University of Engineering and Technology

***Document Management System***

***OOP Project***

***Submitted By:***

**Names: Roll Numbers:**

Hasan Ahmed (Documentation + UML) 2019-CS-145

Usman Ghani (Code) 2019-CS-121

Nimra Zahoor (Code) 2019-CS-112

Abdul Samad (UML) 2019-CS-132

Ifrah Ahmad (Documentation) 2019-CS-116

***Submitted To:***

Sir Usman Asghar

Synopsis:

In this project i.e. Document Management System we have created a system in which we can do the following things:

1. A person can add his details like:

* Name
* Course ID
* Course Name etc

1. A person can do the following things with his data:

* Add
* Edit
* Delete
* Save
* Load
* Print

Basis of code:

This Document Management System is made up of following classes:

Base Class:

* Document

Derived Classes:

* Category
* Topics
* Tags

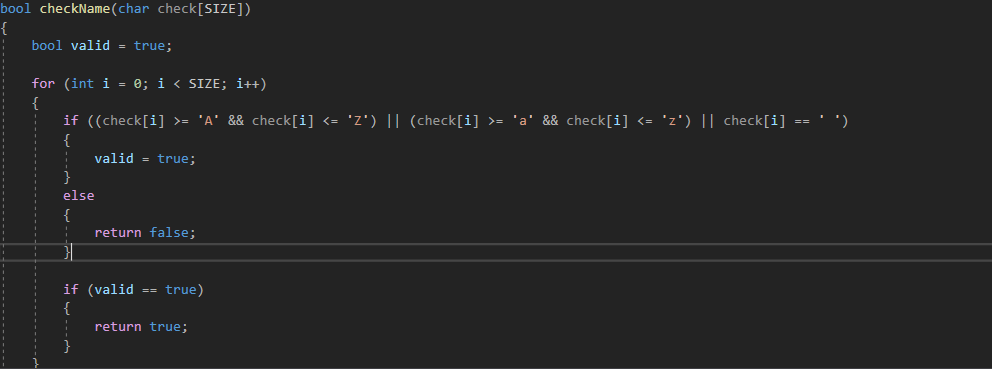
Each class consists of the functions for adding/editing/deleting/saving/loading and printing the details. There are validity functions to check if the inputs of ID or name etc are valid or not.

1. **Details of code:**

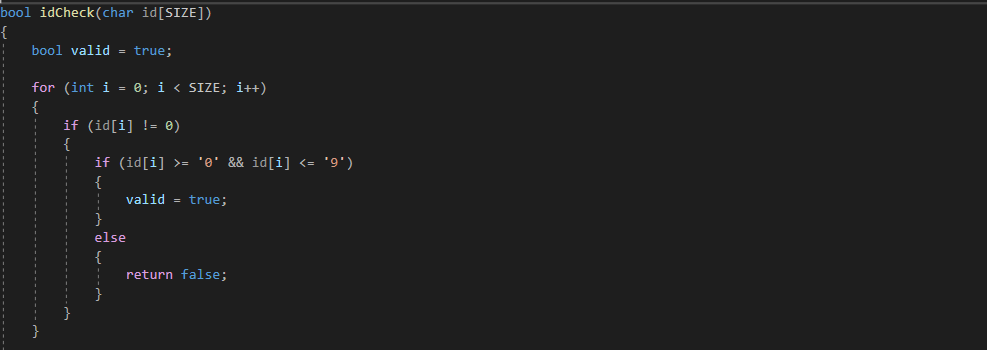


This is the start of program in which there are header files that we use for making our code. The header files <cstring> is used so that we can use string and the header file <fstream> is used for file handling. A global variable SIZE is declared as int and is given the value of 100 so that it can be use anywhere in the code. The pragma warning is used to remove the warning of strcpy in Visual Studio.

* Validity Functions:



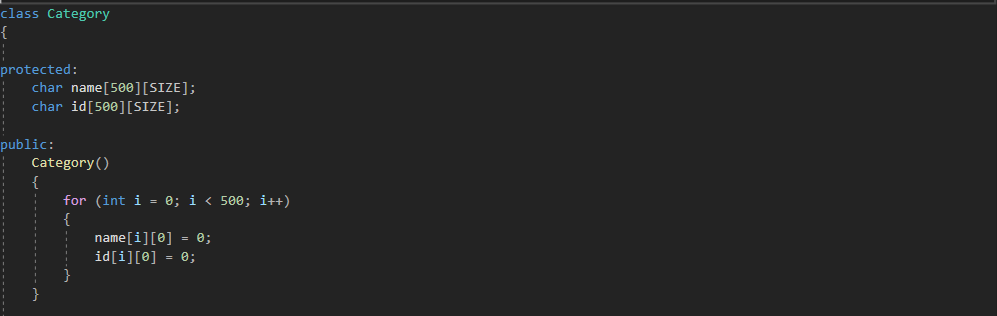
Here a bool function is written to check the validity of names that user is going to input. A for loop is used to check each char of the input. If there is any input in which something other than alphabets (both capital and small letters) and space is entered it will give and error because the input will be invalid.

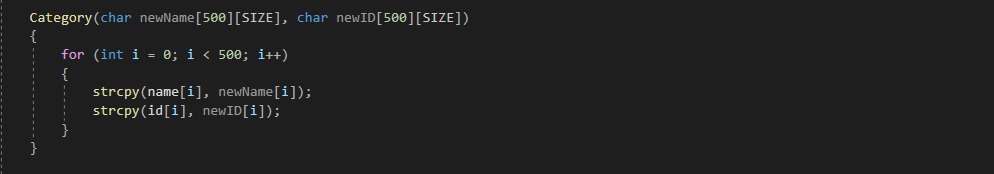




This is the second bool function in which the validity of the ID that user will input is checked. Again, here a for loop is used which check each char of the input to check if a number is given. If there is an input other than the number it will give an error because the input will be invalid.

1. Category Class:

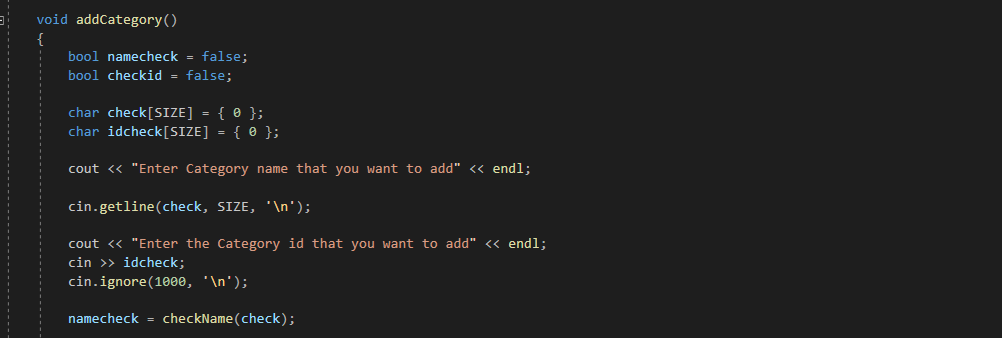


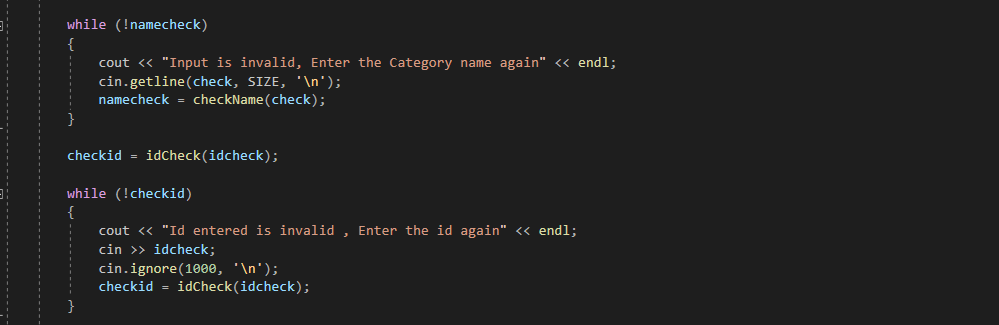


This is the class of category in which two data members i.e. name and id are declared through char and 2D arrays are used. Next a default constructor is used in which the 2D array of name and id are initialized to 0. Next a parameterized constructor is made in which two parameters i.e. newName and newID are passed and they are set equal to the original data members. Strcpy is used to copy the name and ID from parameters to the orginial data members.

