We chose Pipes-and-Filters as the best software architecture to be described by the Object-Oriented design paradigm we chose that best fits our prototype. In Object-Oriented design it consists of classes which contain methods that are relevant to the class. How it relates to the Pipes-and-Filters Architecture is that Object-Oriented design takes in data and then spits out an output using the class's methods. Pipes-and-Filters works by taking in an input data and then transforms it through a bunch of computational components and returns an output data. Then that output data could be sent as an input data for the next. Both Object-Oriented design and Pipes-and-Filters have a similar style which is why we chose Pipes-and-Filters to be described by Object-Oriented design.

To explain further using the context of Object-Oriented design let us use the prototype we are making. Our prototype is a Blackjack game so we have two classes: A blackjack class and an executive class. In executive we take in an input from the player and put it into a blackjack class method. That class method turns into an output that would be put on display by the executive class. So in summary, the player's input gets changed by blackjack, and the output of it goes into a display method in executive and outputs a view for the player. To explain Pipes-and-Filters the input the player does is the input data of the Pipes-and-Filters. The transformation through a bunch of computational components(filters) is the class method of the blackjack class. The output of that can be used as an input of another Pipes-and-Filters. This can be shown when the blackjack class's output becomes an input for the executive class method.

Therefore, this is how Pipes-and-Filters Architecture works in the context of Object-Oriented design paradigm.