

תכנות מע' דפנסיבי – ממ"ן 11

מאת: יאיר חריט 207282955

1. לאחר הרצת הקוד הנתון יודפס `Foo::baz()`.
בכדי להבין מדוע נסתכל על סדר הפעולות המתבצעות בפועל:
 - `New Bar()` - תחילה קוראים לפעולה הבונה של `Bar`
 - לפני שהפעולה הבונה של `Bar` תתבצע - יקרא הבנאי של המחלקה `Foo` (מאחר `Bar` יורש מ-`Foo`)
 - מתוך הפעולה הבונה של `Foo` תקרא הפונקצייה `baz()`. מאחר והפונקצייה נקראת מתוך המחלקה `Foo` - יעשה שימוש בפונקציית `baz` של `Foo` (לא יופעל מנגנון הפולימורפיזם משום שבתוך המחלקה `Foo` האובייקט בכלל אינו מודע למחלקה `Bar`...)

2. מאחר ו"בנאי ההעתקה" מצביע לאותן קורדינטות של האובייקט שקיבל ואינו מקצה זיכרון חדש עבור הקורדינטות של האובייקט החדש - הקורדינטות (`_coord`) של 2 האובייקטים יחלקו את אותו המקום בזכרון ולמעשה יהיו **ממש אותן קורדינטות**, כלומר, כאשר נשנה קורדינטות של אובייקט אחד נשפיע גם על הקורדינטות של האובייקט השני, הם יחלקו זכרון משותף.
יתר על כך, בעת מחיקת האובייקטים יהיה נסיון למחוק פעמיים את אותו הזכרון (`_coord`) וזה יהיה שגיאת סגמנטציה (`segmentation fault`).

ניתן לתקן באג זה על ידי תיקון הבנאי המעתיק, יש להקצות זכרון חדש ל-`_coord` של האובייקט החדש והשמת הערכים בהתאמה:

```
Point(const Point& other)
{
    _coord = new int[2];
    _coord[0] = other._coord[0];
    _coord[1] = other._coord[1];
}
```