PYTHON ASSIGNMENT 12

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

2. **From a PdfFileReader object, how do you get a Page object for page 5?**
   To extract text from a page, you need to get a Page object, which represents a single page of a PDF, from a PdfFileReader object. You can get a Page object by calling the getPage() method on a PdfFileReader object and passing it the page number of the page you're interested in—in our case, 0. PyPDF2 uses a zero-based index for getting pages: The first page is page 0, the second is page 1, and so on. This is always the case, even if pages are numbered differently within the document. For example, say your PDF is a three-page excerpt from a longer report, and its pages are numbered 42, 43, and 44. To get the first page of this document, you would want to call pdfReader.getPage(0), not getPage(42) or getPage(1).

3. **What PdfFileReader variable stores the number of pages in the PDF document?**
   The total number of pages in the document is stored in the numPages attribute of a PdfFileReader object.

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?**
   Before we obtain the page object, the pdf has to be decrypted by calling .decrypt('swordfish').

5. **What methods do you use to rotate a page?**
   The rotateClockwise() and rotateCounterClockwise() methods. The degrees to rotate is passed as an integer argument.

   ```
   pageObj.rotateClockwise(180)
   ```

6. **What is the difference between a Run object and a Paragraph object?**
   - 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 RETURNS while typing in a Word document.)

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

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

```
#!pip install python-docx
    import docx
    doc = docx.Document('abc.docx')
    doc.paragraphs
#By using doc.paragraphs
```

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

   A Run object has bold, underline, italic, strike and outline variables.

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

   - 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)

   - True always makes the Run object bolded and False makes it always not bolded, no matter what the style's bold setting is. None will make the Run object just use the style's bold setting.

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

    By Calling the docx.Document() function.

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

```
import docx
doc = docx.Document()

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

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

    - 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.