

1. הדרישות ההכרחיות שעל הטיפוס T לקיים הן:
 - א. האופרטור < . על מנת להשוות שני עצמים מטיפוס T, יש לספק את האופרטור < (לצורך מיון הרשימה בסדר עולה)
 - ב. copy constructor כדי ליצור עותק חדש של עצם מסוג T.
 - ג. Destructor כדי למנוע דליפות זיכרון.
2. הבעיה שעלולה להיווצר לו היינו ממשמים non-const iterator היא שיתאפשר למשתמש גישה לערוך את הרשימה "מבפנים" ובכך לפגוע בסדר של הרשימה, ולהפוך אותה לרשימה לא ממוינת.
3. דרך א': בעזרת Function objects. ניתן להעביר את הפרדיקט כאובייקט Fucntion Object.
דרך ב': בעזרת מצביעים לפונקציות. ניתן להעביר את הפרדיקט כמצביע לפונקציה, (אשר המשתמש כתב) ולקרוא לפונקציה הזאת.
ההבדל בין שני הדרכים הללו הוא בסינטקס. כאשר שימוש בFunction Objects הוא מאוד נוח לקריאה ולהבנה הלוגית של הקוד.
אנחנו לא צריכים לספק שני מימושים שונים כדי לתמוך בשני הדרכים הללו כיוון שאין צורך להשתמש במצביעים לפונקציות בשפת ++C כי השימוש בFunction Objects מאפשר למשתמש להעביר כל אובייקט (יכול להיות פונקציה) שתומך באופרטור סוגריים. כלומר, כל מקרה שבו אפשר לפתור בעזרת מצביע לפונקציה, ניתן לפתור זאת במקום בעזרת Function Objects.