* **Add Function:**

This is the add function for category that is used to add a new category. The function void is used because there is no need to return any value.





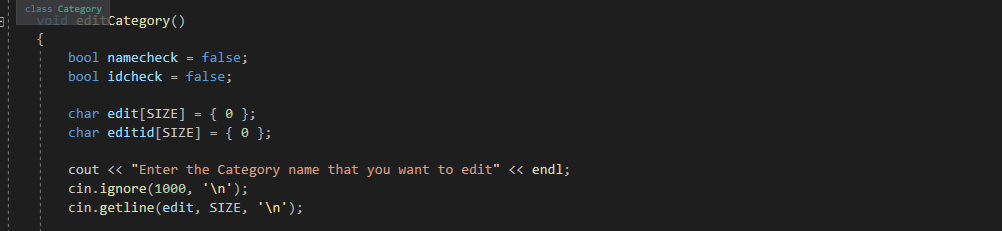
First the user is asked to enter category name and category ID and it is checked that whether the input entered by the user is valid or not. For this purpose, a while loop is used. If the input is invalid it will display “Input is invalid” and will ask the user to input the value again. Two while loops are used to check the input of name and ID separately and if the input is invalid asking the user to again input the values.



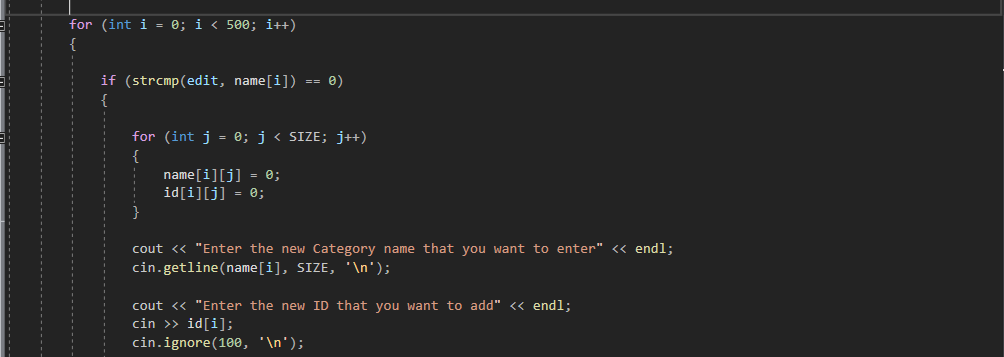
Now two for loops are used to add the names and ID to the document separately. A limit of 500 is given that the compiler can check upto 500 chars of names and IDs. In the loops strcpy is used and the name and IDs are added seperately. If the names and IDs are added successfully the keyword *break* breaks the loop.

* **Edit Function:**

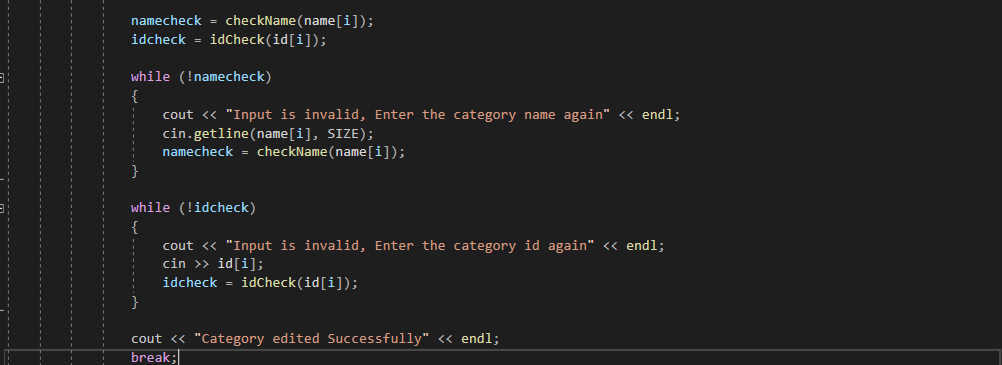
This is the edit function for category that is used to edit any category. The function void is used because there is no need to return any value.

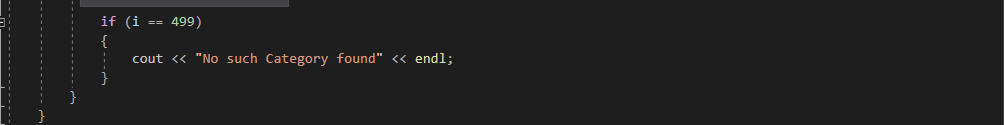


In this code the user is asked to input the category that he wants to edit.



Now a nested for loop is used to edit the category. The user is asked to enter the new name that he wants in that category and next the user is asked to enter the new ID that he wants in that category. Strmp is used to compare the strings.

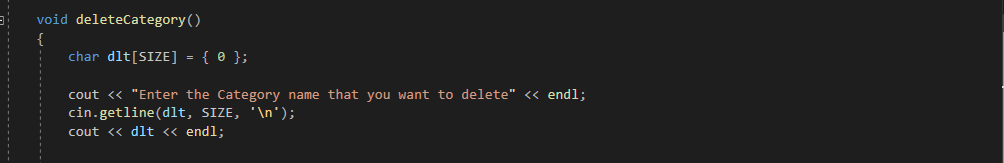




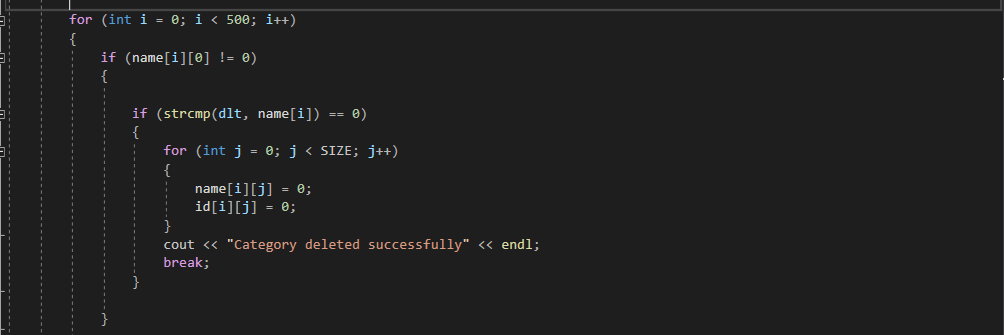
The new input of name and ID by user and checked through validity functions, to check if the inputs are valid according to the requirement. For this purpose, again two while loops are used. If the input is invalid the user is asked to enter the values again. If the name and ID are valid the following message is displayed “Category edited successfully” and again the *break* keyword is used to break the loop. Similarly, if user enters a category that does not exists a message is displayed saying “No such category found”.

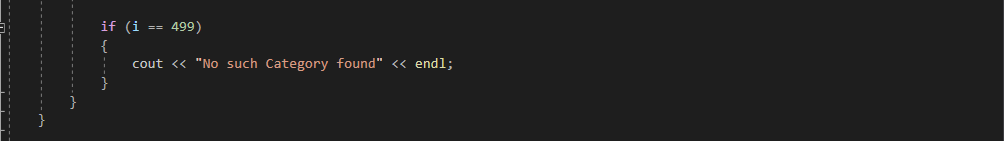
* **Delete Function:**

This is the delete function for category that is used to delete any category. The function void is used because there is no need to return any value.



In this code the user is asked to input the category that he/she wants to delete and that category is displayed.

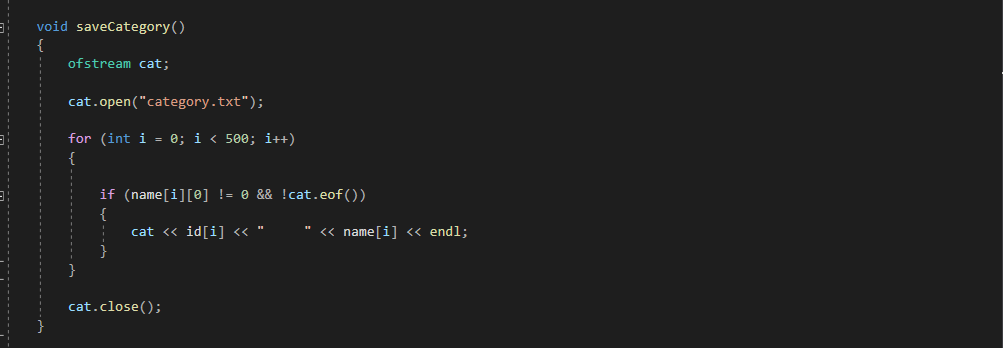




Here, again a nested for loop and nested if is used to delete the category that user inputs. Once that category is deleted it displays “Category deleted successfully” and again the keyword *break* is used to break the loop. If the user inputs and invalid category it is displayed “No such category found”.

