

## שאלות נכון\לא נכון – הכנה למבחן בעקרונות שפות תוכנה

### מועד א':

א. פונקציה מסדר גבוה ( high-order function ) זוהי פונקציה מובנת ( built-in ) של שפה, שמקבלת פונקציות משתמש כארגומנטים.

לא נכון. זוהי לא בהכרח פונקציה מובנת. יכולה להיות גם פונקציית משתמש.

ב. ב- Lexical Scoping ניתן להחזיר פונקציה מפונקציה אחרת, אך הפונקציה המוחזרת תהיה ללא סביבה. לא נכון. תמיד פונקציה מוחזרת עם סביבה.

ג. לפי Lexical Scoping הערכה של הפעלת פונקציה מתבצעת בסביבה חדשה שמרחיבה את הסביבה של הפונקציה המופעלת. נכון.

ד. מפרש של שפות פשוטות אינו חייב לממש רקורסיה הדדית בין eval ו- apply .  
נכון. לדוגמא: במימוש המפענח של המחשבון שלנו (Calculator) מימשנו את כל הלוגיקה בפונקציית 'apply' ללא קריאות רקורסיביות ל 'eval'.

ה. ניתן באופן כללי לקחת פונקציה רקורסיבית המבצעת חישובים חוזרים על אותם פרמטרים ולשפר את זמן הריצה שלה ללא שינוי קוד הפונקציה (בשפה עם פונקציות שהן אובייקטים מסדר ראשון).  
נכון – זה נקרא memoization.

### מועד ב':

א. ב Python 3 משתמשים בהצהרה global - על מנת לעדכן קשירה של משתנה במסגרת X מעל המסגרת הנוכחית (X) אינה בהכרח מסגרת גלובלית). לא נכון.

ב. אובייקטים מסדר ראשון (first-order object) ניתן להחזיר מפונקציה ולהעביר לפונקציה כארגומנט. נכון.

ג. בשפות עם Lexical Scoping ניתן לממש טיפוסים נתונים חדשים שהם immutable (כמו tuple) תוך שימוש בפונקציות בלבד. נכון. לדוגמא כיצד שמימשנו את 'pair'.

ד. לפי מודל הסביבות (Environment Model) הגדרת פונקציה חדשה מוסיפה קשירה חדשה לסביבה גלובלית. לא נכון. הקשירה יכולה להיות לכל סביבה ולא דווקא לסביבה הגלובלית. זה תלוי היכן הוגדרה הפונקציה (ייתכן שפונקציה הוגדרה בתוך פונקציה אחרת).

ה. מפרש של שפות תכנות כמו Python מזמן את פונקציית eval מפונקציית apply על מנת להעריך את הביטוי המתקבל מהצבת ערכי ארגומנטים לפרמטרים פורמליים של הפונקציה המופעלת. נכון.