

1.

reference ב ++c

זה כמו לתת שם שני למשתנה  
כשאתה יוצר משתנה שהוא reference זה כמו ליצור עותק של המשתנה המקורי כל פעולה שבוצע על  
המשתנה של reference תתבצע גם על המשתנה המקורי, משתנה reference מצביעה לאותו מקום בזיכרון  
כמו המשתנה המקורי.  
משתנה reference ממש מזכיר את הפוינטרים רק בטיחותי יותר ואם כמה הגבלות.

2. יתרונות

חיסכון בזיכרון וחיסכון במשאבי המחשב, במקרים איפה שמעבירים reference ולא משתנה רגיל זה חוסך  
למחשב ולמעבד את העבודה של לעשות את העתק של המשתנה ובכך חוסך זיכרון ומשאבים של הcpu

כאשר מעבירים by reference, כל פעולה על המשתנה בפונקציה מעדכנת ישר את המשתנה וזה מונע את  
הצורך להחזיר ערכים מהפונקציה

3. למרות שreferences דומים מאוד לpointers יש ביניהם הבדלים כמו:

1. אי אפשר להכריז על reference כ משתנה מסוג void ואי אפשר להגיד references שווה לnull אך  
אפשר לעשות את הדברים האלו על pointers
2. כאשר אנחנו מגדירים pointers אפשר לעשות אין סוף פוינטרים לפוינטרים לפוינטרים אך  
בreferences אפשר להגיד רק reference למשתנה המקורי ואי אפשר להגיד reference  
לreference גם כן מאוחר יותר בקוד אי אפשר לשנות לאיפה הreferncen מצביעה

בגלל זה reference נחשב לבטוח יותר כי  
תמיד צריך להתחל refernces ובגלל שהם חייבים להתייחס למשתנה קיים יותר קשה להסתבך עם  
references שישברו לך את הקוד

4.

1. תקין

2. לא תקין, כשמעבירים משתנה שהוא כבר reference לפונקציה שמקבלת reference לא צריך לשים את  
ה& בהתחלה של המשתנה ולכן זה יציג שגיאה

3. לא תקין, int& result אמור לקבל משתנה שהוא reference כשאנחנו מעבירים לו מספר רגיל  
המספר הוא לא משתנה ולא reference ולכן זה יציג שגיאה

5. 1. הפונקציה תנסה להחזיר reference לאיקס אך בגלל שאיקס כמשתנה קיים רק בscope של הפונקציה  
כאשר הפונקציה תגמר היא תחזיר reference למקום לא יודעה בזכרון
2. הפונקציה מחזירה רפרנס למערך ולא פוינטר, בגלל שאנחנו לא עושים שום delete זה יכול להביא לזליגות  
זיכרון