**על מנת לקרוא לרוטינה שקומפלה כתוכנית C מתוך תוכנית C++ יש צורך להכריז הכרזת extern עם "C". לדוגמא**

**extern "C" int f(int);**

**ואז לא ניתן להעמיס את השם f יותר.**

**פונקציות inline**

* **מקרו שנראה כמו פונקציה רגילה**
* **כל "קריאה" לפונקצית inline פורשת קוד בינארי במקום (ללא הסתעפות ליעד כלשהוא).**
* **יוצר קוד מהיר יותר במחיר קוד גדול יותר.**
* **קומפילרים מגבילים מה יכול להיות בפונקצית inline.**
* **מוטיבציה המקורית היתה לעודד מתכנתים לממש מתודות set ו-get.**

 **תכנות OOP ב-C++**

**קוד של אובייקטים, המסורת היא לממש אותם תחת מילת המפתח class, שהוא דומה מאד ל-struct כאשר ההבדל היחיד הוא ברירת המחדל לרמת הנגישות של השדות:**

**class**

**רמת נגישות ברירת מחדל היא private**

**struct**

**רמת נגישות ברירת מחדל היא public**

**ב-C++ רמת הנגישות מוגדרת ע"י labels**

**למשל**

**private:**

**עד לליבל אחר, כל שדה, מתודה כאן יהיה ברמת נגישות private.**

**הבדל נוסף מ-C.**

**ב-C++ אם אני מגדיר**

**struct my\_stc**

**{**

**}**

**אני יכול להגדיר משתנה מסוג הרשומה**

**my\_stc x;**

**ב-C הייתי חייב**

**struct my\_stc x;**

**אפשר לעשות זאת גם עם class.**

**פונקציה ההורסת destructor**

**ל-C++ אין "אוסף פסולת" ברקע, בכל מחלקה שמכילה מצביעים לשטחים מוקצים דחנמי, יש לממש פונקציה הורסת שמשחררת אותם.**