***חלק שלישי – תכנות מונחה עצמים***

1. **הסבירו מהו עקרון הכימוס?**

עקרון הכימוס - עקרון זה בעצם מאפשר הסתרת מידע ופעולות של אובייקט כל שהוא מפני אובקטיים אחרים, על ידי שימוש בנתונים כמו protected privet, .

עקרון זה מאפשר הגנה על האובייקט כך שרק דרך פונקציות ממשק שהוגדרו לאותו אובייקט יכלו לגשת אליו ובכך למנוע שינוים במאפייני האובייקט ובחוקו.

1. **מה ההבדל בין מחלקה למחלקה אבסטרקטית?**

ממחלקה אבסטרקטית ניתן ליצר אובייקטים, מחלקה אבסטרקטית מהבא כבסיס ומרכזת את התכונות המשותפות שיש למחלקות היורשות ממנה. מחלקה אבסטרקטית מכילה פונקציות כמו במחלקה רגילה אך ללא מימושן לא כמו במחלקה רגילה.

לעומת מחלקה רגילה מחלקה אבסטרקטית יכולה לכלול בתוכה גם פונקציה אבסטרקטית.

כאשר יורשים מחלקה אבסטרקטית יש לממש את כל הפונקציות האבסטרקטיות שיש בה.

1. **מה ההבדל בין מחלקה למחלקה סטטית?**

במחלקה סטטית כל הפונקציות היו מסוג סטטי, וכדי לממש את הרכיבים במחלקה זו יש רק להשתמש בשם המחלקה ולא כמו במחלקה רגילה אשר צריך ליצור מופע של המחלקה.

ממחלקה סטטית לא ניתן לרשת ולא ניתן מופעים מהמחלקה.

כדי לפנות לפונקציה מתוך המחלקה הסטטית יש לפנות לשם המחלקה וציון הפונקציה שלא כמו במחלקה רגילה בה הגישה היא דרך שם האובייקט וציון הפונקציה.

בדרך כלל משתמשים במחלקה סטטית כדי לעבוד עם ה Console.

1. **מי יכול לגשת לmembers בעלי access modifier של internal ?**

עצם הפיכת טיפוס או Member ל internal מונעת את חשיפתם לעולם החיצון. זה אומר שבד”כ מחוץ לAssembly שלנו, לא תהיה לנו גישה למחלקות או Memberים אלה, ורק למחלקות הנמצאות בתוך ה Assembly יכולות לגשת לפונקציה פנימית בתוך מחלקה אחרת.

הדבר הזה יכול ליצירת צמידות בין מחלקות, כי כעת מחלקה אחת מכירה את המימוש הפנימי של מחלקה אחרת. אם זו פונקציה שלא באמת פנימית, אין סיבה שלא להפוך אותה לpublic. אחרת, עדיף שזה יישאר protected/private.