובייקט ב. עוזר למנוע דליפת זכרון במקרים שלא מגיעים ל delete, כגון כשיש חריגה. האם ניתן לממש יותר ממפרק Destructor אחד במחלקה ? הסבירו! לא. ה Destructor מופעל אוטומטית וללא פרמטרים. אין העמסה לגביו

## שאלה 2 [10 נק']

נתונה מחלקה גנרית של עץ חיפוש בינרי

```
template <typename T>
struct TreeNode {
    T value;
    TreeNode<T>* left;
    TreeNode<T>* right;
    explicit TreeNode(const T& val) :
        value(val), left(nullptr), right(nullptr) {}
};

template <typename T>
class BinarySearchTree {
private:
    TreeNode<T>* root;
public:
    BinarySearchTree() : root(nullptr) {}

    void insert(const T& value) {
        root = insertNode(root, value);
    }

    TreeNode<T>* insertNode(TreeNode<T>* node, const T& value) {
        if (node == nullptr) {
            return new TreeNode<T>(value);
        }
        if (value < node->value) {
            node->left = insertNode(node->left, value);
        }
    }
};
```

```

        else {
            node->right = insertNode(node->right, value);
        }
        return node;
    }
}

```

כזכור ניתן לסרוק את העץ בשיטות שונות, לדוגמא, in-order וכו' הוסיפו מחלקה בשם BSTLevelIterator שמאפשרת לסרוק את העץ לפי רמות משמאל לימין על המחלקה לממש פעולות של prefix ו postfix ++ וכן אופרטור גישה (dereference). יש להוסיף את הפעולות begin, end למחלקה הנתונה, כדי שיחזירו את האיטרטור המתאים.

רמז: ניתן להשתמש במיכל מתוך ספריה סטנדרטית std::queue או std::deque

מצורפת תוכנית ראשית המדגימה את פעולת האיטרטור – יש לממש את כל השיטות הדרושות כך שהתוכנית תעבוד כנדרש:

```

int main() {
    BinarySearchTree<int> bst;
    bst.insert(8);
    bst.insert(2);
    bst.insert(1);
    bst.insert(10);
    bst.insert(3);

    std::cout << "Elements by level:" << std::endl;
    for (auto it = bst.begin(); it != bst.end(); ++it) {
        std::cout << *it << " ";
    }
    //prints 8 2 10 1 3

    std::cout << std::endl;

    return 0;
}

```

### מימוש האיטרטור:

```

class BSTLevelIterator {
private:
    std::queue<TreeNode<T>*> nodes;
public:
    BSTLevelIterator(TreeNode<T>* root) {
        if (root != nullptr) {
            nodes.push(root);
        }
    }
    BSTLevelIterator& operator++() {
        TreeNode<T>* currentNode = nodes.front();
        nodes.pop();
    }
}

```

```

        if (currentNode->left != nullptr) {
            nodes.push(currentNode->left);
        }
        if (currentNode->right != nullptr) {
            nodes.push(currentNode->right);
        }
        return *this;
    }
    BSTLevelIterator& operator++(int) {
        // TODO
    }
    T operator*() const {
        return nodes.front()->value;
    }
    bool operator!=(const BSTLevelIterator& other) const{
        if (nodes.empty() && other.nodes.empty()) return false;
        if (nodes.empty() || other.nodes.empty()) return true;
        return nodes.front() != other.nodes.front();
    }
};

```

### מימוש הפונקציות ליצירת איטרטורים.

```

BSTLevelIterator begin() {
    return BSTLevelIterator(root);
}

BSTLevelIterator end() {
    return BSTLevelIterator(nullptr);
}

```

הערה: לא התייחסנו לשאלה אם המחלקה צריכה להיות פנימית או חיצונית (קיבלנו את שתי האפשרויות).

