

שלב 1: לימוד עצמי – האופרטור &

שלב 6 מחלקת Social Network	שלב 5 מחלקת Profile - i Page	שלב 4 מחלקת User	שלב 3 מחלקת Device	שלב 2 מחלקת מחזור	שלב 1 האופרטור &
-------------------------------	---------------------------------	---------------------	-----------------------	----------------------	---------------------

למידה עצמית

קראו ב**מאמר הבא** על המושג reference וענו על השאלות הבאות:

1. מהו reference וכיצד מגדירים reference למשתנה?

רפרנס זו גישה ישירה למשתנה המקורי (גם בפונקציה), כשמעבירים משתנה ללא רפרנס נוצר עותק של המשתנה וכך כל שינוי שתעשה הפונקציה לא תשפיע על הערך המקורי, לעומת זאת אם מעבירים משתנה עם רפרנס בעזרת אופרטור "&" אז כל פעולה תתבצע על הערך המקורי של המשתנה ותעדכן אותו

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

- פחות העתקות (יעילות): כשעובדים עם רפרנסים, אין צורך להעתיק את ערכי המשתנים שוב ושוב. זה חשוב במיוחד אם עובדים עם מבני נתונים גדולים (כמו מערכים או מחלקות כבדות).
- שימוש בסקופים שונים:
- הרפרנס יכול להיות בשימוש גם אם הוא הוגדר במקום אחר בקוד. הוא מאפשר גישה ישירה למשתנה המקורי, גם אם זה קורה בתוך פונקציות שונות או קבצים שונים.

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

- Pointers נותנים גמישות רבה יותר: אפשר לשנות את הכתובת שהם מצביעים אליה או להשאיר אותם ריקים (null). הם שימושיים במצבים שבהם יש צורך בגמישות הזו, כמו עבודה עם מערכים דינמיים או פונקציות שמחזירות null.
- References נותנים בטיחות ונוחות: הם תמיד מקושרים למשתנה תקף, פשוטים יותר לשימוש, ומונעים טעויות נפוצות כמו גישה לכתובת לא חוקית.
- רפרנס חייב להיות מקושר למשתנה מההתחלה (אי אפשר להגדיר רפרנס בלי להגדיר אותו למשתנה), רפרנס תמיד נשאר מקושר לאותו משתנה זאת אומרת שאחרי שהגדרנו רפרנס אי אפשר לגרום להם "להצביע" למשתנה אחר, אין סיכון לטעויות כתוצאה משימוש לא נכון בכתובת ויש פחות סיכון לדליפות זיכרון

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;`

א. תקינה

ב. לא תקינה, כשהפונקציה מוגדרת לקבל רפרנס אין צורך להוסיף "&" בקריאה לפעולה. מכיוון שקריאה זו מייצגת את הכתובת של המשתנה

ג. לא תקינה, השגיאה היא שגיאה תחבירית לא רק בגלל שהסוגיים במקום לא טוב גם המספר 6 הוא קבוע ואי אפשר להעביר קבוע ברפרנס, מכיוון שרפרנס נועד למשתנים

5. מה הבעיה בכל אחת מהפונקציות הבאות:

a.

```
int& getLocalVar()
```

```
{
```

```
    int x = 10;
```

```
    return x;
```

```
}
```

b.

```
int& getDynamicVar()
```

```
{
```

```
    int *x = new int(10);
```

```
    return *x;
```

```
}
```

a.

הבעיה בפונקציה היא שהיא מחזירה רפרנס למשתנה מקומי. משתנים מקומיים מוגדרים בתוך הפונקציה ונשמרים על הסטאק שלה וכשהפונקציה מסתיימת הסטאק מתנקה מה שאומר שהמשתנה כבר לא קיים והכתובת שהרפרנס "הצביע" עליה כבר לא קיימת

b.

הפונקציה יוצרת משתנה דינמי (איקס) על ההיפ ומחזירה רפרנס לערך שאליו מצביע המשתנה הבעיה היא שהפונקציה לא משחררת את הזיכרון הדינמי שנוצר בה והדבר הזה עלול לגרום לדליפת זיכרון