



פקולטה: מדעי הטבע. מחלקה: מדעי המחשב
שם הקורס: תכנות מערכות ב .
קוד הקורס: 7020411 כל הקבוצות
מועד: ב סמסטר: ב שנה: ה'תשפ"ג
תאריך הבחינה: יד אב תשפ"ג, 1.8.23
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)

בכלל סעיפים אלו יש לענות בצורה ברורה ולתת דוגמאות קוד.

1. הסבירו מהו מרחב-שם (namespace), למה נועד ומה מטרת השימוש בו ?

מרחב-שם מאפשר להגדיר משתנים ופונקציות כך שלא "יתנגשו" עם משתנים ופונקציות עם שם זהה במרחב-שם אחר. לדוגמה:

```
namespace myspace {  
    int a = 5;  
}  
int a = 6;  
cout << myspace::a; // prints 5  
cout << a; // prints 6
```

2. ציינו מה ההבדלים בין struct לבין class בשפת C++

ב-struct ברירת המחדל היא שהמשתנים והשיטות הם ציבוריים, וב-class ברירת המחדל היא שהם פרטיים. לדוגמה:

```
struct mystruct {  
    int a; // public  
};  
class myclass {  
    int a; // private  
};
```

[גם בירושה, ברירת המחדל ב-struct היא ירושה ציבורית, וב-class ירושה פרטית.]

3. הסבירו את המושג ביטוי למדא - lambda expression, ומה היתרון לשימוש בו? תנו דוגמא קצרה.

ביטוי למדא הוא ביטוי שמאפשר להגדיר פונקציה ללא שם, ע"פ הארגומנטים והקוד בלבד. הביטוי מאפשר להעביר פונקציה כפרמטר לפונקציה אחרת. לדוגמה, נניח שרוצים לכתוב פונקציה בשם plot שמציירת פונקציה אחרת. אפשר להעביר לה את הפונקציה שרוצים לצייר כביטוי למדא:

```
plot( [](float x) {return 2*x+1; } )
```

הביטוי המודגש בתוך הסוגריים הוא ביטוי למדא.

השתמשנו גם בעת העברת פונקציות לבדיקה וכו' באלגוריתמים ב STL, לדוגמה:

```
cout << "all_of(v, <10): " << all_of(begin(v), end(v), [](int x){return x<10;}) << endl;
```

4. הסבירו והדגו את ההבדלים בין אופרטור השמה (operator=) לבין בנאי העתקה (copy

constructor) ומתי יופעל כל אחד מהם? תנו דוגמא.

בנאי מעתיק הוא קודם-כל בנאי, כלומר הוא **יוצר עצם חדש**. הוא מופעל כשיוצרים עצם חדש מתוך עצם קיים, לדוגמה:

```
List new_list(old_list);
```

```
List new_list = old_list;
```

אופרטור השמה **משנה עצם קיים**. הוא מופעל כשמעתיקים עצם לתוך עצם קיים אחר, לדוגמה:

```
List new_list;
```

```
new_list = old_list;
```

לכן, בפרט, אופרטור השמה צריך לשחרר את המשאבים שנתפסו ע"י העצם הקודם. לדוגמה, בנאי מעתיק עבור רשימה:

```
List(const List& other) {  
    // Allocate memory for this list;  
    // Copy items from other list to this list.  
}
```

אופרטור השמה עבור רשימה:

```
List& operator=(const List& other) {  
    // Delete memory for this list;  
    // Allocate new memory for this list;  
    // Copy items from other list to this list.  
}
```

5. ציינו והסבירו שני הבדלים בין שפת JAVA לשפת C++ בתחומי \ נושאי מחלקות וירושה (לא תחביר\SYNTAX).

לימדנו את השקף הבא:

	C++	Java
Keyword	:	"extends"
Access to base	public, protected or private	public
Poly-morphic	Only if requested	Always
Multiple parents	Yes (useful mainly for interfaces)	No (except interfaces)
Interface keyword	"class"	"interface"
Universal base	None	Object

שאלה 2 [10 נק'] – איטרטור

להלן מחלקה גנרית המממשת רשימה מקושרת חד כיוונית עליכם להוסיף למחלקה איטרטור דו – כיווני להזכירכם איטרטור דו - כיווני מאפשר לעבור גם קדימה וגם אחורה הבהרה חשובה: יש להוסיף את האיטרטור ללא שינוי במבנה ה Node

```
template<typename T>
class SinglyLinkedList {
private:
    struct Node {
        T data;
        Node* next;
        Node(const T& value) : data(value), next(nullptr) {}
    };
    Node* head;
    size_t size;
public:
    SinglyLinkedList() : head(nullptr), size(0) {}
    ~SinglyLinkedList() {
        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;
}
};

