**חלק היבש**

1. תכנות גנרי:א. הקוד:

**template** <**class** container, **typename** Predicate>  
**int** countPairs(container first, **const** container last, Predicate predicate) {  
 **int** count = 0;  
 **for** (container iterator1 = first; iterator1 < last; ++iterator1) {  
 container iteratorToHelp = iterator1;  
 ++iteratorToHelp;  
 container iterator2;  
 **for** (iterator2 = iteratorToHelp; iterator2 < last; ++iterator2) {  
 **if** (predicate(\*iterator1, \*iterator2))  
 count++;  
 }  
 }  
 **return** count;  
}

ב. הקוד:

**bool** isBigger(**int** a, **int** b){  
 **return** (b > a);  
}  
  
**bool** isSorted(vector<**int**>& v){  
 **int** count = countPairs(v.begin(), v.end(), isBigger);  
 **int** sum = 0;  
 **for** (**int** i = 1; i < v.size(); ++i) {  
 sum = sum + i;  
 }  
 **return** (count == sum);  
}

2. ירושה ופולימורפיזם:א. רשמו מה ידפיס הקוד הנ''ל ונמקו מדוע.

הקוד ידפיס:

from Base

from A

from Base

נימוק: 1. ;()Base\* base = new A: השורה הזאת קוראת לבנאי המחלקה היורשת B ואז יתחיל מאתחול מחלקת הבסיס Base, לאחר מכן יבצע את רשימת האתחול של הבנאי של A, אבל הבנאים הם דיפולטיבים ולהן רשימות האתחול שלהם ריקה ולכן מדפיס כלום.

2. ;()base -> baseMethod: השורה הזאת קוראת לפונקציה method(); של האובייקט על פי הטיפוס הסטטי של המצביע אליו כלומר Base ולכן ידפיס: from Base.

3. delete base;: השורה הזאת קוראת להורס של המחלקה A כך שיבצע קודם את ההורס של A, ובסוף ייקרא להורס של Base.

ב. כיצד ניתן לשנות את השורה הראשונה בפלט על ידי שינוי המחלקה Base בלבד? נמקו את תשובתכם.

נוסיף virtual לפני הפונקציה method();שבמחלקת Base, כלומר המחלקת Base תראה עכשיו כך:

**class** Base  
{  
 **virtual void** method() {std::cout << **"from Base"** << std::endl;}  
**public**:  
 **virtual** ~Base() {method();}  
 **void** baseMethod() { method();}  
};

נימוק: עכשיו השורהbase -> baseMethod(); תקרא ל-method(); של המחלקה האמיתית של האובייקט על פי הטיפוס הדינאמי שלו שהוא A, והשאר יישאר אותו דבר ולכן הקוד ידפיס:

from A

from A

from Base

ג. בתגובה לביצוע שורת הקודBase\* base = new A(); ברצוננו לקבל את הפלט הבא:

from Base

from A

הראו כיצד ניתן לקבל זאת בלי להוסיף ו/או לשנות פקודות פלט (אופרטור >>).

נוסיף בנאי לכל אחת מהמחלקות כך שיכילו רק את הפונקציה method();, כלומר הקוד עכשיו יראה כך:

**class** Base  
{  
 **void** method() {std::cout << **"from Base"** << std::endl;}  
**public**:  
 Base(){method();}  
 **virtual** ~Base() {method();}  
 **void** baseMethod() { method();}  
};  
**class** A: **public** Base  
{  
 **void** method() {std::cout << **"from A"** << std::endl;}  
**public**:  
 A(){method();}  
 ~A () { method();}  
};

נימוק: עכשיו השורה Base\* base = new A(); תקרא לבנאי של מחלקה יורשת A כך שיתחיל מאתחול   
מחלקת הבסיס Base, ולאחר מכן יבצע את רשימת האתחול של הבנאי של A ולכן השורה תדפיס:

from Base

from A