```
import logging
import sys

logging.basicConfig(filename="String_s.log", level=logging.DEBUG,
format="%(levelname)s %(asctime)s %(message)s")


class String_s:
    logging.info("We are  accessing class String_s")

    def filter_vowel(self, S):
        """Filter out vowels from a string database"""
        logging.info("Inside filter_vowel function")
        try:
            self.S = S
            self.i = 0
            while self.i < len(S):
                logging.info("Checking %s", self.i)
                if S[self.i].lower() in "aeiou":
                    print(S[self.i], end=", ")
                    logging.info(" vowels filtered :- %s ", S[self.i])
                self.i += 1
        except Exception as e:
            logging.exception("Exception encountered :- %s", e)
            logging.exception(sys.exc_info())

    def length(self, S):
        """ Find the length of string"""
        logging.info("Inside length function")
        try:
            self.S = S
            self.c = 0
            for i in S:
                self.c += 1
            else:
                logging.info("length of string  is :- %s", self.c)
                return 'length : ' + str(self.c)
        except Exception as e:
            logging.error(e)
            return e

    c = lambda x: ''.join([i for i in x])

    def concat(self, *x):
        """ function to return lambda function from class  to concatenate
strings"""
        logging.info("Inside cancat Function")
        try:
            self.x = x
            logging.info("Concatenated string %s ", String_s.c(self.x))
            return String_s.c(self.x)
        except Exception as e:
            logging.exception(e)
            return e
```

```
S = """ Python is a high-level, interpreted, general-purpose programming
language. Its design philosophy emphasizes code readability with the use of
significant indentation.[32]

Python is dynamically-typed and garbage-collected. It supports multiple
programming paradigms, including structured (particularly procedural),
object-oriented and functional programming. It is often described as a
"batteries included" language due to its comprehensive standard library.[33][34]

Guido van Rossum began working on Python in the late 1980s as a successor to the
ABC programming language and first released it in 1991 as Python 0.9.0.[35]
Python 2.0 was released in 2000 and introduced new features such as list
comprehensions, cycle-detecting garbage collection, reference counting, and
Unicode support. Python 3.0, released in 2008, was a major revision that is not
completely backward-compatible with earlier versions. Python 2 was discontinued
with version 2.7.18 in 2020.[36]

Python consistently ranks as one of the most popular programming languagesc
 """
str_var = String_s()
# print(str_var.length(S))
# print(str_var.filter_vowel(S))
print(str_var.concat("a", "s", "dfs"))
```