Assignment-12

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

Ans1-you don't explicitly open the PDF file objects using open() with specific modes when working with PdfFileReader and PdfFileWriter in PyPDF2. Instead, you provide the PDF file's path when creating instances of these classes, and the library handles the file I/O internally

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

Ans2- from PyPDF2 import PdfFileReader

pdf\_reader = PdfFileReader(open('example.pdf', 'rb'))

page\_5 = pdf\_reader.getPage(4)

To get a Page object for page 5 from a PdfFileReader object in PyPDF2, you can use the getPage() method.

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

Ans3- In PyPDF2, the number of pages in a PDF document is stored in the numPages attribute of a PdfFileReader object. You can access this attribute to retrieve the total number of pages in the PDF document.

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?

Ans4-If a PdfFileReader object's PDF is encrypted with the password "swordfish," you must provide the password before you can obtain Page objects or perform any operations on the PDF. To do this, you should set the password using the decrypt() method of the PdfFileReader object.

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

Ans5- We open the PDF file for reading using PdfFileReader.

We create a PdfFileWriter object to modify the PDF.

We loop through each page in the PDF using a for loop.

For each page, we use the rotateClockwise() method to rotate it clockwise by 90 degrees. You can also use rotateCounterClockwise() to rotate it counterclockwise.

The rotated page is added to the PdfFileWriter object using the addPage() method.

Finally, we save the modified PDF to a new file

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

Ans6-a) Paragraph Object:A "Paragraph object" represents a single paragraph of text within a document.

A paragraph can contain multiple runs and other formatting elements.

b) Run Object:A "Run object" represents a contiguous range of text within a paragraph that has the same formatting properties.

Runs are the smallest units of text that can be formatted independently within a paragraph.

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

Ans7- To obtain a list of Paragraph objects from a Document object stored in a variable named doc, you can use the paragraphs attribute of the Document object. The paragraphs attribute contains a list of all the paragraphs in the document

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

Ans8- In libraries like python-docx for working with Microsoft Word documents, these attributes are often found within a "Run object." A "Run object" represents a contiguous range of text within a paragraph that has the same formatting properties.

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

Ans9a) -False:When you set bold to False, it means that the text should not be bold.

This value explicitly specifies that the text should not have the bold formatting applied.

b) True:When you set bold to True, it means that the text should be bold.

This value explicitly specifies that the text should have the bold formatting applied.

c) None:When bold is set to None, it typically means that the formatting is not explicitly specified for boldness. In many text processing libraries, setting bold to None means that the text inherits its formatting from the surrounding context or the document's default styles.

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

Ans10 -from docx import Document

doc = Document()

doc.add\_heading('My New Document', level=1)

doc.add\_paragraph('This is the first paragraph of my new document.')

doc.add\_paragraph('This is the second paragraph.')

doc.save('new\_document.docx')

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

Ans11- from docx import Document

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

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

Ans12- Heading Level 1 (Heading 1): Represented by the integer 1. This is typically the highest-level heading and is used for major sections or chapters in a document.

Heading Level 2 (Heading 2): Represented by the integer 2. Used for subsections or subchapters within Heading Level 1 sections.

Heading Level 3 (Heading 3): Represented by the integer 3. Used for further subsections within Heading Level 2 sections.

Heading Level 4 (Heading 4): Represented by the integer 4. Used for even deeper subsections within Heading Level 3 sections.

Heading Level 5 (Heading 5): Represented by the integer 5. Used for additional levels of subsections within Heading Level 4 sections.

Heading Level 6 (Heading 6): Represented by the integer 6. Used for the lowest-level subsections, typically nested within Heading Level 5 sections.