1. In what modes should the PdfFileReader() and PdfFileWriter() File objects will be opened?

Ans. The PdfFileReader() File object should be opened in binary read mode ('rb'), and the PdfFileWriter() File object should be opened in binary write mode ('wb').

2. From a PdfFileReader object, how do you get a Page object for page 5?

Ans.

from PyPDF2 import PdfFileReader

pdf\_reader = PdfFileReader('example.pdf') # Replace 'example.pdf' with your actual PDF file

page\_number = 4 # Page numbers are 0-indexed, so 4 corresponds to page 5

page = pdf\_reader.getPage(page\_number)

3. What PdfFileReader variable stores the number of pages in the PDF document?

Ans. The numPages variable of a PdfFileReader object stores the number of pages in the PDF document. We can access it like this:

from PyPDF2 import PdfFileReader

pdf\_reader = PdfFileReader('example.pdf') # Replace 'example.pdf' with your actual PDF file

num\_pages = pdf\_reader.numPages

4. If a PdfFileReader object’s PDF is encrypted with the password swordfish, what must you do before you can obtain Page objects from it?

Ans. If a PdfFileReader object's PDF is encrypted with the password 'swordfish', you need to use the decrypt() method before obtaining Page objects. This method takes the password as an argument:

from PyPDF2 import PdfFileReader

pdf\_reader = PdfFileReader('encrypted.pdf') # Replace 'encrypted.pdf' with your actual encrypted PDF file

pdf\_reader.decrypt('swordfish')

5. What methods do you use to rotate a page?

Ans. To rotate a page, we can use the rotateClockwise(degree to rotate) or rotateCounterClockwise() methods of the Page object:

6. What is the difference between a Run object and a Paragraph object?

Ans. Run object represents a contiguous run of text with the same character style, and a Paragraph object represents a single paragraph of text.

7. How do you obtain a list of Paragraph objects for a Document object that’s stored in a variable named doc?

Ans. To obtain a list of Paragraph objects for a Document object stored in a variable named doc, you can use the paragraphs attribute:

from docx import Document

doc = Document('example.docx') # Replace 'example.docx' with your actual Word document file

paragraphs = doc.paragraphs

8. What type of object has bold, underline, italic, strike, and outline variables?

Ans . A Run object has attributes such as bold, underline, italic, strike, and outline that allow you to apply character formatting to the text within a paragraph.

9. What is the difference between False, True, and None for the bold variable?

Ans. The bold attribute of a Run object can take the values True, False, or None. True means the text is bold, False means it is not bold, and None means the bold property is inherited from the paragraph style.

10. How do you create a Document object for a new Word document?

from docx import Document

doc = Document()

11. How do you add a paragraph with the text 'Hello, there!' to a Document object stored in a variable named doc?

Ans.

from docx import Document

doc = Document()

doc.add\_paragraph('Hello, there!')

12. What integers represent the levels of headings available in Word documents?

Ans. In Word documents, heading levels are represented by integers ranging from 0 to 9, where 0 is the Title level, 1 is the Heading 1 level, 2 is the Heading 2 level, and so on up to 9 being the smallest heading level. So, for example, the integer 1 represents the Heading 1 level.