

שלב 1 - לימוד עצמי: האופרטור &

שאלה 1 - מהו reference וכיצד מגדירים reference למשתנה?

רפרנסים (הפניות) מאפשרים לנו ליצור שם שני עבור משתנה שישמש בלקרוא או לשנות את המידע (הנתונים) המאוחסנים באותו משתנה.

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

יתרון ראשון - מתייחסים להפניה כאילו היא המשתנה המקורי ונוכל לשנות ולערוך אותו גם אם ההפניה ממוקמת מחוץ לפונקציה או בתוכה.

יתרון שני - מאפשר לנו להפחית משמעותית את כמות ההעתקה שקורה "מאחורי הקלעים" כלומר פשעושים זאת זה חוסך משמעותית בזיכרון ובביצועים ומייעל את התוכנית שלנו

**שאלה 3 - מה הם ההבדלים בין pointer ל reference?
מדוע reference נחשב ל"בטוח" יותר?**

ההבדל הוא שפוינטר הוא משתנה שמכיל בתוכו את הכתובת (ביכרון) למשתנה אחר (ערך מסוים) וצריך להתעסק ולהתייחס אליו בדרך אחרת (למשל להקצות זיכרון ולהיזהר מזה) לעומת זאת, להפניה מתייחסים כאל משתנה רגיל ולא צריך להקצות זיכרון.

הפניה נחשבת לבטוחה יותר מפני שלא צריך לדאוג להקצות לו זיכרון וכך לא יהיו בעיות לא צפויות.

שאלה 4 - נתונה הפונקציה ונתון משתנה y מסוג int

```
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);

סעיף א - תקין

סעיף ב - לא תקין, לא נכלל לשלוח כתובת (&), כי הפונקציה חוצה לקבל הפניה למשתנה ולא לכתובת.

סעיף ג - לא תקין, מועבר ערך ולא משתנה הפונקציה לא תקבל זאת ויהיה שגיאה

שאלה 5 - מה הבעיה בכל אחת מהפונקציות הבאות:

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

a.

הבעיה היא שפונקציה לא יכולה להחזיר הפניה למשתנה שהוגדר בתוך הפונקציה (למדנו שנה שעברה שבין הסוגיים המסולסלות, לאחר הסוגיים כל המשתנים המוגדרים בתוכה "מתים" וגם הכתובת שלהם משתחררת - לא תקפה יותר).

b.

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

הבעיה היא שפונקציה מסוכנת (עלולה להוביל
לדליפת זיכרון - כמו שלמדנו ב C), הפונקציה
מחזירה הפניה לכתובת שהוקצתה דינמית אבל
הזיכרון לא מנוקה אוטומטית (לכן נצטרך לעשות
delete בצורה ידנית).

שאל אטלי