**Pipe and Filter Architectural style.**

1. Pipe and filter
2. Introduction: The pipe and filter is an Architectural style which is made up of a stream of a data. It is where the data flows in a stream through pipes and filters.
3. Structure and functioning: In this structure the components of the software are known as the Filters and the connectors in the software are known as pipes. Each filter or component is supposed to apply a function or make changes to a given data or filter it and make something out of it. And the connectors also known as pipes are supposed to connect those outputs to new pipes and filters where they will be filtered again and new changes will be made.
4. Application: This architectural style is best for large projects where it can be broken down into simple and multiple steps. By doing this we are making sure that each filter is responsible for one change made during the process. They can all run simultaneously to produce changes required to reach the final outcome from a software. This kind of style is used for making UNIX programs where the output of one program can be linked to the input of another program. They are also used in compilers where the consecutive filters perform lexical analysis, parsing, semantic analysis and code generation to pass on to the pipe.
5. Advantages: Any changes can be made very easily to the software as we can always introduce new filters and steps. We can always create separate filters for new steps and can grow the pipe according to our convenience. It is also very useful in reusing certain filters for different pipes or connectors as different filters can work at the same time. Filters also do not share their state and what they are doing and only communicate through input/output channels or connectors. Which makes it very easy to make changes to any connector and new filters whenever needed.
6. Disadvantages: If a filter needs to wait until it has received all data (sort filter), its data buffer may overflow. Another disadvantage to this approach is that if the pipe allows only for one kind of data (a byte or character) the filters will have to do some parsing. This will complicate things and slow down the whole process. If you create different datatypes then you cannot link any pipe to any filter. Another disadvantage could be the interaction between different pipes may not be very efficient as in this style filters only communicate by input/output functions.
7. The most special thing about this software is that you can add new features to it by making new filters and then connecting them wherever they fit in best and are convenient to the developer.
8. References: <http://www.dossier-andreas.net/software_architecture/pipe_and_filter.html>

<https://www.student.cs.uwaterloo.ca/~cs446/1171/Arch_Design_Activity/PipeFilter.pdf>