```

שימו לב כי יש להוסיף למחלקה של איטרטור וגם של רשימה מקושרת את כל הפונקציות הנדרשות כדי שMAIN הבא יעבוד כמצופה, ואין צורך להוסיף פונקציות מעבר לדרוש.

```

int main() {
    SinglyLinkedList<int> list;
    list.push_back(10);
    list.push_back(20);
    list.push_back(30);

    // Test forward traversal using iterator
    std::cout << "Forward traversal:" << std::endl;
    SinglyLinkedList<int>::Iterator it = list.begin();
    for ( ; it != list.end(); ++it) {
        std::cout << *it << " ";
    }
    std::cout << std::endl; //10 20 30
    while (it != list.begin() )
    {
        --it;
        std::cout << *it << " ";
    }
}

```

```

std::cout << std::endl; //30 20 10
return 0;
}

```

המחלקה איטרטור :

```

class Iterator {
    Node *start;
    Node *current;

public:
    Iterator(Node *start, Node *current) {
        this->start = start;
        this->current = current;
    }
    bool operator==(Iterator other) { return current == other.current; }
    bool operator!=(Iterator other) { return current != other.current; }
    T operator*() { return current->data; }
    Iterator &operator++() {
        current = current->next;
        return *this;
    }
    Iterator &operator--() {
        if (current == start) {
            current = NULL;
        } else {
            Node *previous = start;
            while (previous != NULL && previous->next != current) {
                previous = previous->next;
            }
            if (previous->next == current) {
                current = previous;
            } else {
                throw "could not find the previous node";
            }
        }
        return *this;
    }
};

```

(אפשרות יעילה יותר היא להשתמש במחסנית). פונקציות שיש להוסיף בתוך המחלקה LinkedList:

```

Iterator begin() { return Iterator(head, head); }
Iterator end() { return Iterator(head, NULL); }

```

מפתח ניקוד:

2 begin+end ,2 * ,1 != ,1 == ,3 -- ,1 ++

שאלה 3 [10 נק']

נתונות המחלקות הבאות :

```

class B
{
public:
    virtual void func1() { cout << "bf1"; }
    virtual void func2() { cout << "bf2"; }
    virtual void func2(int x) { cout << "f2"; }
    void func3() const { cout << "bf3"; }
    ~B() { func2(); }
}

```

```

};
class C : public B
{
public:
    virtual void func2() { cout << "cf2"; }
    virtual void func3() { cout << "cf3"; }
    void func5() { cout << "cf5 "; }
    ~C() { func2(); }
};
class D : public C
{
public:
    void func2() { cout << "df2"; }
    virtual void func5() { cout << "df5"; }
    virtual void func6() { cout << "df3"; }
};

```

א . [4 נק'] ציינו עבור כל מחלקה את טבלת vtable אשר נשמרת עבורה
יש לציין מהי הפונקציה הרלוונטית (מימוש) עבור כל למחלקה – לדוגמא מימוש של פונקציית
foo במחלקת base - יש לרשום base::foo()

```

class B:
    B::func1()
    B::func2()
    B::func2(int)

```

```

class C:
    B::func1()
    C::func2()
    B::func2(int)
    C::func3()

```

```

class D:
    B::func1()
    D::func2()
    B::func2(int)
    C::func3()
    D::func5()
    D::func6()

```

ב . [3 נק'] נתון הקוד הבא. האם הקוד תקין? אם כן – כתבו מה יודפס; אם לא – כתבו מה
השגיאה.

```
void main() {
    B* b1 = new B();
    B* b2 = new C();
    B* b3 = new D();
    B* b4 = new B();
    b1->func2();
    b2->func3();
    b3->func5();
}
```

התוכנית לא תתקמפל. יש שגיאת קומפילציה בשורה האחרונה: הפונקציה func5 לא מוגדרת במחלקה B.

error: no member named 'func5' in 'B'

[אם נמחק את השורה האחרונה, יודפס:

bf2

bf3

כי המשתנה הוא מסוג B, ובמחלקה B הפונקציה func3 אינה וירטואלית]

ג. [3 נק']

