

אוניברסיטת אריאל בשומרון

פקולטה: מדעי הטבע

מחלקה: מדעי המחשב

שם הקורס: מבנה זיכרון ושפת C++

קוד הקורס: 2-7027810 כל הקבוצות - קבוצות 1,2,3,4

מבחן לדוגמה

משך הבחינה: 3 שעות

שם המרצים: אראל סגל-הלוי, חרות סטרמן

שאלה 1 [10 נקודות]

א [1 נק']. שגיאת זמן-ריצה.

ב [4 נק']. שורה 16, גישה לכתובת מחוץ לגבולות המערך.

ג [5 נק']. הוספת שיטה:

```
char& at(int index) {  
    if (index >= length*width) throw exception();  
    return pixels[index];  
}
```

ושינוי שורה 16 ל:

```
16 at(i*width+j) = 65;
```

שאלה 2 [10 נקודות]

א [1 נק']. שגיאת קומפילציה.

ב [4 נק']. בשורה 19 אנחנו מקבלים את האובייקט c כפרמטר קבוע. לכן בשורות 20,23 ייקראו הפונקציות הקבועות שלא מאפשרות שינוי של האובייקט.

ג [5 נק']. מחיקת ה-const בשורה 19:

```
19 istream& operator>> (istream& in, Complex& c) {
```

שאלה 3 [10 נקודות]

א. 120 בתים. יש בערימה מערך של עשרה אובייקטים. כל אובייקט בגודל שלושה int, כלומר כל אובייקט 12 בתים. כלומר, סך הכל 120 בתים.

ב. N הוא 120. כמו שראינו בסעיף א, זה מספר הבתים שהוקצו במהלך התכנית ולא שוחררו.

ג. 280 בתים. יש בערימה מערך של עשרה מצביעים לאובייקטים, כל המצביעים 80 בתים. כל אובייקט בגודל שלושה int + מצביע לטבלה וירטואלית. כלומר כל אובייקט 20 בתים. אז כל האובייקטים 200 בתים. יחד עם המצביעים, קיבלנו 280 בתים.

ד. 120 בתים. הוספנו פונקציות לא וירטואליות, ולכן לא נוסף שדה עבור מצביע לטבלה וירטואלית, ולכן גודל האובייקט לא משתנה.

ה. 280 בתים. הוספנו פונקציות ולא שדות, ולכן גודל האובייקט לא משתנה.

שאלה 4 [10 נקודות]

א [1 נק']. שגיאת זמן-ריצה.

ב [4 נק']. בשורות 17-21 אנו יוצרים אובייקט חדש, משנים את ערכי המערך שלו, שומרים את כתובת המערך באובייקט הנוכחי ומוחקים את האובייקט החדש, כולל המערך שלו. בשורה 32 כאשר מוחקים את האובייקט b2, מנסים שוב למחוק את אותו מערך שכבר מחקנו קודם.

ג [5 נק']. לקצר את שורות 18-20 לשורה אחת:

```
18 for (int i=0; i<size; ++i) this->doubles[i] = d;
```

שאלה 5 [10 נקודות]

א [1 נק']. שגיאת זמן-ריצה.

ב [4 נק']. בבנאי המעתיק (שורות 8-12) לא מאתחלים את השדה size, ונשאר בתוכו זבל מריצה קודמת. אחר כך במתודה fill (שורות 13-17) מנסים להשוות את i למה שכתוב בתוך size ואז התכנית עפה.

ג [5 נק']. שינוי שורה 8 ל:

```
08 Board(const Board& other): size(other.size) {
```

שאלה 6 [40 נקודות]

תוכן הקובץ AbstractTable.h:

```
#pragma once
```

```
class AbstractTable {
    static int _count;
protected:
    int size;
public:
    AbstractTable(int size): size(size) {
        _count++;
    }
    virtual ~AbstractTable() {
```

```

        _count--;
    }
    virtual void print() const = 0;
    static int count() { return _count; }
};

```

```
int AbstractTable::_count = 0;
```

תוכן הקובץ :IntegerTable.h

```
#pragma once
```

```
#include <iostream>
```

```
#include "AbstractTable.h"
```

```

class IntegerTable: public AbstractTable {
    int* values;
public:
    virtual void print() const {
        std::cout << "Integer table with values: ";
        for (int i=0; i<size; ++i) std::cout << values[i] << " ";
        std::cout << std::endl;
    }
    IntegerTable(int size): AbstractTable(size), values(new int[size]) {
    }
    ~IntegerTable() {
        delete[] values;
    }
};

```

תוכן הקובץ :StringTable.h

```
#pragma once
```

```
#include <iostream>
```

```
#include "AbstractTable.h"
```

```

class StringTable: public AbstractTable {
    std::string* values;
public:
    virtual void print() const {
        std::cout << "String table with values: ";
        for (int i=0; i<size; ++i) std::cout << values[i] << " ";
        std::cout << std::endl;
    }
}

```

```

StringTable(int size): AbstractTable(size), values(new std::string[size]) {
    for (int i=0; i<size; ++i) values[i] = ".";
}
~StringTable() {
    delete[] values;
}
};

```

שאלה 7 [10 נקודות]

```

#include <algorithm>

struct sequence1_is_longer_than_sequence2{};
struct sequence2_is_longer_than_sequence1{};

template <typename Iterator1, typename Iterator2>
void minmax(Iterator1 begin1, Iterator1 end1, Iterator2 begin2, Iterator2 end2) {
    while (true) {
        if (begin1==end1 && begin2==end2)
            break; // OK: both sequences have the same length.
        else if (begin1==end1 && begin2!=end2)
            throw sequence2_is_longer_than_sequence1{};
        else if (begin1!=end1 && begin2==end2)
            throw sequence1_is_longer_than_sequence2{};
        else {
            if (*begin1 > *begin2)
                std::swap(*begin1,*begin2);
            ++begin1;
            ++begin2;
        }
    }
}

```