1.

עקרון הכימוס:

עקרון הכימוס הינו עיקרון חשוב בעיקר בתכנות מונחה עצמים. מטרתו המרכזית של העקרון היא הימנעות משגיאות, והפיכת הקוד לקריא יותר וברור יותר ומודולרי. דבר זה בא לידי ביטוי על ידי הגדרת הגישות לנתונים של אובייקט במקום לעשות שהכל יהיה פומבי ונגיש מכל מקום דבר שעלול ליצור בלבולים במידה ויש משתנה עם אותו השם במחלקה אחרת לדוגמה, יצירת getters ו-setters אשר נותנים גישה למשתנים פנימיים אבל נוכל לשנות בהמשך את הדרך שבא הפעולות האלה משנות ומחזירות את המידע מבלי לגרום לשגיאות בקוד כי בשאר השימושים של המשתנה עדיין יקראו לאותה הפונקציה של הget או הset.

לסיכום, לפי עקרון הכימוס על מחלקות להסתיר את התוכן שהם לא צריכים להציג החוצה ואת הדרך שבה הוא פועל מאחרים על מנת להקל על השימוש בקוד, על הקריאה שלו, על השינוי שלו בעתיד מבלי לשנות את כל שאר הקוד בשאר המחלקות אשר משתמשות באותה המחלקה אשר שינינו ובכך גם למנוע מהרבה משתנים להיות פומביים כאשר הם לא צריכים ולמנוע שימוש כפול בשמות של משתנים ממחלקות שונות, דבר אשר עלול לגרום לשגיאות בקוד כתוצאה מחוסר הבנה.

2.

ההבדל בין מחלקה למחלקה אבסטרקטית הינו שמחלקה אבסטרקטית היא מחלקה אשר אך ורק מהווה בסיס עבור מחלקות אחרות אשר ירשו ממנה.  
במחלקה אבסטרקטית ניתן להגדיר גם פונקציות אבסטרקטיות (אשר יהיו פונקציות ללא מימוש אשר ימומשו במחלקה אשר תירש מן המחלקה האבסטרקטית), והיא יכולה להכיל כל מה שמחלקה רגילה יכולה להכיל, אך לא ניתן ליצור אובייקטים של מחלקה אבסטרקטית.

3.

ההבדל בין מחלקה למחלקה סטטית היא שבמחלקה רגילה ניתן לכתוב פעולות בונות, להגדיר משתנים ופונקציות (סטטים ולא סטטים) וניתן לרשת ממחלקה "רגילה".  
לעומת זאת, מחלקה סטטית היא מחלקה אשר יכולה להכיל רק פעולות ומשתנים סטטים (כלומר הם "שייכים" למחלקה עצמה, ולא לאובייקטים של המחלקה). מכיוון שהשימוש במחלקה סטטית הוא סטטי בלבד, לא ניתן גם ליצור פעולות בונות (constructors) במחלקה סטטית, ולא ניתן לרשת מן מחלקות אחרות במחלקה סטטית.

4.

ניתן לגשת לmember בעל access modifier של internal מכל מקום באותו הassembly של הפרויקט (אוסף של מחלקות, ממשקים ועוד) גם אם זה לא מאותו הnamespace שבו הוגדר הmember עצמו, אך לו ניתן לגשת אל הmember מפרוייקטים אחרים (מאסמבלי אחר) באותו הsolution.