בקורס נלמדו המושגים Static resolution, Dynamic resolution (לפעמים גם קראנו לזה קישור\binding). הסבירו את המושגים ומה ההבדלים ביניהם. תנו דוגמא קצרה.

static resolution = ההחלטה איזו גרסה של הפונקציה להפעיל מתקבלת בזמן הקומפילציה, בהתאם לסוג המוגדר של המשתנה. לדוגמה, כשקוראים לפונקציה b2->func3 בתוכנית למעלה, מתבצעת static resolution: כיוון שסוג המשתנה הוא B, והפונקציה אינה וירטואלית במחלקה B, נקראת הגרסה של הפונקציה מהמחלקה B.

dynamic resolution = ההחלטה איזו גרסה של הפונקציה להפעיל מתקבלת בזמן הריצה, בהתאם לסוג האמיתי של העצם. לדוגמה, אילו הפונקציה func3 הייתה מוגדרת כווירטואלית במחלקה B, אז כשקוראים לפונקציה b2->func3 בתוכנית למעלה, מתבצעת dynamic resolution: כיוון שסוג העצם שנבנה הוא C, נקראת הגרסה של הפונקציה מהמחלקה C.

שאלה 4 [10 נק'] -

חלק א' - לינוקס \ bash

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

א [1 נק']. נתון קובץ בשם **data** אשר מכיל אוסף מספרים (כל מספר בשורה נפרדת). כתבו פקודה בלינוקס אשר תדפיס את כל השורות הכוללות "789".

```
grep 789 data
```

ב [1 נק']. כתבו פקודה אשר משנה את הרשאות הקובץ text.log כך שיהיה ללא הרשאות קריאה וכתיבה.

```
chmod a-rw text.log
```

ג [3 נק']. כתבו סקריפט אשר מבצע את הדברים הבאים
מבקש מהמשתמש לבחור שם קובץ וקולט אותו
יוצר קובץ חדש בשם שנתקבל מהמשתמש

קולט מספר חדש מהמשתמש. במידה והמספר גדול או שווה 5
יועתק המידע מקובץ data.log כפול כמספר הפעמים
אשר התקבל מהמשתמש לתוך הקובץ החדש שנוצר. אחרת יש לשמור ספרה 0 לקובץ.

לדוגמא במידה ומשתמש בחר בשם של קובץ **test.txt** ונתקבל המספר 5 מהמשתמש ובקובץ data.log קיים המידע "11111"

כעת הקובץ החדש בשם **test.txt** יכיל 5 פעמים את המחרוזת "11111"

```
#!/bin/bash
```

```
echo "Please enter a filename:"  
read filename
```

```
echo "Please enter a number:"  
read number
```

```
if (( number >= 5 )); then  
    touch $filename  
    for (( i=0; i<number; i++ )); do  
        cat data.log >> $filename  
    done  
else  
    # If the number is less than 5, write the digit 0 to the file  
    echo "0" > "$filename"  
fi
```

4. חלק ב [5 נק']

נתון הקוד

```
class SomeClass{  
  
    void f1(int i,int j) {  
        i / j;  
    }  
    void main() {  
        SomeClass ob;  
        int i, j;  
    }  
}
```

```

SomeClass* ptr = new SomeClass();
std::cin >> i >> j;
f1(i, j);
delete ptr;
}

```

א. תארו מצב שבו התוכנית תרוץ כשורה, ומצב אחר שבו תהיה זליגת זיכרון.

א. אם הקלט למשתנה j שונה מאפס, התוכנית תרוץ כשורה.

אם $j=0$, אז הפונקציה $f1$ תזרוק חריגה, ptr אינו משתחרר ויש זליגת זכרון.

ב. תארו פתרון פשוט למניעת זליגת הזיכרון בעזרת כלים של הספריה התקנית.

ב שימוש במצביע חכם ובדיקה ב $f1$:

```

void main() {
    SomeClass ob;
    int i, j;
    unique_ptr<SomeClass> ptr (new SomeClass());
    std::cin >> i >> j;
    f1(i, j);
}

```

שאלה 5 [10 נק']

חלק א' - 5 נק'

נתון הקוד הבא

```

1  #include <iostream>
2
3  template <class T>
4  class Test
5  {
6  public:
7      T temp;
8      static int sum;
9      Test(){++sum;}
10     void Print(){std::cout<<"Fun";}
11 };
12
13 template<class T>
14 int Test<T>::sum=0;
15
16 int main()
17 {
18     Test<int> a;
19     Test<float> b;
20     Test<int> c;
21     std::cout<<Test<int>::sum<<std::endl;
22     std::cout<<Test<float>::sum<<std::endl;
23     return 0;
24 }

```

א. האם הקוד תקין? במידה וכן מה יופיע בעת הרצת הקוד?
 במידה והקוד לא תקין, ציינו מדוע והיכן ומה נדרש בכדי לתקנו
 הקוד תקין: יודפס בהתחלה 2 ואחר-כך 1.

