

1. מהו reference וכיצד מגדירים reference למשתנה?

משתנה reference הוא כינוי למשתנה אחר. הוא לא יוצר משתנה חדש, אלא מתייחס למשתנה קיים. כדי להגדיר reference, נשתמש בסמל & אחרי סוג המשתנה.

2. מנו שני יתרונות בהעברת משתנים לפונקציה by reference על פני שיטות אחרות.

1. מעבר לפי reference מונע את הסיכוי להעתיק אובייקטים או מערכים גדולים לפונקציה, מה שיכול להיות יעיל יותר עבור מבני נתונים גדולים.
2. **שינוי ערכים מקוריים**: מאפשר לפונקציות לשנות את המשתנה המקורי, שיכול להיות שימושי במצבים שונים.

3. מה הם ההבדלים בין reference pointer? מדוע reference נחשב ל"בטוח" יותר?

references נחשבות בדרך כלל בטוחות יותר מכיוון שאינן יכולות להיות null ולא ניתן להקצותן מחדש להצביע על משתנה אחר אחרי האתחול. אבל מצביעים יכולים להיות null וניתן להקצותם מחדש, מה שעלול להוביל לשגיאות אם לא מטפלים בזהירות.

references לא יכולות להיות ריקות, בעוד שמצביעים יכולים.
references חייבות להיות מאותחלות בהצהרה, בעוד שמצביעים יכולים להיות מוצהרים ללא אתחול.

```
void square(int x, int& result)
{
    result = x * x;
}
```

האם הקריאות הבאות תקינות? אם לא, הסבירו מה הבעיה.

- א. `square(3, y);`
- ב. `square(3, &y);`
- ג. `square(3, 6);`

א, נכונה

- ב, העברת `y&` לפונקציה אינה נכונה. הפונקציה מצפה לreference של מספר שלם
parameter, אבל כאן מעבירים את הכתובת של מספר שלם.
- ג, לא נכון, אי אפשר להעביר ערך ליטרלי כ reference.

5. מה הבעיה בכל אחת מהפונקציות הבאות :

a.

```
int& getLocalVar()
{
    int x = 10;
    return x;
}
```

מחזירה reference למשתנה מקומי, שמתיישן לאחר שהפונקציה חוזרת.

b.

```
int& getDynamicVar()
{
    int *x = new int(10);
    return *x;
}
```

מחזירה reference למשתנה שהוקצה דינמית, אך הזיכרון לא מנוהל כראוי, מה שמוביל לדליפת זיכרון.