1. C# چیست و چگونه استفاده می‌شود؟  
   # یک زبان برنامه‌نویسی مدرن و شیء‌گرا است که توسط مایکروسافت به عنوان بخشی از ابتکار .NET توسعه یافته است. این زبان برای توسعه طیف گسترده‌ای از برنامه‌ها از جمله وب، موبایل، دسکتاپ و بازی‌ها طراحی شده است. C# قدرت C++ را با سادگی Visual Basic ترکیب کرده و شامل تایپ قوی، اصول دستوری، اعلانی، تابعی، عمومی، شیء‌گرا (بر پایه کلاس) و برنامه‌نویسی مولفه‌گرا می‌شود. C# به طور معمول برای موارد زیر استفاده می‌شود:
   1. ساخت برنامه‌های ویندوز با فریم‌ورک‌هایی مثل Windows Forms و WPF.
   2. توسعه برنامه‌ها و خدمات وب با استفاده از ASP.NET.
   3. ایجاد برنامه‌های موبایل با Xamarin.
   4. توسعه بازی با استفاده از Unity.
   5. برنامه‌های ابری با استفاده از Azure.
2. ویژگی‌های اصلی C# چیست؟  
   # دارای چندین ویژگی کلیدی است که آن را به یک زبان برنامه‌نویسی قدرتمند و منعطف تبدیل می‌کند:
   1. شیء‌گرا: پشتیبانی از اصولی مانند ارث‌بری، محصورسازی و چندریختی.
   2. تایپ قوی: اعمال ایمنی نوع، کاهش خطاهای زمان اجرا.
   3. مدیریت خودکار حافظه: استفاده از جمع‌آوری زباله (garbage collection) برای مدیریت تخصیص و آزادسازی حافظه.
   4. کتابخانه استاندارد غنی: ارائه مجموعه کاملی از کتابخانه‌ها برای وظایف رایج.
   5. نمایندگان و رویدادها: پشتیبانی از توابع کلاس اول و برنامه‌نویسی مبتنی بر رویداد.
   6. جنریک‌ها: امکان تعریف ساختارهای داده‌ای با ایمنی نوع.
   7. LINQ (زبان یکپارچه پرس و جو): امکان پرس و جوی داده با استفاده از نحو مشابه SQL.
   8. برنامه‌نویسی ناهمگام: پشتیبانی از کلمات کلیدی async و await برای برنامه‌نویسی ناهمگام.
   9. مدیریت استثناها: ارائه مدیریت خطاهای ساختاریافته با استفاده از بلوک‌های try، catch و finally.
3. تفاوت class و object در چیست؟  
   object ها یک نوع در C# به شمار می روند که در واقع نوع پایه محسوب می شود. یعنی تمام کلاس ها و type های مختلف در C# قابلیت cast شدن به نوع object را دارا هستند. در نتیجه، هر کلاسی قابل تبدیل شدن به object را دارد.  
   class ها از سوی دیگر مانند یک قالب یا شکل هستند که میتوان با آنها type های مختلف را در برنامه ایجاد کرد. با class ها می توان type های جدید تعریف کرده و خصوصیت ها و کاربرد های مختلف به آنها نسبت داد..
4. نحوه تبدیل کد Csharp به کد ماشین چگونه است؟  
   در .NET، لایه های مختلفی وجود دارد که کد Csharp را به کد ماشین تبدیل می کنند:
5. تفاوت بین early binding و late binding چیست؟  
   در early binding، کامپایلر در زمان کامپایل (compile time) میتواند تشخیص دهد که object مورد استفاده از چه نوعی بوده و چه method ها و property هایی دارد. مزیت این مدل binding در سرعت بالاتر برنامه و کاهش ارور ها در runtime است.  
   در late binding اما نوع object در زمان اجرا (runtime) مشخص می شود و تا زمان اجرا، نوع آن object مشخص نیست. این روش گرچه انعطاف پذیری برنامه را بالاتر می برد اما در عین حال سرعت اجرا را کمتر و امکان بروز ارور در کد آن هم در زمان اجرا را بالا می برد.
6. تفاوت override و overload چیست؟  
   در overload، امضا یا signature یک متد عوض می شود مانند input parameter ها. اما در override، نحوه پیاده سازی متد تغییر می کند. مثلا نحوه انجام Read() تغییر می کند (خواندن از console، فایل و...)