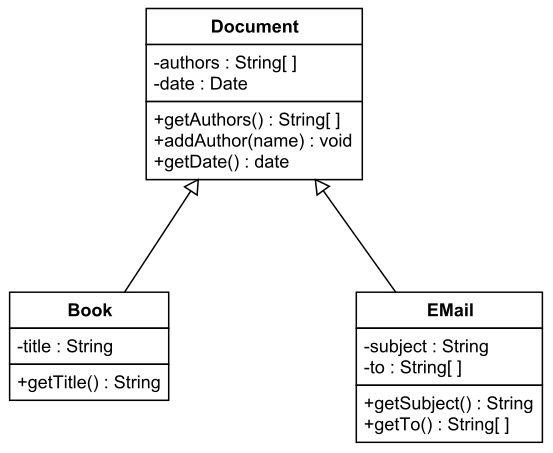
1. Use the Python to declare the following classes [the UML diagram is from



class Document:

    def \_\_init\_\_(self, authors, date):

        self.authors = authors

        self.date = date

    def getAuthors(self):

*return* self.authors

    def addAuthor(self, name):

        self.authors.append(name)

    def getDate(self):

*return* self.date

class Book(Document):

    def \_\_init\_\_(self, authors, date, title):

        super().\_\_init\_\_(authors, date)

        self.title = title

    def getTitle(self):

*return* self.title

class EMail(Document):

    def \_\_init\_\_(self, authors, date, subject, to):

        super().\_\_init\_\_(authors, date)

        self.subject = subject

        self.to = to

    def getSubject(self):

*return* self.subject

    def getTo(self):

*return* self.to

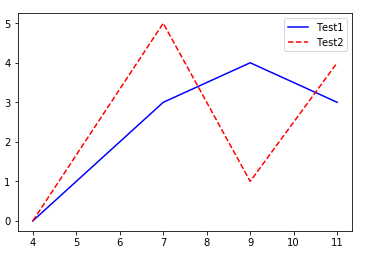
1. Use O(n) to measure the following formular

Answer: O(n)

Answer: O()

Answer: O()

1. If the probability of an event occurring is 0.75, the probability of it not occurring must be 0.25.
2. Replace the ‘?’ with the correct information according to above output histogram.



import pylab

xVals = [4, 7, 9, 11]

yVals1 = [0, 3, 4, 3]

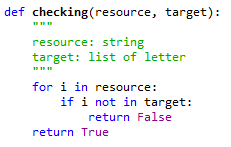
yVals2 = [0, 5, 1, 4]

pylab.plot(xVals, yVals1, '-', label='Test1')

pylab.plot(xVals, yVals2, '--', label='Test2')

pylab.legend()

1. Describe what the following code is doing?



This code is looping through each letter in the string and making sure that each letter is in the target (which is a list of letters). It effectively searches to make sure each unique character in the resource is included in the target list, while allowing more letters than in the given resource to be in the target

For example if given the resource = ‘hello’ and the target = [‘h’, ‘e’, ‘l’, ‘o’, ‘q’], this would return true. If given the resource = ‘hello’ and the target = [‘h’, ‘e’], this would return false as the letter l and o are missing.