Assignment 12:

## Q1

For PdfFileReader(): The PDF file object should be opened in "rb" mode, which stands for "read binary." This mode is used to indicate that the file will be read as a binary file, which is necessary for reading PDF files.

For PdfFileWriter(): The PDF file object should be opened in "wb" mode, which stands for "write binary." This mode is used to indicate that the file will be written as a binary file. Opening the file in "wb" mode allows the PyPDF2 library to write PDF content into the file.

## Q2

To get a Page object for page 5 from a PdfFileReader object, we can use the getPage() method and pass the page number as an argument.

page\_5 = PdfFileReader.getPage(4)

## Q3

The numPages variable in the PdfFileReader object stores the number of pages in the PDF document.

## Q4

Before obtaining Page objects from a PdfFileReader object that has an encrypted PDF with the password "swordfish", we need to decrypt the PDF by calling the decrypt() method of the PdfFileReader object and passing the password as an argument.

## Q5

The rotate() method is used to rotate the page in the PDF file. Can use the rotateClockwise() or rotateCounterClockwise() methods of the PageObject.

## Q6

A Paragraph object represents a single paragraph of text in a document. It typically consists of multiple runs of text. A paragraph is a logical unit that separates blocks of text and is often delimited by line breaks or paragraph breaks.

A Run object represents a continuous span of text within a paragraph that shares the same formatting properties. It is a smaller unit within a paragraph that can be styled independently.

## Q7

To obtain a list of Paragraph objects from a Document object in python, we can use the paragraphs attribute. **.paragraphs** returns a list of all the Paragraph objects in the document.

## Q8

The Run object in python-docx has the variables bold, underline, italic, strike, and outline to represent different formatting options for text. The bold, underline, italic, strike, and outline variables control the formatting of the text in the Run object.

## Q9

True means that the text in the Run object will be bold.

False means that the text in the Run object will not be bold.

None means that the text in the Run object will inherit the bold setting from the style of the paragraph.

## Q10

To create a Document object for a new Word document using the python, we can simply call the **Document()** constructor. **Document()** creates a new Document object, which represents a new, empty Word document.

## Q11

To add a paragraph with the text 'Hello, there!' to a Document object stored in a variable named doc, we can use the **add\_paragraph()** method. **add\_paragraph()** is called on the Document object to add a new paragraph with the specified text. The method returns a Paragraph object, which can be stored in a variable if needed.

## Q12

In Word documents, the levels of headings available are represented by integers from 0 to 4. The integer 0 represents 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. The level of a heading can be set using the **add\_heading()** method of the Document object. The **add\_heading()** method takes two arguments: the text of the heading and the level of the heading.