בס"ד

**נקודות להתייחסות במצגת – עקרונות שפות תוכנה**

1. היסטוריה של השפה, הרקע לפיתוחה.
2. מטרות ומאפייני השפה
3. **יתרונות וחסרונות**
4. **לאילו פרדיגמות השפה משתייכת ולמה**? (פרוצדוראלית, אימפרטיבית, מונחית עצמים, פונקציונאלית, לוגית, מקבילית)
5. **נתוח ביקורתי על פי המושגים שנלמדו בהרצאה** :

Readability, Writability, Orthogonality, Reliability, Cost, Portability, Well-definedness,

1. טיפוסים בסיסיים (char, int,…). טיפוסים מורכבים (מערכים, רשומות,enum, ...) וטיפוסים שמוגדרים על ידי המשתמש.
2. **טיפוס נתונים** דינאמי (נקבע בזמן ריצה) או סטטי (נקבע בזמן קומפילציה)
3. **מנגנון טיפוסים** חזק=המרה מפורשת / חלש =המרה מרומזת duck typing
4. פונקציות ופרוצדרות, רקורסיות
5. העברת **ערך** הפרמטרים (By Value, By Reference, By Pointer, By Name)
6. העברת **סדר** פרמטרים (By Position, By Keyword)
7. תחביר השפה (בקצרה): מבנה של מזהים, קבועים וביטויים, מבנה של פקודות, מבני בקרה וזרימה ומבנה תוכנית המורכבת מהם. BNF של המבנים העיקריים וכן דוגמאות פשוטות לשימוש בהם בשפה.
8. **טווח הכרה** (scope) סטטי (לפי עץ התחביר הסטטי) או דינאמי (לפי סדר הקריאה לפונקציות הדינאמי)
9. הקצאת נתונים דינאמיים
10. חישובים, השמות וסוגי אופרטורים.
11. פונקציה טהורה – שאין לה side effects כלומר כל הרצה של פונקציה עם אותם ערכם תחזיר את אותו ערך.
12. ביטויים רגולריים
13. מצביעים
14. **איסוף זבל**. איזה סוג של ניהול זיכרון נתמך ע"י השפה ואיך זה בא לידי ביטוי במבנים תחביריים של השפה? איך מנוהלים הקצאה ושחרור של זיכרון בכל אחד מסוגי במשתנים
15. הפשטה, האבחנה בין השימוש למימוש. באיזה סוג של הפשטות (procedural abstraction, data abstraction, וכו') השפה תומכת? איך זה בא לידי ביטוי במבנים התחביריים של השפה? פרוצדורה, אובייקט, class, וכו'.
16. האם השפה **תומכת במושג של module, information hidding , ירושה, פולימורפיזם, הכמסה** וכו',
17. מקביליות, tasks, threads, וכו'
18. טיפול בשגיאות (exceptions)
19. קבצים
20. מחרוזות
21. **ניידות השפה**