

## ממן 11 שאלה 1

בשאלה זו הקוד שיודפס הוא "Foo::baz()". במקרה הכללי, כאשר יש מחלקה יורשת ממחלקה אחרת ובמקרה שיש פונקציה בעלת אותה חתימה ב++C אופן הפעולה הסטנדרטי הוא שפונקציית במחלקת האב תפעל (early/static binding), אך כאשר נרצה שפונקציית הבן תפעל אז נצטרך לציין את לקומפיילר על ידי המילה השמורה virtual (late/late/dynamic binding). יש מקרה פרטני לפונקציה וירטואלית שבו הפונקציה שתקרא היא דווקא הפונקציה במחלקת האב וזה קורה כאשר יוצרים אובייקט בהקצאה דינמית כאשר בבנאי המחלקת האב יש קריאה לפונקציה וירטואלית. מקרה זה דומה למקרה הזה ולכן יודפס "Foo::baz()".

נתאר את התהליך הריצה:

השורה הזו נקראת: `Foo *pFoo = new Bar();`, תחילה נכנס לבנאי של המחלקה Bar, מכיוון שזו מחלקה יורשת אז ישר אחרי שנכנס לבנאי הזה נלך לבנאי של מחלקת האב Foo, בבנאי הזה יש קריאה לפונקציה וירטואלית baz אך מכיוון שכרגע האובייקט הוא מטיפוס Foo אז הפונקציה baz שתופעל היא הפונקציה של המחלקה Foo (הקומפיילר "יודע" רק על הפונקציות של המחלקה הזו), לאחר מכן יודפס "Foo::baz()" ונסיים לבנות את האובייקט שהוא מטיפוס Foo. נחזור לבנאי של Bar ונסיים לבנות את האובייקט, כלומר רק עכשיו האובייקט הוא מטיפוס Bar. בסוף נשייך את האובייקט Bar שיצרנו למצביע `*pFoo` מטיפוס Foo (פולימורפיזם).

בשורה הזו: `delete pFoo;` בעצם אנו מוחקים את ההקצאה הדינמית שעשינו, כלומר נפעיל את הדיסטרוקטור, במנגנון הירושה סדר הפעלת הדיסטרוקטורים הוא בסדר הפוך לסדר הפעלת הבנאים ולכן הדיסטרוקטור שיופעל ראשון הוא הדיסטרוקטור של Bar ולאחר מכן יופעל הדיסטרוקטור של Foo.

בשורה: `return 0;` בעצם נחזיר 0 בפונקציה הראשית ונסיים את הריצה של התוכנית.