**Dotnet core**

1.Difference between dotnet framework vs dotnet core  
**2.what is middleware ? how request handle in middleware ? how to register? when to use ? how it is different from filters ?**

* Middleware operate on the level of ASP.NET Core and can act on every single request that comes in to the application.
* if you don't require the context of MVC (let's say you're concerned about flow and execution, like responding to headers some pre-routing mechanism, etc.) then use middlewares. But if you require the context of MVC and you want to operate against actions then use filters.
* The execution of middleware occurs before the MVC context becomes available in the pipeline. That is, middleware does not have access to the ActionExecutingContext or the ActionExecutedContext in the case of an ActionFilter for example. What you do have access to is the HttpContext, which will allow you to perform actions on the request as well as the response. Since model binding hasn’t occurred yet, using middleware would not be suited to running a validation function or modifying values. Middleware will also run on every request regardless of which controller or action is called.
* On the other hand, filters will only run on specified actions and controllers unless you register the filter globally in the startup. Since you have full access to the context you can also access the controller and action itself

**What are the similarities:**

* Incoming requests pass through a middleware component and outgoing responses to those requests are passing again through the same middleware when going to the client. Some filters (resource, action, and result) are also two-way, others (authorization and exception) run only once for a request, and page filters (Razor pages specific) run three times.
* Middlewares can short-circuit a request by directly returning a response, instead of executing the entire middleware pipeline and not passing the request down to later middlewares. Filters can also short-circuit the filter pipeline by directly returning a response.
* Middlewares are used for cross-cutting concerns (eg: logging, performance profiling or exception handling). Filters are also used for dealing with cross-cutting concerns

**What are the differences:**

* A middleware can run for all requests while filters will only run for requests that reach the EndpointMiddleware and execute an action from an API Controller or a Razor Page.
* Filters have access to MVC components (eg: ModelState or IActionResults). Middleware works at a lower level compared to filters and is independent of MVC and Razor Pages, so it can't use any of those related components.
* Filters are designed to be applied to a subset of requests, not to all of them. For instance, we can apply a filter on a single controller or a single Razor Page. In contrast, middlewares don't have this kind of design.

References  
<https://stackoverflow.com/questions/42582758/asp-net-core-middleware-vs-filters>

**3.What is difference between app.run vs app.use ?**

* App.run add terminal middleware means end request pipeline
* App.use will pass request to next middleware

4.security of application what method and tools used for security like jwt?  
5.oops concept  
6.what is Dependency injection ? type of lifespan  
what is garbage collecter ? when it execute  
filters in dotnet core

angular how to pass data from parent to child or child to parent?  
event emitter  
difference between subject behaviour vs subject  
intercepter  
observable  
how to consume webapi in angular  
route guard

#sql

Transaction

stored procedure vs function

join types and detail

cluster index vs non cluster index

scope of CTE foreign key