

```
In [1]: # Q1 PdfFileReader() needs to be opened in read-binary mode by passing 'rb' as the second argument to open().
#       Likewise, the File object passed to PyPDF2.PdfFileWriter() needs to be opened in write-binary mode with 'wb'.
```

```
In [8]: # Q2 You can get a Page object by calling the getPage() method 2 on a PdfFileReader object
#       and passing it the page number of the page you're interested in, in our case, 0.
```

```
In [3]: # Q3 The total number of pages in the document is stored in the numPages attribute of a PdfFileReader object 1.
```

```
In [4]: # Q4 Before we obtain the page object, the pdf has to be decrypted by calling .decrypt('swordfish').
```

```
In [5]: #Q5 pageObj.rotateClockwise(180)
#       The rotateClockwise() and rotateCounterClockwise() methods. The degrees to rotate is passed as an integer argument
```

```
In [9]: # Q6 Paragraph Object : A document contains multiple paragraphs. A paragraph begins on a new line and contains multiple
#       runs. The Document object contains a list of Paragraph objects for the paragraphs in the document.
#       (A new paragraph begins whenever the user presses ENTER or RETURN while typing in a Word document.)

#       Run Objects : Runs are contiguous groups of characters within a paragraph with the same style
```

```
In [ ]: #Q7 #pip install python-docx
import docx
doc = docx.Document('abc.docx')
doc.paragraphs
#By using doc.paragraphs
```

```
In [12]: # Q8 A Run object has bold, underline, italic, strike and outline variables
```

```
In [13]: # Q9 Runs can be further styled using text attributes. Each attribute can be set to one of three values:
#       True (the attribute is always enabled, no matter what other styles are applied to the run),
#       False (the attribute is always disabled),
#       None (defaults to whatever the run's style is set to)
```

```
In [14]: # Q10 By Calling the docx.Document() function.
```

```
In [16]: # Q11
import docx
doc = docx.Document()

doc.add_paragraph('Hello there!')
doc.save('hellothere.docx')
```

```
In [ ]: # Q12 integer from 0 to 4
#       The arguments to add_heading() are a string of the heading text and an integer from 0 to 4.
#       The integer 0 makes the heading the Title style, which is used for the top of the document.
#       Integers 1 to 4 are for various heading levels, with 1 being the main heading and 4 the lowest subheading
```