* **Save Function:**

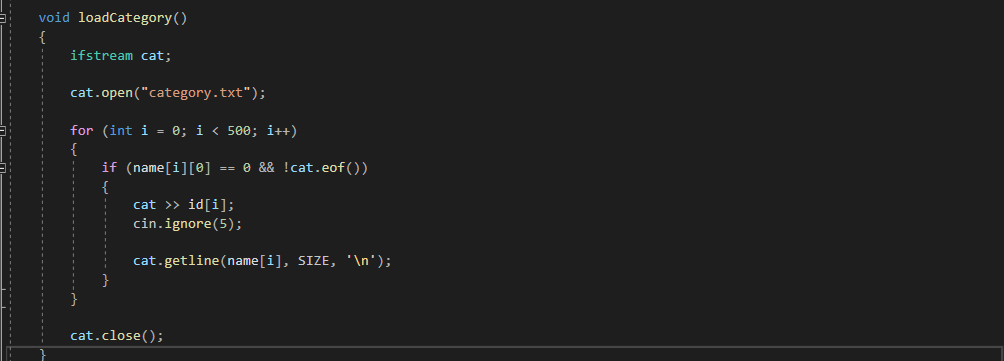
This is the save function for category that is used to save any category using file handling. The function void is used because there is no need to return any value.



In this code file handling is used to save any category. For this purpose, a for loop is used. The keyword ofstream is used to save the data entered in a txt file as written in the code (“category.txt”).

* **Load Function:**

This is the load function for category that is used to load any category using file handling. The function void is used because there is no need to return any value.

****

This is the function used to retrieve the data saved in the system. It is done by file handling and ifstream is used to retrieve the data from txt file. Again, a for loop is used and the data is retrieved from the txt file present in the system.

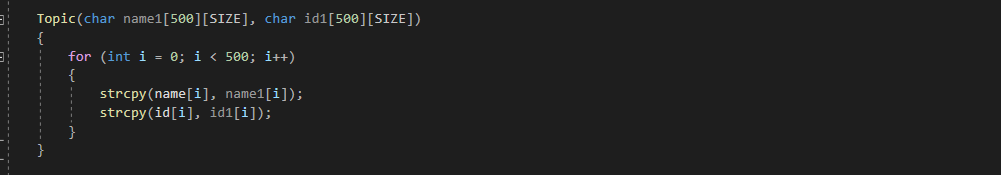
* **Getter Function:**



This function is used to get the category.

1. Topic Class:

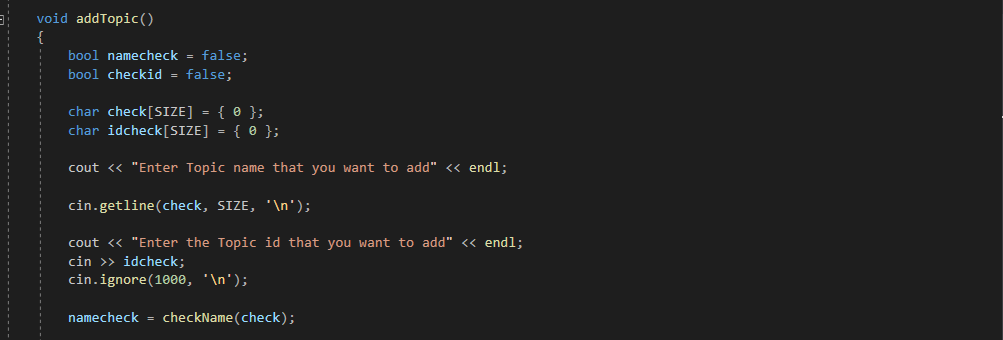


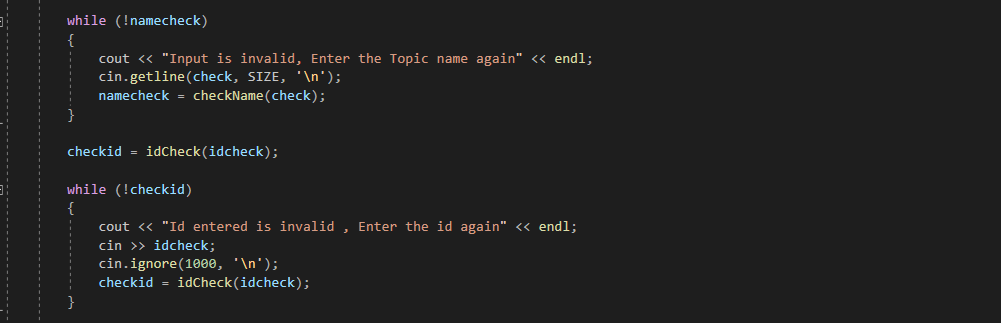


Just like category class two constructors (i.e. default and parameterized) are made and the values are passed.

* **Add Function:**

This is the add function for category that is used to add a new topic.

****

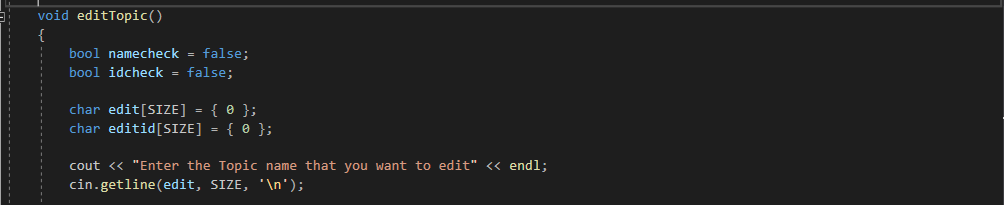
****

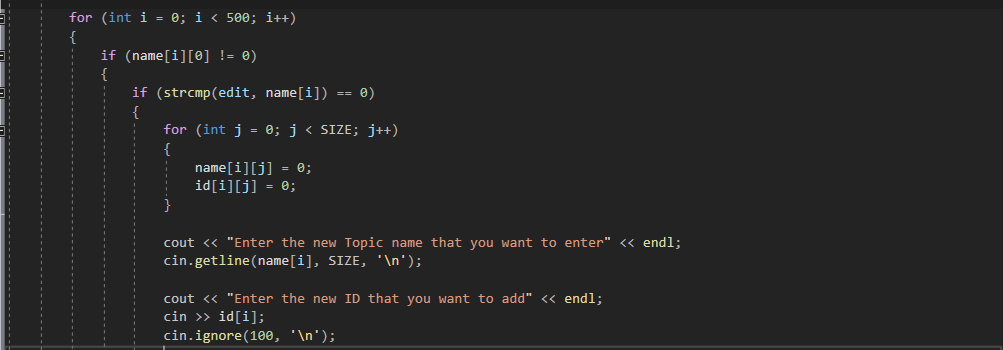
****

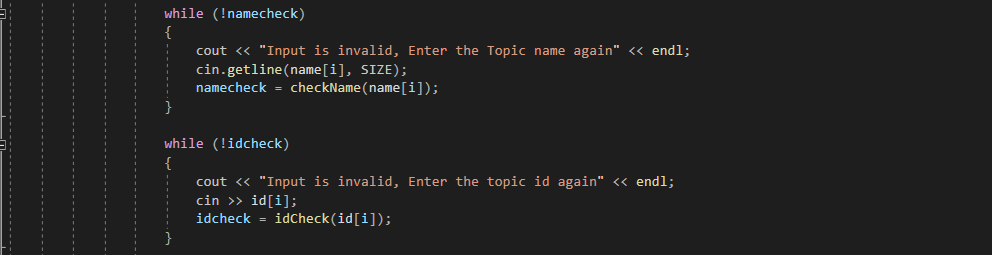
This function is similar to the add function of category except that in it user will be able to add a topic instead of a category.

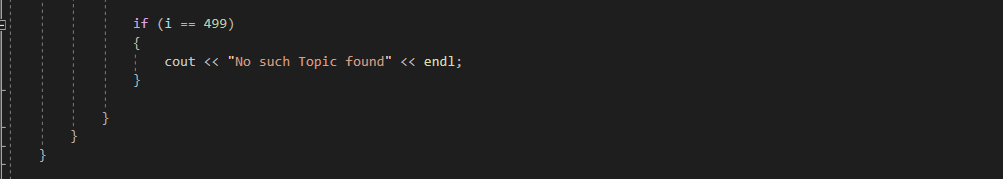
* **Edit Function:**

This is the edit function for topic that is used to edit any topic.





