Dry Part – Ex.3

סעיף א'

13 out << "B: " << n;

Error Description:
n is a non-static variable, therefore you cant access it without an object.
Solution:
13 out << "B: " << b.n;

22 if(b1 < b2)

Error Description:

b1 is a const B object and "b1 < b2" → "b1.operator<(b2)" is not a const function, therefor the compiler won`t allow using < operator on a const object (b1)

Solution:

16 bool operator < (const B& rhs) **const**;

30 const B b4 = b1 + (b2 + b3);

Error Description:

Operator + as defined in line 9 receives the right parameter by reference and returns by value.

(b2 + b3) executes first and returns a temporary object (not assigned anywhere) – than that same temporary object is transferred by **reference** to b1.operator+(b2+b3).

We cannot transfer by reference a temporary object.

Solution:

Assign a temp object:

const B temp = b2 + b3;

const B b4 = b1 + temp;

סעיף ב'

רצף ההדפסות:

applying function f:

A copy ctor

This is A

A copy ctor

This is A

A dtor

A dtor

applying function g:

This is B

This is B

B dtor

A dtor

ריצת התכנית:

A* pa = new B();

קריאה לבנאי של B שקורא לבנאי של A.

cout << "applying function f:" << endl;

הדפסה לערוץ הפלט הסטנדרטי

f(*pa).type();

pa* הוא אובייקט ומועבר לפונקציה by value ל – f. לכן בתחילת הפונקציה, מתבצעת קריאה לבנאי ההעתקה של A (כי f מצפה לקבל כפרמטר אובייקט מסוג A) – בעצם מתבצע פה slicing.

כעת הקריאה לפונקציה type בתוך f מתבצעת על אובייקט מסוג A ולכן תתבצע קריאה ל ()type של A ולא של B.

כעת, כדי להחזיר את a כפרמטר by value מתבצעת קריאה לבנאי העתקה שמעתיק את a ומחזיר אובייקט זמני שלו ל main

האובייקט המועתק שחזר גם הוא מסוג A ולכן מתבצעת קריאה לפונקציה ()type של A ולא של B גם מה main.

כשחזרנו מפונקציית ה ()type ל main הגענו ל "end of the line" ולכן מתבצעת קריאה להורסים.

תחילה להורס של A שהורס את האובייקט a שהועבר כפרמטר by value בתחילת השורה לפונקציה f

ולאחר מכן לערך המוחזר מ f, גם הוא by value שהועתק ולא נשמר בשום מקום ב main - כלומר אובייקט זמני.

cout << "applying function g:" << endl;

הדפסה לערוץ הפלט הסטנדרטי

g(*pa).type();

*pa מועבר כרפרנס לפונקציה g ולכן לא מתבצעת קריאה לבנאי העתקה.

(*) מכיוון ש type() היא פונקציה וירטואלית, מתבצעת בדיקה בזמן ריצה ומתקבל כי a הוא אובייקט מסוג B ולכן תפעל הפונקציה type() של B שדורסת את type() של A.

הפרמטר a מועבר חזרה גם כרפרנס ולכן גם כאן לא מתבצעת קריאה לבנאי העתקה.

כעת מאותם נימוקים ב (*) תתבצע בדיקה בזמן ריצה לגבי האובייקט שחזר כרפרנס מהפונקציה g

(g(*pa)) ותתבצע קריאה ל type() של B.

delete pa;

בעת פינוי המצביע pa מתבצעת קריאה להורס, בגלל שההורס מוגדר כוירטואלי בזמן ריצה מתבצעת בדיקה ומתגלה כי *pa הוא אובייקט מסוג B ולכן מתבצעת קריאה להורס של B שקורא להורס של A.

שאלה 2:

```
class Road {
public:
    double length(); // km
    int speed(); // km per hour
};


class Car {
public:
    virtual double getFuelConsumption(int speed) const = 0;
};

double getPetrol(std::vector<Road> roads, const Car& car) {
    double petrol_sum = 0;
    for(Road road : roads) {
        double km_per_litre = car.getFuelConsumption(road.speed());
        petrol_sum += road.length() / km_per_litre;
    }
    return petrol_sum;
}
```