

**Part01**

**What are the benefits of using a generic sorting algorithm over a non-generic one?**

الـ generic sorting يشتغل مع أي نوع بيانات، بيخلي الكود أقل تكرار، وأكتر أمان وأسهل في الصيانة.

**How do lambda expressions improve the readability and flexibility of sorting methods?**

Lambda expressions بتخلي الكود أقصر وأسهل في القراءة، وتقدر تحدد طريقة الترتيب بسرعة بدون كتابة دوال منفصلة.

يعني بدل ما تعمل function جديدة لكل ترتيب، تقدر تكتب الشرط اللي عايزه جوه نفس السطر.

**Why is it important to use a dynamic comparer function when sorting objects of various**

**data types?**

لما ترتب كائنات بأنواع مختلفة، dynamic comparer بيسمحلك تحدد طريقة المقارنة لكل نوع بسهولة.

ده مهم عشان الكود يشتغل صح مع كل نوع بيانات ويكون مرن وقابل لإعادة الاستخدام

**How does implementing IComparable<T> in derived classes enable custom sorting?**

لما الكلاس الوريث يطبق IComparable<T>، بيقدر يحدد بنفسه قاعدة المقارنة بين الكائنات.

ده معناه إنك تقدر ترتب الكائنات بطريقة مخصصة على حسب اللي انت محتاجه، بدل الترتيب الافتراضي.

**What is the advantage of using built-in delegates like Func<T, T, TResult> in generic**

**programming**?

استخدام built-in delegates زي Func<T, T, TResult> بيسهّل تمرير دوال كوسيطات، بيخلي الكود مرن، قابل لإعادة الاستخدام، وأقصر من كتابة دوال منفصلة لكل حالة.

**How does the usage of anonymous functions differ from lambda expressions in terms of**

**readability and efficiency?**

Lambda expressions أقصر وأسهل في القراءة من anonymous functions، لكن الاتنين بيعملوا نفس الحاجة تقريبًا.

**Why is the use of generic methods beneficial when creating utility functions like Swap?**

الـ generic methods بتخليك تكتب دالة وحدة زي Swap تشتغل مع أي نوع بيانات، بدل ما تعمل نسخة لكل نوع. ده يوفر وقت ويخلي الكود أنظف وأسهل للصيانة.

**What are the challenges and benefits of implementing multi-criteria sorting logic in**

**generic methods?**

الفوائد: تقدر ترتب الأشياء على أكثر من معيار، الكود يبقى مرن وسهل إعادة الاستخدام.

التحديات: لازم تعرف ترتيب الأولويات كويس، والكود ممكن يبقى أصعب شوية لو المعايير كتير.

**Why is the default(T) keyword crucial in generic programming, and how does it handle**

**value and reference types differently?**

الـ default(T) مهم لأنه بيدي القيمة الافتراضية لأي نوع بيانات في الـ generic method أو class.

للـ value types زي int, doubleبيكون صفر أو القيمة الافتراضية للنوع.

للـ reference types زي string, objects بيكون null.

**How do constraints in generic programming ensure type safety and improve the**

**reliability of generic methods?**

الـ constraints في الـ generics بتحدد أنواع البيانات اللي ممكن تستخدمها.

ده بيخلي الكود آمن لأن الـ compiler يتأكد إن النوع صح، وكمان يحسن موثوقية الدوال لأنها هتشتغل بس مع الأنواع المناسبة.

**What are the benefits of using delegates for string transformations in a functional**

**programming style?**

الـ delegates بتخليك تغير أو تعالج النصوص بسهولة، الكود يبقى مرتب ومرن وقابل لإعادة الاستخدام.

**How does the use of delegates promote code reusability and flexibility in implementing**

**mathematical operations?**

الـ delegates بتخليك تمرر أي دالة رياضية للكود بدل ما تكتب كل عملية منفصلة.

ده بيخلي الكود مرن، أسهل في التغيير، وكمان قابل لإعادة الاستخدام مع أي عملية جديدة.