## שאלה 3 [10 נק']

נתונות המחלקות הבאות :

```
class A
{ public:
  virtual void f1() {cout<<"af1";}
  virtual void f2() {cout<<"af2";}
  virtual void f3() {cout<<"af3";}
  int _a;
};
```

```
class B : public A
{ public:
  virtual void f1() {cout<<"bf1";}
  void f2() {cout<<"bf2";}
  void f4() {cout<<"bf4";}
  int _b;
};
```

```
class C : public B
{ public:
  virtual void f1() {cout<<"cf1";}
  virtual void f2() {cout<<"cf2";}
  void f3() const {cout<<"cf3";}
  int _c;
};
```

א . [4 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()

טבלה מחלקה A

A:f1()  
A:f2()  
A:f3()

מחלקה B

B:f1()  
B:f2()  
A:f3()

מחלקה C

C:f1()  
C:f2()  
A:f3()

ב. [4 נק'] נתון הקוד הבא – כתבו מה יודפס בעת הרצת הקוד הבא :

```
Int main()
{
A* a1 = new A();
A* a2 = new B();
A* a3 = new C();
a2->f2();
a2->f3();
a3->f2();
}
```

bf2

af3

cf2

ג. [2 נק'] ציינו חסרונות (תקורות) בעת שימוש בפונקציות וירטואליות. פרטו את גודל התקורות.

תקורות: נוסף מצביע – כ-8 בתים לכל עצם. נוספת גם טבלת ניהול פונקציות וירטואליות – כ-8 בתים לכל פונקציה.

תקורה בזמן ריצה : נדרש לעבור דרך מצביע וטבלה – שני מעברים דרך מצביעים, לכל קריאה.

---

## שאלה 4 [10 נק'] -

### חלק א' - לינוקס \ bash

כתבו פקודות בלינוקס המבצעות את המשימות הבאות (כל משימה בפקודה אחת בלבד):

א [2 נק']. כתבו פקודה אשר מדפיסה את רשימת הקבצים בתיקיה הנוכחית לתוך קובץ בשם files.txt. `ls>files.txt`

ב [1 נק']. כיצד ניתן להריץ את הקובץ run.exe כך שרק הודעות השגיאה ישמרו לתוך קובץ log.txt

`./run.exe 2> log.txt`

(הראנו ש cerr ערוץ שגיאות)

ג [2 נק']. כתבו סקריפט אשר קולט מספר כלשהו מהמשתמש, ולאחר מכן מדפיס את כל המספרים הזוגיים מהמספר עד 0 בסדר יורד.

לדוגמא בעת קלט מספר 8 יודפס התוצאה 6,4,2,0

`#!/bin/bash`

```

echo "please insert number"
read num
while [ [$num -ge 0]]; do
    if [[ $num % 2 -eq 0 ]]; then
        echo "$num"
    fi
    ((num--))
done

```

## חלק ב' [5 נק']

```

// factorial.cpp
#include <iostream>

template <int N> struct Factorial {
    enum { value = N * Factorial<N-1>::value };
};
template <> struct Factorial<1> {
    enum { value = 1 };
};

// example use
int main()
{
    const int f = Factorial<15>::value;
    std::cout << f ;
    return 0;
}

```

הקוד, כצפוי, כותב למסך את הערך של 15 עצרת. הסבירו איך זה עובד:

- באיזה שלב בדיוק מתבצע החישוב? - בשלב הקומפילציה.
- איך נקראת טכניקה / שיטה זו לחישוב? – `template metaprogramming`.
- מה היתרון של שימוש בשיטה זו על-פני פונקציה רקורסיבית רגילה? – כיוון שכל החישוב נעשה בזמן הקומפילציה, זמן הריצה יהיה מהיר יותר.