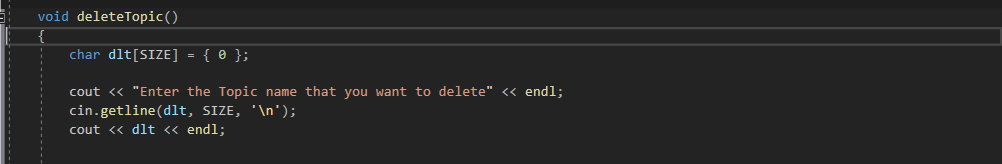


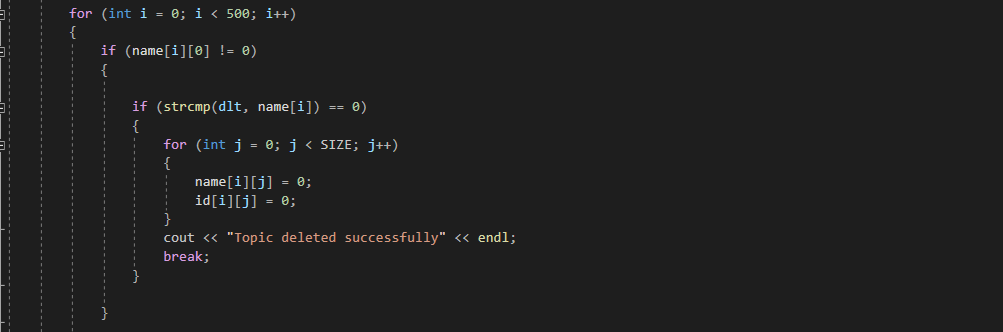


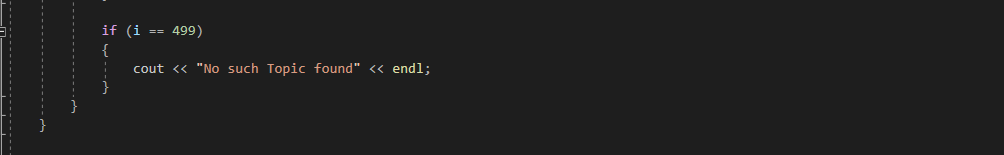
This function is similar to the edit function of category except that in it user will be able to edit a topic instead of a category.

* **Delete Function:**

This is the delete function for topic that is used to delete any topic.



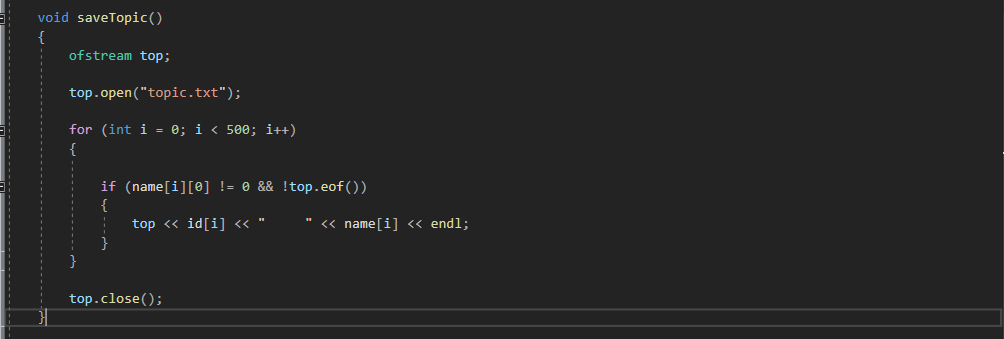




This function is similar to the delete function of category except that in it user will be able to delete a topic instead of a category.

* **Save Function:**

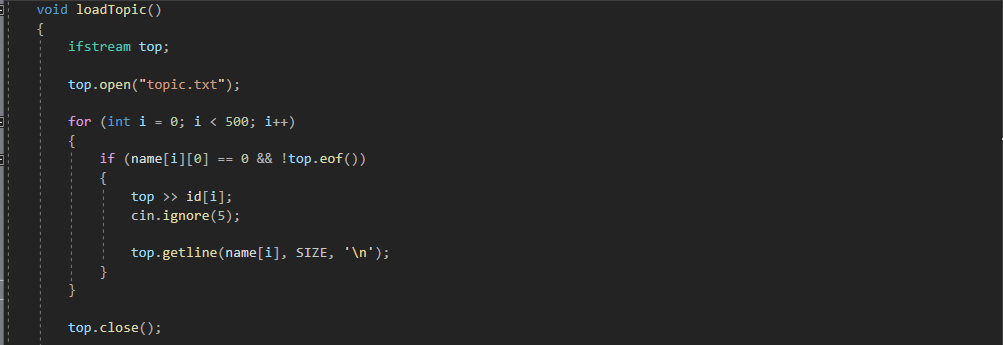
This is the save function for topic that is used to save any category using file handling.



This function is similar to the save function of category except that in it user will be able to save a topic instead of a category.

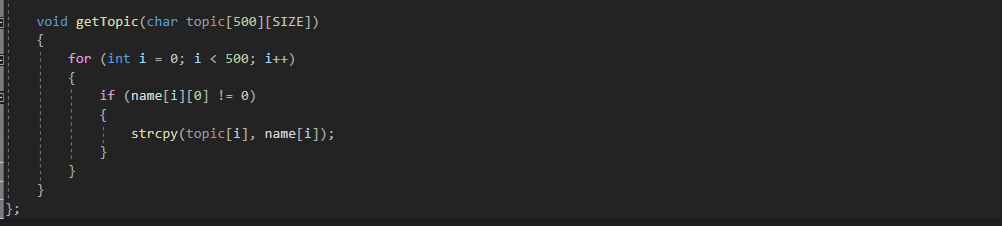
* **Load Function:**

This is the load function for topic that is used to load any category using file handling.



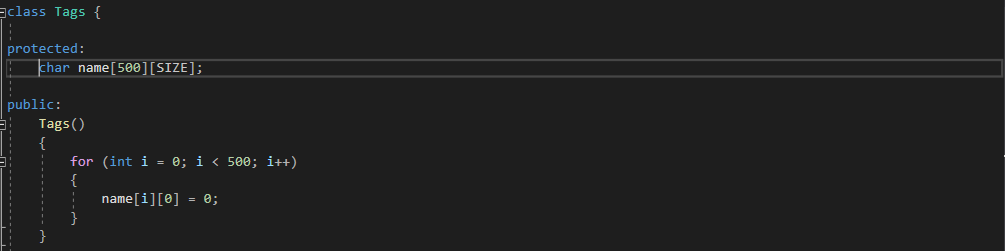
This function is similar to the load function of category except that in it user will be able to retrieve the topic instead of a category.

* **Getter Function:**

****

This function is used to get the topic.

1. Tag Class:

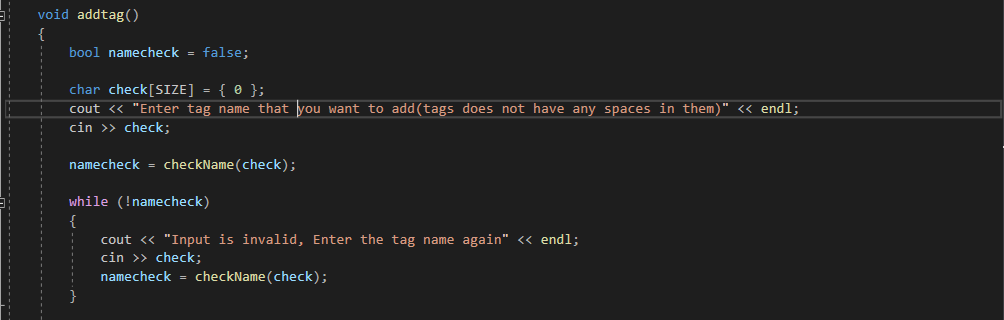




Just like category class two constructors (i.e. default and parameterized) are made and the values are passed.

* **Add Function:**

This is the add function for category that is used to add a new tag.