5. חלק ב' - 5 נק'

נתון הקוד הבא

```

int main() {
    int timeInSeconds = 1_h + 30_m + 12_s;
    std::cout << "1_h + 30_m + 12_s = " << timeInSeconds << std::endl;
}

```

א. [2 נק'] בעת הרצת הקוד נקבל שגיאה - מהי?
 השגיאה היא שהתחביר "קו-תחתי ה" דורש אופרטור גרשיים ("operator"), והאופרטור הזה לא הוגדר עבור מספרים שלמים.

ב. [3 נק'] כתבו קוד (מחוץ לתוכנית הראשית) כך שהקוד הנ"ל יוכל לרוץ תקין ולהדפיס את מספר השניות הנכון (במקרה זה 5412):

```

int operator""_h(unsigned long long x) { return x * 3600; }
int operator""_m(unsigned long long x) { return x * 60; }
int operator""_s(unsigned long long x) { return x; }

```

שאלה 6 – [10 נק']

חלק א' [5 נק'] כתבו מחלקה גנרית (template) אשר תכיל את הדברים הבאים :

- שני משתני-תבנית תבנית מסוגים שונים.
 - פונקציית הדפסה print אשר תדפיס את משתני המחלקה.
 - פונקציה printtypes אשר תדפיס את סוגי המשתנים.
 - פונקציה areequal הבודקת האם מדובר ב-2 משתנים מסוגים זהים או שונים.
- לאחר הגדרת המחלקה, הראו קוד דוגמא לשימוש במחלקה בעת קבלת משתנים מסוג זה ומסוג שונה.

```
template <typename A, typename B> class Test {  
    A a;  
    B b;  
  
public:  
    Test(A a, B b) {  
        this->a = a;  
        this->b = b;  
    }  
  
    void print() { cout << "a: " << a << " b: " << b << endl; }  
  
    void printtypes() { cout << "a: " << typeid(a).name() << " b: " <<  
        typeid(b).name() << endl; }  
  
    bool areequal() { return (typeid(a) == typeid(b)); }  
};  
  
int main() {  
    Test<int, string> test1(5, "abc");
```

```

test1.print();

test1.printtypes();

cout << test1.areequal() << endl;

Test<int, int> test2(7,5);

test2.print();

test2.printtypes();

cout << test2.areequal() << endl;

}

```

6. חלק ב' [5 נק']

1. [2 נק'] ציינו שני יתרונות משמעותיים בגללם נעדיף להשתמש בספרייה התקנית STL במקום לכתוב את האלגוריתמים בעצמנו;

1. קוד שנבדק רבות; 2. יעילות גבוהה (פותח ע"י מפתחי השפה)

2. [3 נק'] כתבו תוכנית קצרה ככל האפשר (המשתמשת ברכיבי ספריית STL) המגדירה וקטור עם מספרים שלמים לאו דווקא לפי סדר ואולי עם כפילויות. מדפיסה אותם בסדר עולה ורק פעם אחת. ניקוד מלא יינתן רק למי שיעשה את זה בצורה הקצרה ביותר וללא שימוש ב SORT. השלימו את התוכנית הבאה:

```

#include <iostream>
#include <vector>
#include <set>
#include <algorithm>

int main() {
    std::vector<int> sourceVec = { 3, 1, 2, 2, 4, 3, 5 };
    -----
    -----
    -----
    -----
    return 0;
}

```

פתרון:

```

#include <iostream>
#include <vector>
#include <set>
#include <algorithm>

int main() {

```

```
std::vector<int> sourceVec = { 3, 1, 2, 2, 4, 3, 5 };
std::set<int> destinationSet ( sourceVec.begin(), sourceVec.end());
std::copy(destinationSet.begin(), destinationSet.end(),
          std::ostream_iterator<int>(std::cout, " "));

return 0;
}
```

שאלה 7 בונוס [2 נק']

ייתן 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.