Pipe And Filter is an [architectural pattern](https://en.wikipedia.org/wiki/Architectural_pattern) very simple, yet powerful architecture, that is also very robust. It consists of any number of components (**filters**) that transform or **filter** data, before passing it on via connectors (**pipes**) to other components.. This is done if you have a lot of transformations to perform and need to be very flexible in using them, yet you want them to be robust. The pipe and filter design pattern decouples these major components allowing for efficient [code reuse](https://en.wikipedia.org/wiki/Code_reuse) and parallel development. Traditionally used for desktop user this architecture has become popular for designing  [applications](https://en.wikipedia.org/wiki/Web_application) and even mobile, desktop and other clients. Popular programming languages like [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)), [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)), [PHP](https://en.wikipedia.org/wiki/PHP) and others have popular pipe and filter frameworks that are currently being used in C# application development straight [out of the box](https://en.wikipedia.org/wiki/Out_of_the_box_(feature)).

This simple app generally is a currency converter which converts birr to dollar and vice versa based on the selected choice of the user . This application is done using this architecture that is discussed above and it is implemented based on the following class diagram as shown below.

|  |
| --- |
| Hello Pipeline line |
| User Value |
| Birr to dollar  Dollar to birr  Get calculated value |

|  |
| --- |
| Ipipe line element |
|  |
| Get first name  Get last name  Set phone number |

|  |
| --- |
| Pipe line |
| The hello pipeline line  The ipipe line element |
| Pipe stream ( ) |

|  |
| --- |
| Program |
| Name  Number |
| Action performed |

|  |
| --- |
| Pipe stream |
| Name  Number |
| Action performed |