**What are the advantages of using generic delegates in transforming data structures?**

الـ generic delegates بتخليك تعالج أي نوع بيانات بسهولة، الكود يبقى مرن، أقصر، وأسهل في إعادة الاستخدام مع أي بيانات أو هياكل مختلفة.

**How does Func simplify the creation and usage of delegates in C#?**

الـ Func بيخليك تعرف delegates بسرعة من غير ما تكتب كلاس أو دالة منفصلة.

ده بيخلي الكود أقصر، أسهل للقراءة، وأكثر مرونة في الاستخدام.

**Why is Action preferred for operations that do not return values?**

الـ Action مخصص للدوال اللي مش بترجع حاجة، فبيخلي الكود أوضح وأسهل من استخدام Func اللي محتاج ترجع قيمة.

**What role do predicates play in functional programming, and how do they enhance code**

**clarity?**

الـ Predicates بتخلي الكود يقدر يختبر شرط معين لكل عنصر بسهولة.

ده بيخلي الكود أوضح، مرتب، وأسهل في الفهم بدل ما تكتب شروط طويلة ومعقدة.

**How do anonymous functions improve code modularity and customization?**

الـ anonymous functions بتخليك تكتب كود صغير مباشر في مكانه، الكود يبقى مرن وأسهل للتغيير.

**When should you prefer anonymous functions over named methods in implementing**

**mathematical operations?**

لما تكون العملية بسيطة ومش محتاج تستخدمها في أكتر من مكان، استخدم anonymous function بدل دالة مسماة، عشان الكود يبقى أقصر وأسهل.

**What makes lambda expressions an essential feature in modern C# programming?**

الـ lambda expressions مهمة لأنها بتخلي الكود أقصر، أوضح، وأسهل، وبتساعدك تمرر دوال مباشرة للـ methods زي sorting أو filtering بدون كتابة دوال منفصلة.

**How do lambda expressions enhance the expressiveness of mathematical computations in C#?**

الـ lambda expressions بتخليك تعبر عن العمليات الرياضية بشكل قصير وواضح، وتقدر تمررها بسهولة للدوال بدل كتابة دوال منفصلة لكل عملية.

**Part02**

**2- Search about these topics (Parallel Programming and Concurrency - Unit Testing and Test-Driven Development (TDD) - Asynchronous Programming with async and await)**

**1. Parallel Programming & Concurrency**

التزامن يعني تنفيذ عدة مهام في نفس الوقت بشكل متداخل لكن مش شرط تكون كلها شغالة فعلياً في نفس اللحظة.

التوازي يعني تشغيل مهام متعددة فعلياً في نفس اللحظة على أكثر من معالج.

في C# نستخدم مكتبة Task Parallel Library لتنفيذ المهام المتوازية بسهولة.

**2\. Unit Testing & TDD**

اختبار الوحدة يعني تجربة أجزاء صغيرة من الكود للتأكد من صحتها.

تطوير مدفوع بالاختبارات يعني كتابة الاختبار أولاً ثم كتابة الكود لجعل الاختبار ينجح وبعدها تحسين الكود.

أدوات C# مثل MSTest و NUnit تساعد في تنفيذ الاختبارات.

**3\. Asynchronous Programming & async & await**

البرمجة غير المتزامنة تسمح بتنفيذ عمليات قد تأخذ وقت مثل قراءة الملفات أو استدعاء APIs دون توقف باقي البرنامج.

async و await تجعل الكود غير المتزامن سهل الكتابة والفهم.

مفيد لتطبيقات الويب أو البرامج ذات واجهة المستخدم لضمان سرعة واستجابة أفضل.

**Part03 Bonus**

**2- what is Asynchronous programming**

البرمجة غير المتزامنة بتخلي البرنامج يقدر يعمل حاجات بتاخد وقت من غير ما يوقف باقي الشغل، فالتطبيق يبقى أسرع ومتجاوب أكتر.