חלק 3 – תשובות לשאלות תיאורטיות

1. עקרון הכימוס הוא עקרון שמגדיר את "קבוצות הגישה"(Access Modifiers) למשתנים, שיטות ופונקציות בתוך מחלקה. בתוך עקרון הכימוס בc# יש ארבעה קבוצות גישה public, protected ,internal וprivate כאשר כל אחד מגדיר את הגישה למשתנים שיטות ופונקציות במחלקה בצורה שונה, לדוגמה private מסתיר את מה שהוא מגדיר ממחלקות חיצוניות כאשר public חושף אותם לכולם, נניח שיש לנו מחלקה של אוטו שבתוכה מוגדר קוד סודי בתור משתנה(Member) במחלקה אם נגדיר את המשתנה בתור private נוכל לגשת לאיבר זה רק בתוך המחלקה של האוטו ומחלקות אחרות כמו חניון לצורך העניין שמכילות את המחלקה של אוטו לא יוכלו לגשת למשתנה זה.

Protected מתייחס לאיברים בצורה זהה לאיברים כמו private אך מאפשר גישה לאיברים מתוך מחלקות יורשות, לעומת private שמסתיר את האיברים.

בנוסף יש מנגנון שנקרא "מחלקות חברות" שבניגוד לקבוצות הגישה שמוגדרות עדיין יכולות לאפשר גישה לאיברים אחת של השנייה

1. מחלקה רגילה היא מחלקה שניתן ליצור ממנה איברים(מופעים), כאשר כל הפעולות בתוכה מוגדרות לעומת מחלקה אבסטרקטית שבתוכה לא כל הפועלות מוגדרות ולא ניתן ליצור מופעים שלה. כלומר מחלקה אבסטרקטית מגדירה מן ממשק שכל מי שרוצה להשתמש בו יהיה חייב לממש את הפעולות שלו – משמש לעקרון הפולימורפיזם.
2. מחלקה רגילה מאפשרת ליצור מופעים שלה לעומת מחלקה סטטית שיש רק מופע אחד שלה. כלומר בעוד מחלקה רגילה נותנת לייצר מופעים שלכל אחד מהם יש זהות ייחודית למחלקה סטטית יש רק מופע אחד. כלומר כל שינוי במחלקה סטטית יעודכן בכל מקום – לדוגמה אם קיימת מחלקה סטטית שמייצגת בית ספר במידה ונשנה את מספר התלמידים מ1 ל2 בקטע קוד מסוים וננסה לגשת למחלקה ממקום אחר נוכל לראות שמספר התלמידים יתעדכן. לעומת מחלקה רגילה שבה כל מופע הוא ייחודי ואם נעדכן למכונית אחת את הצבע שלה כל שאר המופעים של המחלקה מכונית לא ישתנו.
3. Access modifier internal אומר כי כל קטע קוד בתוך הקובץ הנוכחי וקובץ באותו פרויקט יוכל לגשת לאיבר לצורך העניין אם באותו קובץ נגדיר את המחלקה מכונית והמחלקה תוכנית נוכל לגשת דרך המחלקה תוכנית לכל האיברים של המחלקה מכונית(צבע, מנוע וכו') אך אם נכתוב בקובץ בפרויקט אחר מחלקה של חניון לצורך העניין וננסה לגשת לאיבר צבע שיש בתוך המחלקה מכונית ישירות(בהנחה והוא הוגדר internal) לא נצליח