**Bugs Fixes and Improvements**

* Categories fetching was not pointing to correct endpoint, updated the same to jokes/categories
* Added exception handling to all functions in JsonFeed.cs file making HttpClient call
* Added Async function for each API call in JsonFeed.cs
* Removed the unused parameter “result” from JsonFeed constructor.
* While fetching categories added proper json deserialization
* While fetching the Random joes the number was not considered. Updated the logic to run the loop and get all the jokes. Added all the jokes in the string array.
* Updated the name replacement to work with the value rather than whole big json string, which would improve on the performance.
* ConsolePrinter class ToString() function is returning null. Replaced this function with Print () function.
* Moved PrintResults function to ConsolePrinter class and added the default parameter for printing the result in different lines.
* Removed global variable and used local variables inside function in Program.cs
* Noticed that we were getting same categories every time we hit the API. So added the caching in the CNController class. This would help in speed up the execution after the first call

**Usability Changes Done**

* Removed the first message – “Press ? to get instructions.” This is extra step which user must perform.
* Changed the first option and gave user the option to get a single joke.
* User was asked to get Categories, which does not make sense as after getting categories, only list of categories is shown. Instead, at time random jokes if user selects to mention category then we can fetch this list as this would also help in performance
* Added third option for user to exit

**Validations**

* While accepting input from user for number of jokes, the input is validated for correct number in range
* While accepting the category, the input is validated with list of categories.

**Code Structure**

Changed the code architecture to have separation. Decoupled the code into different classes each having specialized responsibilities.

Following is the description for different classes

* Program.cs – Handling user inputs.
* ConsolePrinter.cs – This class has responsibility to print on the console.
* CNController.cs – This controller class is responsible to handle the logic of fetching and returning the data
* JsonFeed.cs – This class is handling the API calls.

**Future Maintenance & Extension**

* Writing the Unit test cases
* Dependency Injection can be introduced which could help in mocking at time of unit testing