פקולטה: מדעי הטבע

מחלקה: מדעי המחשב

שם הקורס: מבנה זכרון ושפת C++

קוד הקורס: 7027810

תאריך בחינה: **שאלות לדוגמא**

משך הבחינה: שעתיים

שם המרצים: ד"ר אופיר פלא, ד"ר מירי בן ניסן

חומר עזר: פתוח

שימוש במחשבון: לא

<u>הוראות כלליות (אותן ההוראות יהיו גם במבחן):</u>

- הניסוח הוא בלשון זכר מטעמי נוחות ומתייחס לכולם/ן!
- אין בחירה במבחן. יש לענות על כל השאלות.
- ניתן להסתמך על כל סעיף במבחן גם אם לא פתרתם אותו על מנת לפתור סעיף אחר במבחן.
- ניתן לרשום "לא יודע/ת" על סעיף ולזכות ב 20% מהנקודות המוקנות לסעיף הספציפי.
- במידה ולסעיף ניתנה תשובה ובנוסף נרשם לגבי הסעיף "לא יודע/ת" אזי הניקוד שיינתן לסעיף יהיה 0 מבלי שהתשובה תיקרא.
- אם לא רשום דבר בסעיף או שזה כלל לא נמצא ההנחה היא שנרשם "לא יודע/ת" עבור אותו סעיף.
- יורדו נקודות על פתרון נכון שאינו אופטימלי בהתייחס לנלמד בקורס.
- לכל אורך הבחינה הניחו כי:

```
sizeof (char) = 1
sizeof (int) = 4
sizeof (double) = 8
sizeof (void *) = 8
sizeof (size_t) = 8
```

שאלה 1

נתון קוד בקובץ main.cpp הבא:

```
#include "Computer.hpp"
#include "PC.hpp"
#include "Macintosh.hpp"
int main() {
 Computer* p[2];
 Macintosh imac("Joe", "blue"); // owner (all computers have one),
                                // color (only for macs)
 PC pc("Ben", 6); // owner, weight (only for PCs)
 p[0]= &imac;
 p[1]= &pc;
 p[0]->print(); // prints the line: Joe's blue Macintosh
 p[1]->print(); // prints the line: Ben's 6kg PC
 return 0;
}
```

כתבו קוד ב-3 קבצי ה hpp (כתבו כהערה את שם הקובץ מעל) כך שהקוד יהיה תקין ובסגנון טוב. אין חובה לכתוב את הכללות של קוד מהספרייה הסטנדרטית.

שאלה 2

עבור כל הצהרה כתוב במחברת אם היא נכונה או לא.

א. ברגע שמגדירים את האופרטור =, אזי האופרטור =! מוגדר אוטומטית

ב. לאופרטור שהגדרנו מחדש חייב להיות לפחות אופרנד אחד שהוא user defined

ג. מחלקה היא אבסטרקטית רק אם כל המתודות שלה הן pure virtual

שאלה 3

נתון קטע הקוד הבא:

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

struct A {
      void f()          {cout<<"(1)"<<endl;}
      virtual void g()  {cout<<"(2)"<<endl;}
      virtual void h()  {cout<<"(3)"<<endl;}
      ~A()              {cout<<"(4)"<<endl;}
};

struct B : public A {
      void f(int i)     {cout<<"(5)"<<endl;}
      virtual void g()  {cout<<"(6)"<<endl;}
      void k()          {cout<<"(7)"<<endl;}
      virtual ~B()      {cout<<"(8)"<<endl;}
};

struct C : public B {
      virtual void h()  {g(); cout<<"(9)"<<endl;}
      virtual ~C()      {cout<<"(10)"<<endl;}
};

int main() {
      //-----------------------
      // Basic pointers to 3 objects:
      A* ptr_A = new A;
      B* ptr_B = new B;
      C* ptr_C = new C;

      //-----------------------
      cout << "Part 1:" << endl;
      ptr_B->f(3);

      //-----------------------
      cout << "Part 2:" << endl;
      A* ptr_AtoB = ptr_B;
```

```
    ptr_AtoB->f();

    //-----------------------
    cout << "Part 3:" << endl;
    A* ptr_AtoC = ptr_C;
    ptr_AtoC->g();
    ptr_AtoC->h();

    //-----------------------
    cout << "Part 4:" << endl;
    B* ptr_BtoC = ptr_C;
    ptr_BtoC->k();

    //-----------------------
    cout << "Part 5:" << endl;
    // Note the name of the variables being deleted!!!
    delete ptr_A;
    delete ptr_AtoB;
    delete ptr_BtoC;

    return 0;
}
```

א. כתוב במחברת את הטבלאות הוירטואליות של כל אחת מן המחלקות הנ"ל. הקפד על הקידומת ::A או ::B או ::C. **בנוסף,** הקפידו על הסדר הנכון, כלומר, כך שהפונקציות הוירטואליות של הבן שהן גם וירטואליות אצל האבא יופיעו **באותו המקום בטבלה אצל הבן ואצל האבא, כולל ה destructors,** כך שאם מצביעים על בן ע"י מצביע לאבא הפונקציה נמצאת באותו המקום.

ב. רשום את הפלט של התוכנית הנ"ל, כולל חלקי ה Part.

שאלה 4

מתכנת כתב את הקוד הבא בקובץ q4.cpp:

```
#include <iostream>
class IntBuffer {
 int *_vec;
 size_t _size;
public:
 IntBuffer (size_t size) : _vec (new int[size]), _size(size) {};
 int& operator[] (size_t i)       {return _vec[i];}
 int& operator[] (size_t i) const {return _vec[i];}
};

int main () {
 const IntBuffer v(3);
 v[1]= 15;
 std::cout << v[1] << std::endl;
 return 0;
}
```

משתמש הריץ את הקמפול הפשוט הבא וניסה להריץ את התוכנית, מה הפלט שראה? הסבר.

```
$ g++ -Wall q4.cpp -o q4
$ ./q4
```

שאלה 5:

קראו את התוכנית הבאה. התוכנית מכפילה שתי סדרות של מספרים איבר, איבר ושמה את התוצאות בסדרה שלישית. הפלט של התוכנית הוא 0 10-. כתבו את הקובץ "mul_elements.hpp" כך שהתוכנית תעבוד היטב. כתבו קוד גנרי ככל האפשר. הניחו כי כל הסדרות הינן באותו האורך (כלומר, מותר שהתנהגות התוכנית תהיה לא מוגדרת אחרת).

```
#include "mul_elements.hpp"
#include <list>
#include <vector>
#include <iostream>

int main() {
  int arr[]= {10,2};
  const size_t arrLen= sizeof(arr)/sizeof(arr[0]);

  std::list<int> lst;
  lst.push_back(-1);
  lst.push_back(0);

  // Init a vector with two elements: [-42,-42];
  std::vector<int> resultVec(2, -42);

  mul_elements(arr, arr+arrLen, lst.cbegin(), resultVec.begin());

  for (const auto& val : resultVec) {
    std::cout << val << ' ';
  }

  return 0;
}
```

שאלה 6

מהם הטיפוסים השונים של T שנוצרו במהלך קימפול הקוד הבא:

```
template <typename T>
class A {
 T* _pT;
public:
 A (): _pT (new T ()){}
 ~A () {delete _pT;}
};
int main() {
 A < A<A<int>*> > a;
 A < A<int> > b;
 return 0;
}
```

<u>פתרון שאלה 1</u>

הערות: שימו לב לשימוש ב private, protected, public, const, override, virtual. בבדיקה תהיה חשיבות גם לסגנון!

```
// Computer.hpp
#include <iostream>
#include <string>
class Computer {
  virtual void printSpecific() const = 0;
  std::string _owner;
protected:
  Computer(std::string owner) : _owner(owner) {}
public:
  virtual ~Computer() {}
  void print() const {
      std::cout << _owner << "'s ";
      printSpecific();
  }
};

// Macintosh.hpp
class Macintosh : public Computer {
  std::string _color;
  virtual void printSpecific() const override {
      std::cout << _color << " Macintosh\n";
  }
public:
```

```
  Macintosh(std::string owner, std::string color) : Computer(owner) ,
  _color(color) {}
};
// PC.hpp
class PC : public Computer {
  unsigned int _weight;
  virtual void printSpecific() const override {
      std::cout << _weight << "kg PC\n";
  }
public:
  PC(std::string owner, unsigned int weight) : Computer(owner) ,
  _weight(weight) {}
};
```

<u>פתרון שאלה 2</u>

א. לא נכון.

ב. נכון.

ג. לא נכון.

<u>פתרון שאלה 3 א</u>

```
A virtual table:
A::g()
A::h()
B virtual table:
B::g()
A::h()
B::~B()
C virtual table:
B::g()
C::h()
C::~C()
```

פתרון שאלה 3 ב

```
Part 1:
(5)
Part 2:
(1)
Part 3:
(6)
(6)
(9)
Part 4:
(7)
Part 5:
(4)
(4)
(10)
(8)
(4)
```

פתרון שאלה 4

הפלט יהיה 15. הסבר: ה const על המטודה גורם לכך שהיא תקבל מצביע this שהוא מטיפוס const IntBuffer*. כלומר, לא ניתן יהיה לשנות

דרכו את הערך של השדות אבל יהיה ניתן לשנות דרכו את מה שהשדות מצביעים עליו.

פתרון שאלה 5:

```
template<typename InputIter1, typename InputIter2, typename OutputIter>
void mul_elements(InputIter1 begin1, InputIter1 end1,
                  InputIter2 begin2,
                  OutputIter begin3) {
  while (begin1!=end1) {
      (*begin3)= (*begin1) * (*begin2);
      ++begin1;
      ++begin2;
      ++begin3;
  }
}
```

פתרון שאלה 6:

```
int
A<int>*
A<A<int>*>
A<int>
```