## שאלה 5 [10 נק']

### חלק א' - 5 נק'

נתון הקוד הבא

```
2  #include <iostream>
3  using namespace std;
4
5  int main() {
6      int x = 5;
7      const int cx = 5;
8      const int & r1 = 100;
9      int v = 200;
10     int &r2 = v;
11     int & r3 = 200;
12     const int& r4 = (x+1);
13     int& r5 = cx;
14     cout << r2;
15     return 0;
16 }
```

א. האם הקוד תקין? מה יופיע בעת הרצת הקוד? במידה ולא תקין ציינו מדוע והיכן

לא. יש שגיאות בשורות 11, 13 :  
11 – מצביע על מספר (rvalue)  
13 - רפרנס רגיל שמצביע על קבוע

תשובות שציינו רק שגיאה אחת מתוך השתיים זכו לניקוד מלא בסעיף זה.

### חלק ב' - 5 נק'

נתון הקוד הבא

```
1  #include <iostream>
2  using namespace std;
3  template <typename T>
4  T max(T x, T y)
5  {
6      return (x > y) ? x : y;
7  }
8  int main()
9  {
10     cout << boolalpha;
11     cout << max(5, 8) << std::endl;
12     cout << max(8.0, 6.0) << std::endl;
13     cout << max(6, 8.0) << std::endl;
14     return 0;
15 }
```

## ב. האם הקוד תקין? מה יופיע בעת הרצת הקוד? במידה ולא תקין ציינו מדוע והיכן

לא. יש שתי שגיאות:

\* שורה 4: הגדרה חוזרת של max, שכבר מוגדר בספריה התקנית (נכנס למרחב השם הראשי בגלל using namespace std);

\* שורה 13: שימוש בתבנית max עם פרמטרים מסוגים שונים – שלם לעומת ממשי – לא מתאים לתבנית. תתקבל הודעת שגיאה כגון no candidate function ועוד הערות note וכו'.

תשובות שציינו רק שגיאה אחת מתוך השתיים זכו לניקוד מלא בסעיף זה.

---

## שאלה 6 – [10 נק']

א. [4 נק'] כתבו פונקציה כללית גנרית (תבנית template)

אשר מקבלת מערך איברים מסוג כלשהו, ומחזירה את סכום כלל איברי המערך

דוגמה לשימוש בפונקציה:

```
void main() {  
    int numbers[]={5 ,4 ,3 ,2 ,1};  
    cout << sum(numbers) << endl; // should print 15  
}
```

פתרון: כדי לדעת את גודל המערך, יש להשתמש בפרמטר מספרי לתבנית, כמו שנלמד בהרצאות:

```
template <typename T, int N>  
T sum(const T (&numbers)[N]) {  
    T sumOfArray = 0;  
    for (auto& number : numbers){  
        sumOfArray += number;  
    }  
    return sumOfArray;  
}
```

סטודנטים שלא עשו כך, אלא העבירו פרמטר מספרי לפונקציה, או שינו את סוג הפרמטר לזוטור, קיבלו 2 נקודות.

ב. [3 נק'] מעוניינים ליצור מערך חדש המורכב מאיברי מחלקה חדשה בשם myClass.

כתבו את קטעי-הקוד שיש להוסיף במחלקה זו, על-מנת שנוכל לחשב את סכום

המערך באמצעות הפונקציה שבניתם בסעיף א'.

נדרשת העמסת אופרטור + במחלקת myClass. צריך גם בנאי שיכול לקבל 0, כגון

בנאי שמקבל int.

```
class MyClass {
```

public:

```
MyClass(int x) { ... }
```

```
MyClass operator+(MyClass other) { return MyClass(...) }
```

```
}
```

**[3 נק']** האם יש הבדל בין auto ל decltype ? הסבירו והדגימו את תשובתכם בעזרת קוד.

יש הבדל.

auto מוגדר ע"י השמה בצד שמאל, למשל:

```
int a; double b; auto c = a+b;
```

הפקודה האחרונה שקולה ל: `double c = a+b;`

decltype מוגדר בעזרת סוג משתנה אחר שקיים (יתכן שלא מאותחל), למשל:

```
int a; double b; decltype(a+b) c;
```

הפקודה האחרונה שקולה ל: `double c;`

## שאלה 7 בונוס [2 נק']

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

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