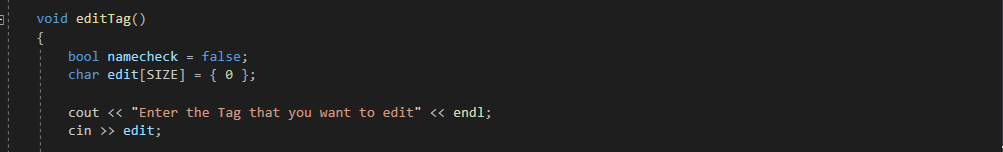


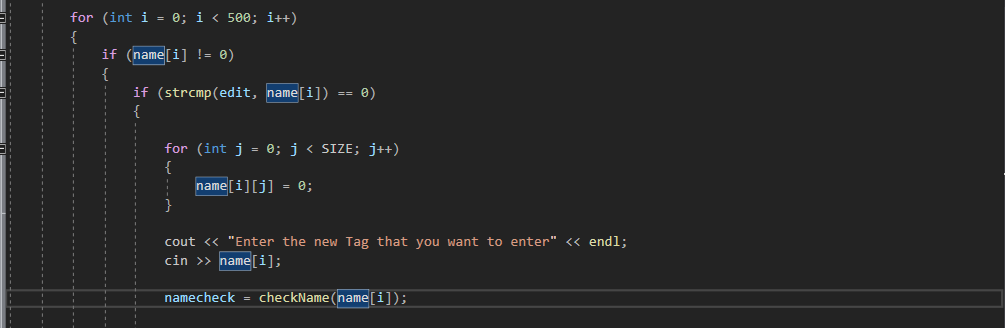
****

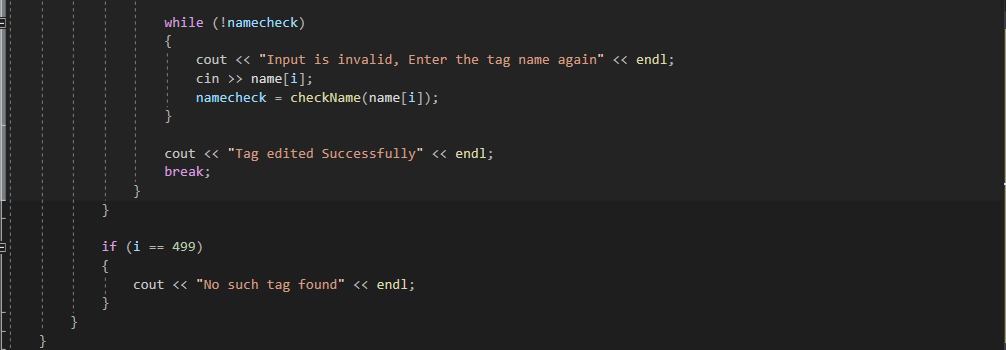
This function is similar to the add function of category except that in it user will be able to add a tag instead of a category and only validation of name will be used because no number is needed in tags.

* **Edit Function:**

This is the edit function for tag that is used to edit any tag.



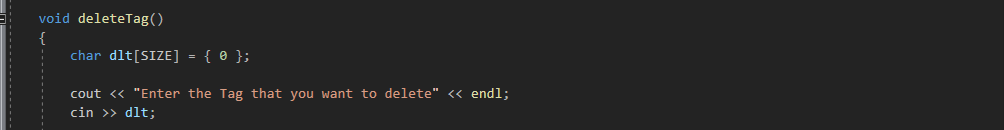


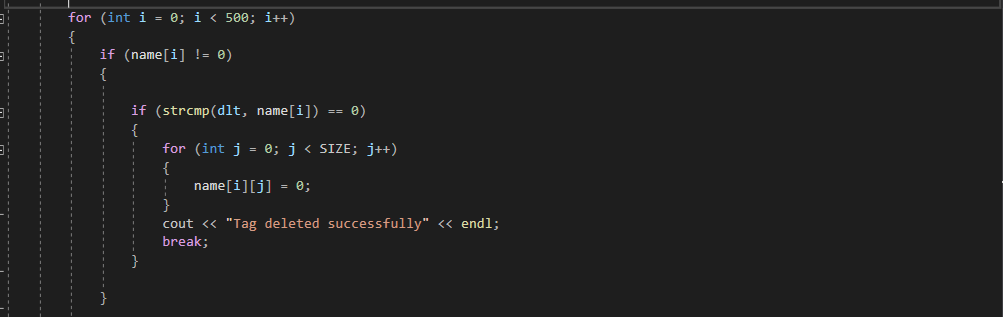


This function is similar to the edit function of category except that in it user will be able to edit a tag instead of a category and only validation of name will be used because no number is needed in tags.

* **Delete Function:**

This is the delete function for tag that is used to delete any tag.



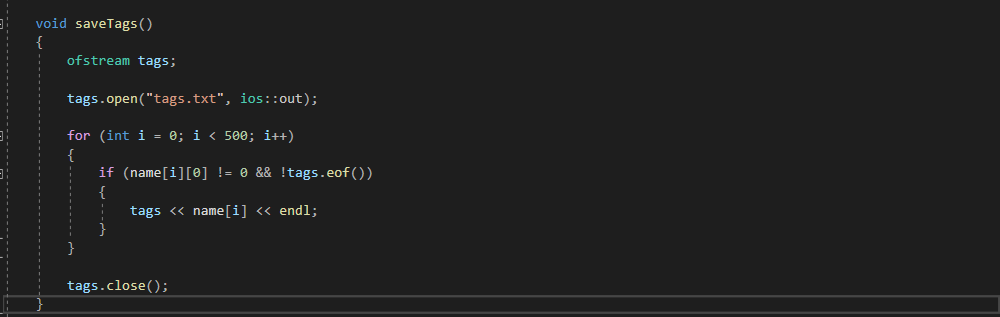




This function is similar to the delete function of category except that in it user will be able to delete a tag instead of a category and only validation of name will be used because no number is needed in tags.

* **Save Function:**

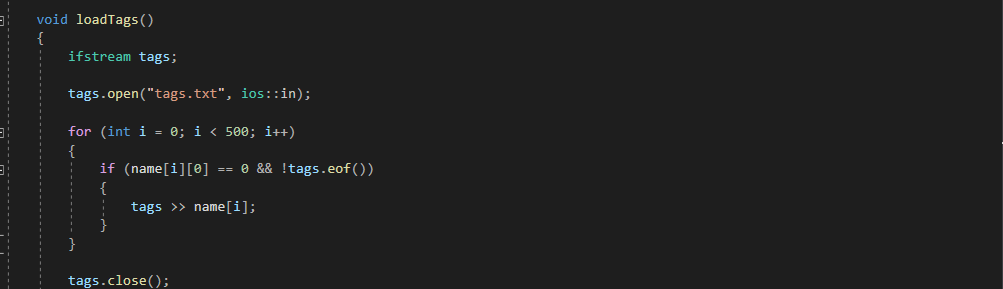
This is the save function for tag that is used to save any category using file handling.



This function is similar to the save function of category except that in it user will be able to save a tag instead of a category and only validation of name will be used because no number is needed in tags.

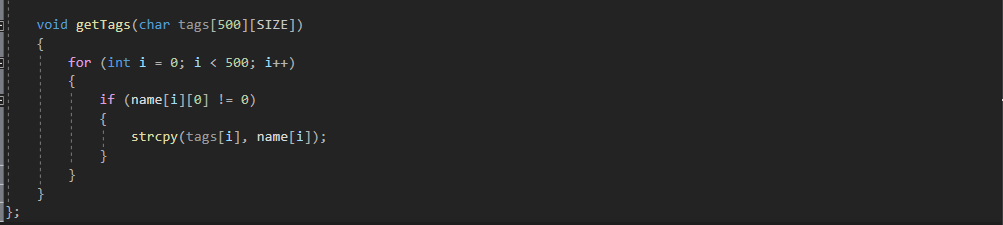
* **Load Function:**

This is the load function for tag that is used to load any category using file handling.



This function is similar to the load function of category except that in it user will be able to retrieve the tag instead of a category and only validation of name will be used because no number is needed in tags.

