

1. Reference ++c הוא שם חלופי בשביל משתנה אחר. הרפרנס מביא גישה ישירה ואפשרות שינוי המשתנה המקורי שקשור לאותו רפרנס זה כמו שם כינוי למשתנה הרגיל. `type& reference = var`

2. (א) בעזרת רפרנס ניתן לפונקציה לשנות ישירות את המשתנה מה שבלתי אפשרי כשעושים בי וואליו

(ב) בעזרת רפרנס הפונקציה פועלת ישיר על המשתנה המקורי בלי ליצור עותק של המשתנה.

3. ההבדל בין פוינטר לרפרנס הוא שרפרנס הוא נחשב לכינוי או שם אחר למשתנה ופוינטר מחזיק את הכתובת של המשתנה והרפרנס מחזיק את אותו זכרון כמנ המשתנה אליו הוא מיוחס.

רפרנס נחשב לבטוח יותר משום שם לרפרנס חייב לאתחל למשנה שאנו רוצים ואי אפשר לא לאתחל אותו כמו פוינטר , רפרנס לא יכול להיות NULL לעומת פוינטר בנוסף אי אפשר להחליף את המשתנה שהרפרנס קשור אליו.

4. הראשון יעבוד משום שבסופו של דבר רפרנס הוא מספר ובנוסף על זה רפרנס יושב על אותו זכרון של המשתנה הרגיל אז גם אם שמים שם משתנה רגיל זה יעבוד

השני לא יעבוד משום שץ הוא משתנה ולא רפרנס אז אם קוראים לץ בתור רפרנס אבל אין רפרנס כזה כי ץ הוא משתנה רגיל

השלישי לא יעבוד `cannot bind non-const lvalue reference of type 'int&' to an rvalue of type 'int'`

`A non-const lvalue reference (int&) can only bind to lvalues (e.g., variables).`

`It cannot bind to rvalues`

(בדקתי באיזה אתר בגוגל מה זה מה)

5. בפונקציה הראשונה מחזירים ערך אשר נמחק לאחר סיום הפונקציה בעצם VALUE אבל בפונקציה מחזירים ערך רפרנס שאמור להישאר בזיכרון ואמור להיות קשור למשתנה אבל בגלל שמחזירים VALUE פשוט אז זה לא הגיוני

בפונקציה השנייה אנחנו מחזירים פוינטר זה כן עובד אבל יש בעיה עם זה
איך אפשר לשחרר את הזכרון שהוקצע בתוך הפונקציה מה שיגרום
לדליפת זיכרון