**הורשה בC++**

**רעיון ייחודי ל-C++:**

**סוג ההורשה (public, protected, private) ברירת המחדל private:**

**class child : סוג ההורשה parent**

**לדוגמא:**

**class a: public b**

**סוג ההורשה קובע את הנגישות המרבית של שדות האב בקוד הלקוח של הבן.**

**סוג ההורשה לעולם לא מעלה את רמת הנגישות של שום שדה.**

**כל סוג הורשה חוץ מ-public הגיוני רק כאשר ההורשה הוא עניין פנימי של המימוש של המחלקה הנגזרת.**

**כל סוג הורשה חוץ מ-public בעצם ויתור על פולימורפיזם:**

**לא ניתן לדרוס מתודה כאשר ההורשה איננה public.**

**הבדלים נוספים:**

* **כיוון שב-C++ יש הורשה מרובה, אין אפשרות למקביל ל-super או base. התייחסות למחלקת האב הוא לפי השם של מחלקת האב.**
* **שפת C++ תומכת בדריסה וירטואלית וסטטית (כמו ב-C#) השאלה האם מתודה היא וירטואלית או לא נקבע ברמה הראשונה בה היא מוכרזת. מילת המפתח virtual יכולה להופיע בקוד של הבן, אבל הוא רק בקשה לבדיקת נכונות / תיעוד.**
* **ניתן להציב ("=") מופע של הבן לתוך משתנה של האב, התוצאה תהיה משתנה אב שיכיל בשדות האב את ערכי מופע הבן.**

**הרכבת מחלקות**

**בנאים של מופעים של מחלקות מרכיבות חייבות להיקרא ברשימות אתחול לפי שמות המשתנים.**

**טמטפלטים – תכנות המשאיר פתוח את סוג המשתנה**

**קוד טמפלטי הוא מקרו - אין לו השתקפות בינארית.**

**לפיכך אם רוצים לממש טמפלטים בקובץ נפרד, חייבים לעשות לו #include, המסורת לעשות בקובץ.hpp**