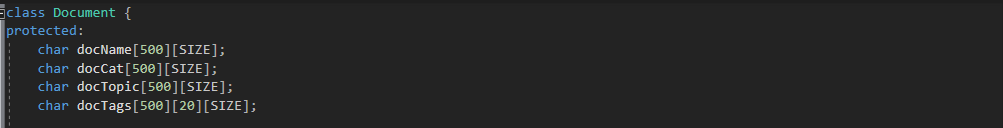
* **Getter Function:**

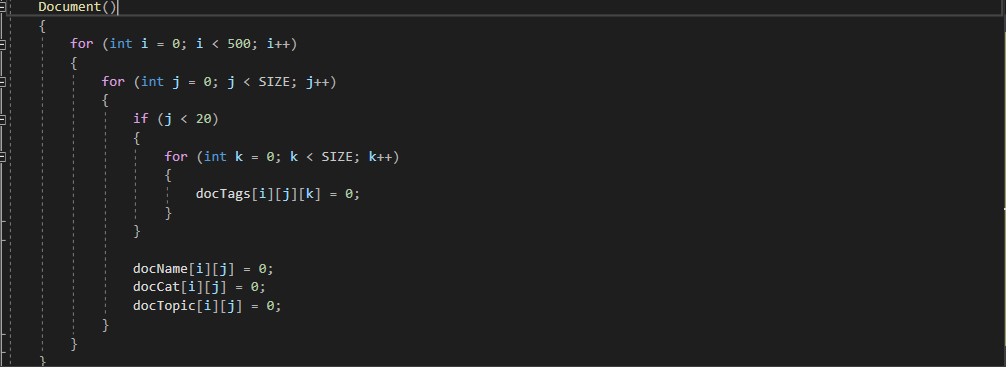
****

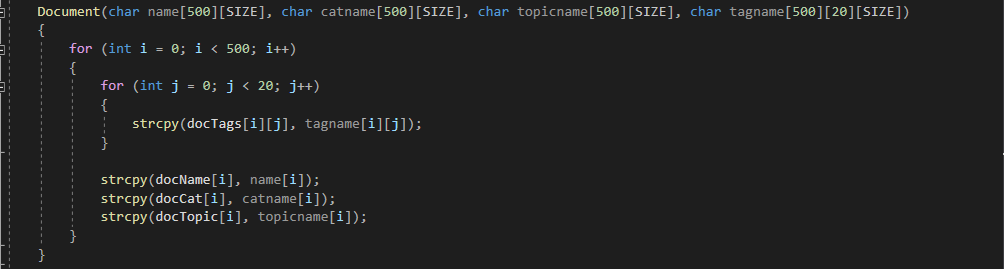
This function is used to get the tag.

1. Document Class:

This is base class and inherits all the other classes.



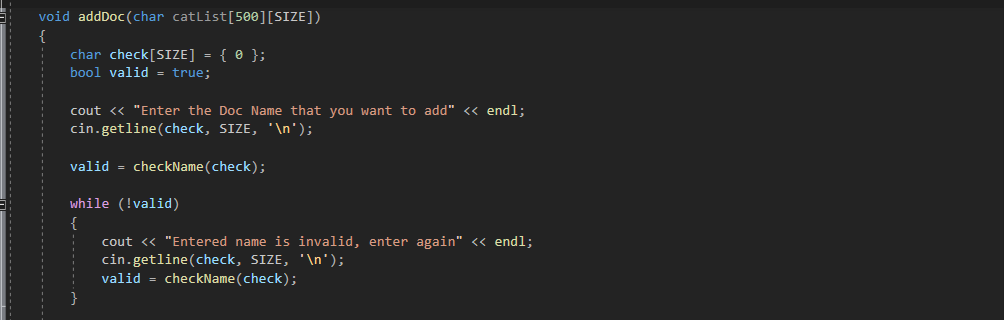


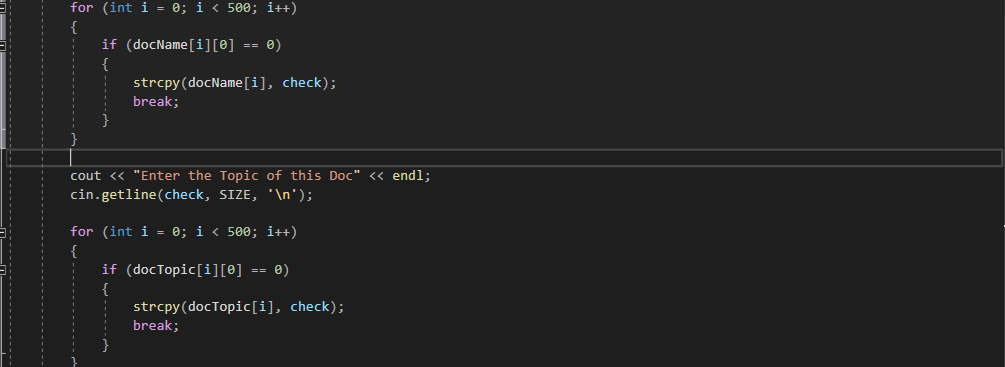


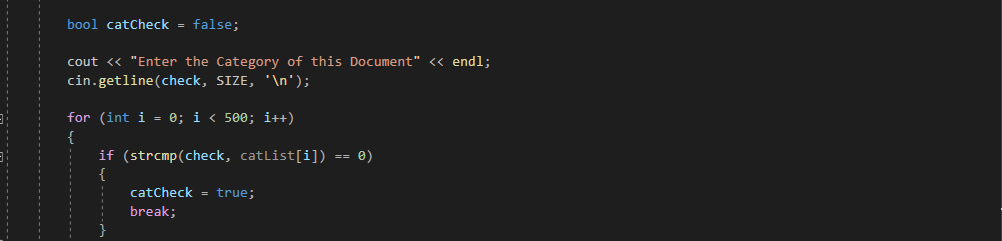
Just like category class two constructors (i.e. default and parameterized) are made and the values are passed.

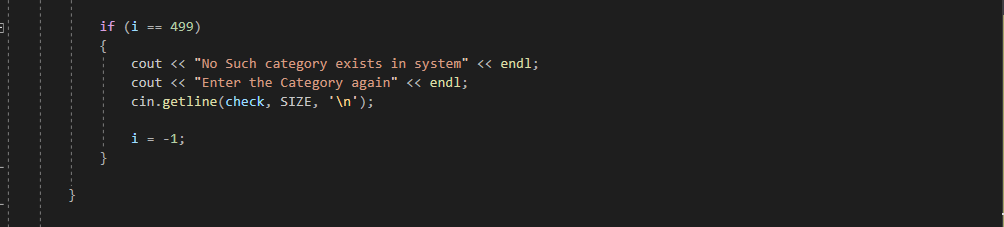
* **Add Function:**

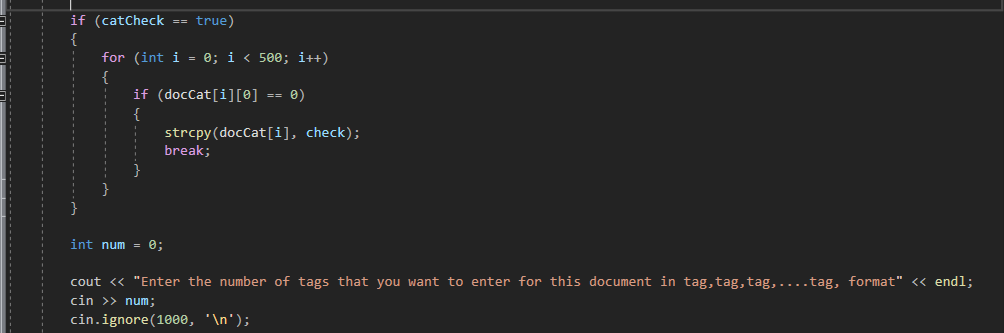
This is the add function for document that is used to add a new document.

