```python
from random import randint

class Dice:
    def __init__(self, defaultValue): # self - represents the object to be created
        """
        constuctor that execute when a Dice object is created
        the default value for the dice is 4
        """
        self.__value = defaultValue   # __ (dunder) ==> PRIVATE attribute
        # self.value = 4   # defining attribute this way make it public attribite

    def roll(self):
        self.__value = randint(1,6)

    def getValue(self):  # accessor
        return self.__value
                                # no mutator for Dice object as it is not ethical to do so
    def __str__(self):
        return "Value of the dice is " + str(self.__value)

if __name__ == '__main__':
    #creation, d1 d2 are unique names for the Dice objects
    d1 = Dice(1)
    d2 = Dice(2)

    print(d1.__str__())
    print(d2)
    print(str(d2))

    d2.roll()
    d1.roll()
    print("d1 is", d1.getValue())
    print("d2 is", d2.getValue())
```

Different ways to call the str method of Dice object

Dice
- value: int
+ __init__( ) + roll( ) + getValue( ): int + __str__(): str

UML diagrams for Deck and Card class