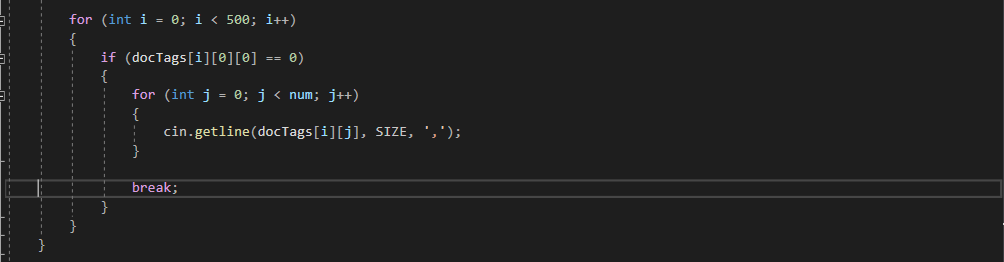


****

****

****

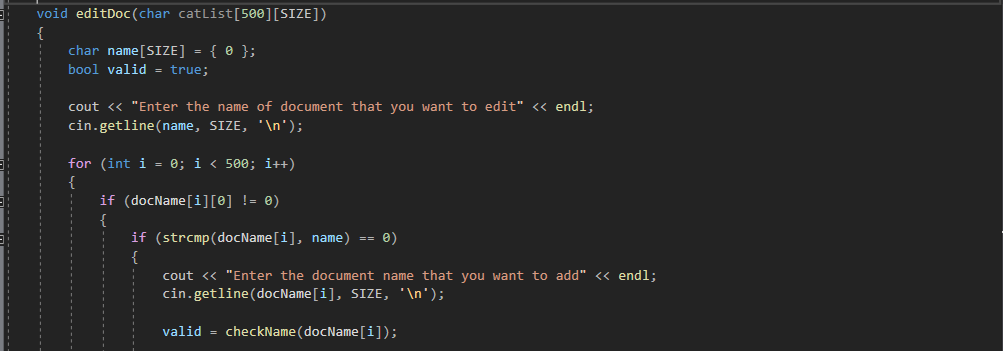
****

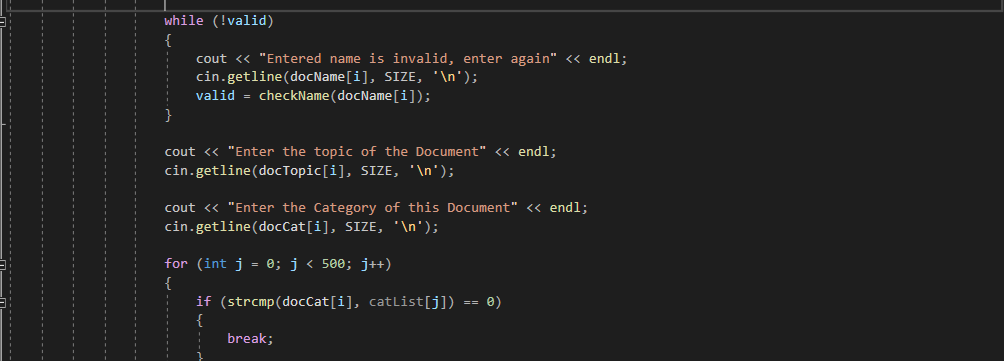
****

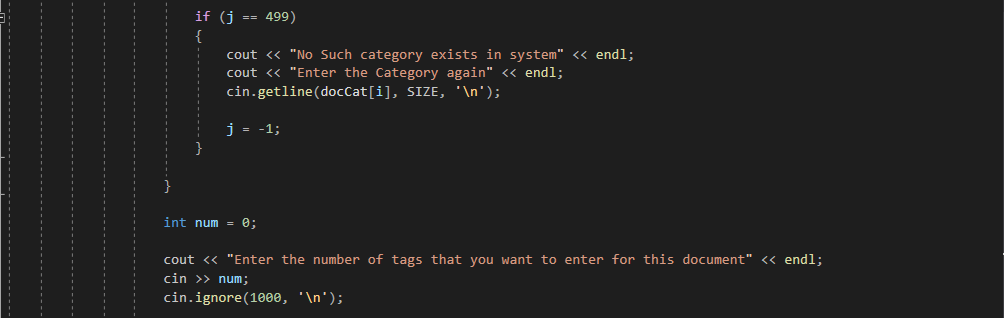
This function is similar to the add function of category except that in it user will be able to add a document which will include all the parameters i.e. category, topic, tags.

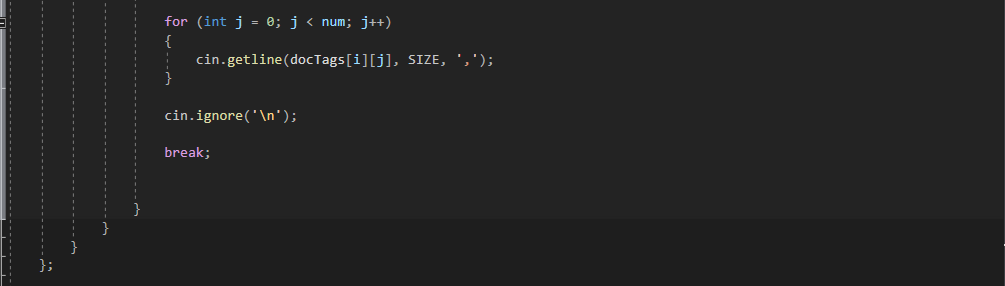
* **Edit Function:**

This is the edit function for tag that is used to edit any document.





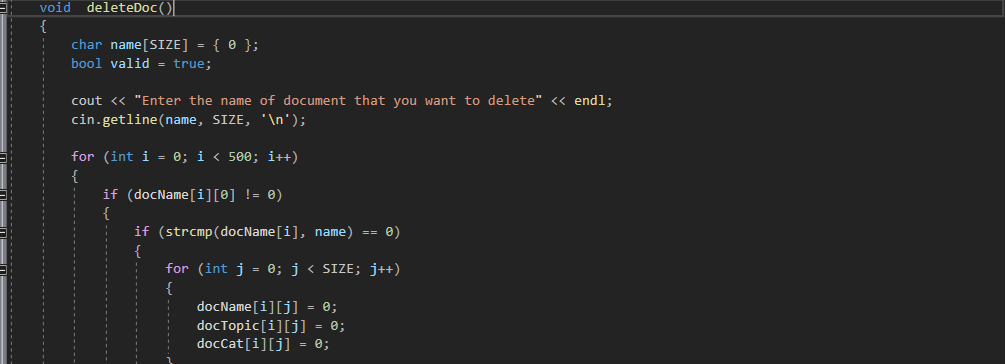


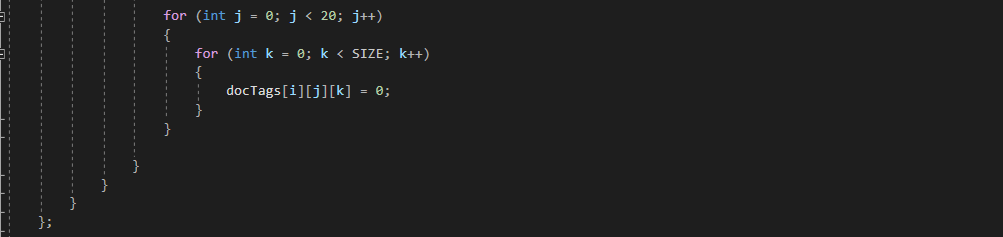


This function is similar to the edit function of category except that in it user will be able to edit a document which will include all the parameters i.e. category, topic, tags.

* **Delete Function:**

This is the delete function for tag that is used to delete any document.

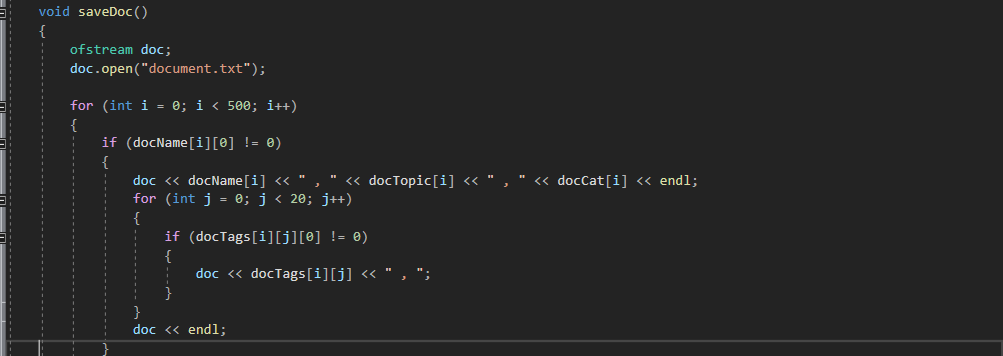




This function is similar to the delete function of category except that in it user will be able to delete a document which will include all the parameters i.e. category, topic, tags.

* **Save Function:**

This is the save function for tag that is used to save any category using file handling.

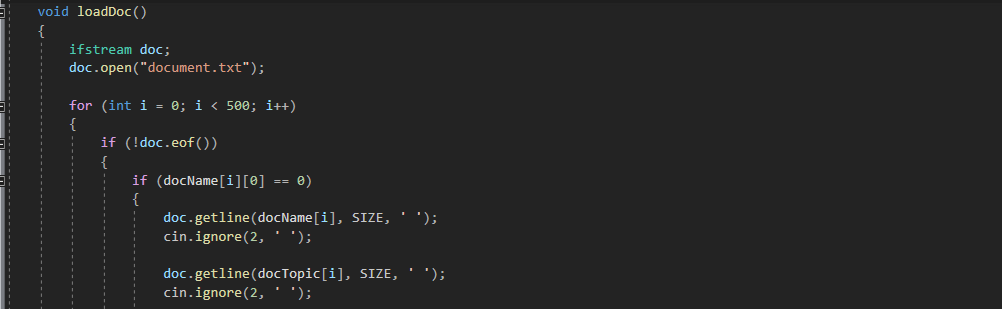


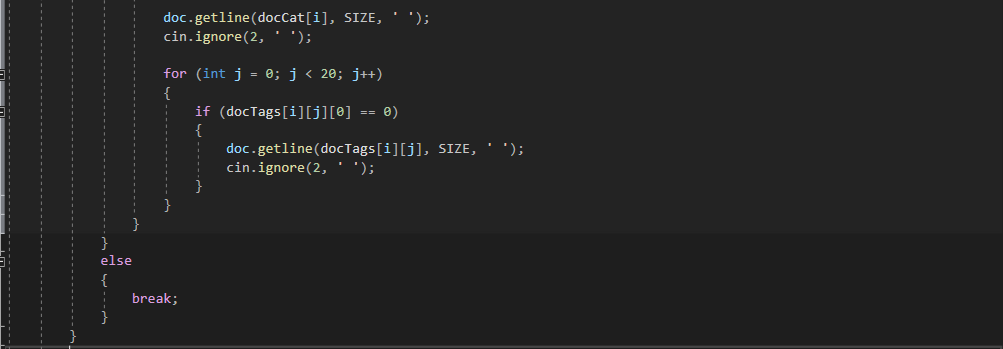


This function is similar to the save function of category except that in it user will be able to save a document which will include all the parameters i.e. category, topic, tags.

* **Load Function:**

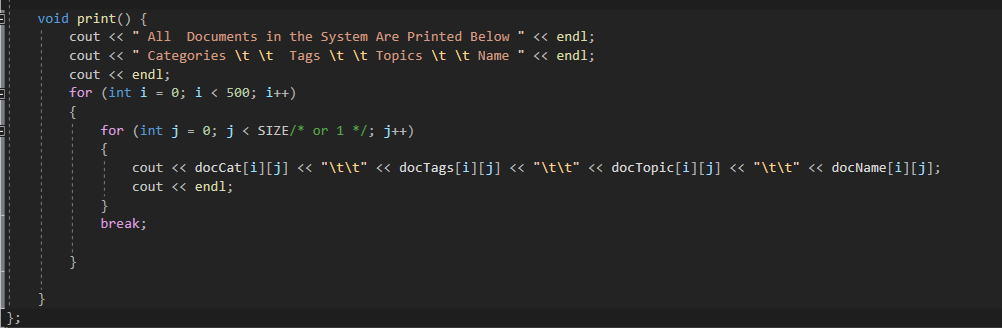
This is the load function for document that is used to load any category using file handling.







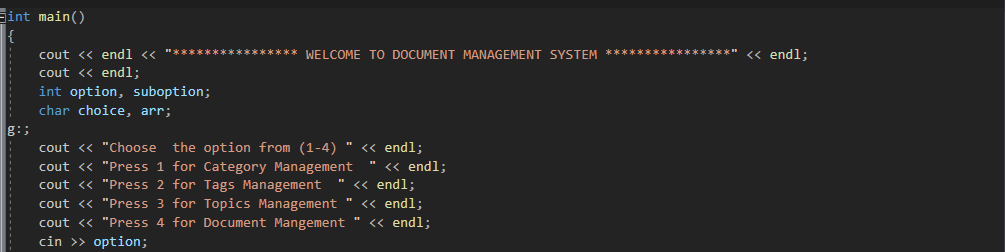
This function is similar to the load function of category except that in it user will be able to retrieve the document which will include all the parameters i.e. category, topic, tags.

* **Print Function:**

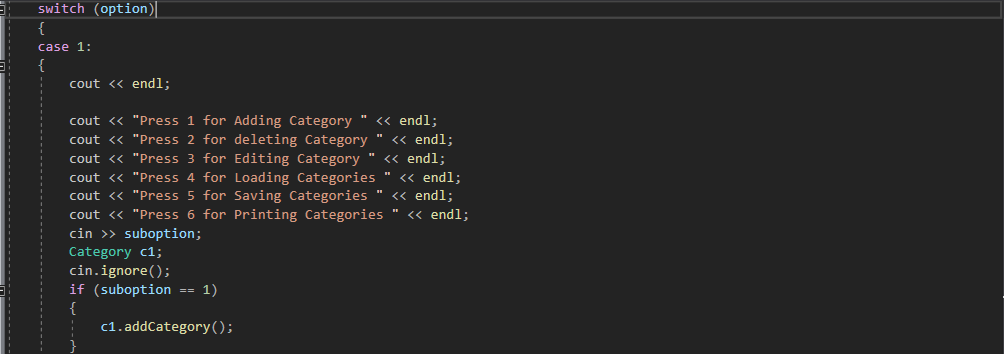
This function is for printing all the parameters i.e. categories, tags and topics according to the required format.

1. Main Function:

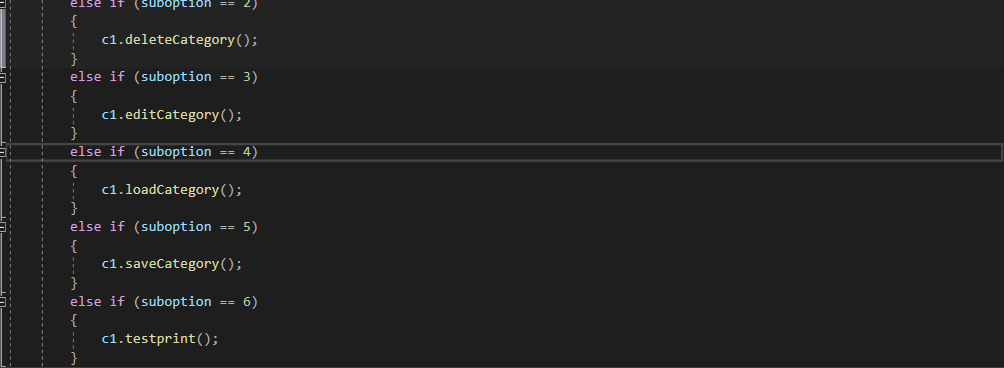
In this main function all the classes and functions are called and are printed on the console. To print all the parameters separately switch/case and if/else is used.



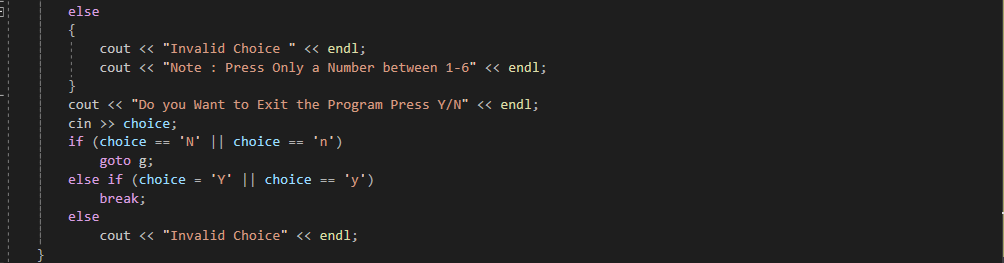
In this code the user is asked to enter the input from 1-4 for management of different parameters like category, tags, topics and document respectively.



According to this code user is asked to input from 1-5. The functions (i.e. add/edit/delete etc) will be performed according to the input.



The functions are called using objects to perform the specific function on specific input.



If there is an invalid input, a message saying “Invalid input” will be displayed and the user will be asked whether he wants to end the program or not.

Similarly, if the user enters 2 he/she will be able to perform all the functions like in category for the tags. If the user enters 3 he/she will again get the similar options like in category and user will be able to perform the functions for topic. Same is the case for inputting 4, user will be able to perform all these functions for document except that in documents all the parameters (topics, tags, category) can be printed by entering the specific option.

***End of Project***