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

**PdfFileReader() should be opened in read-binary ‘rb’, and PdfFileWriter() should be opened in write-binary mode ‘wb’.**

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

**pdf\_file = open(‘filename.pdf’, ‘rb’)**

**reader = PyPDF2.pdfFileReader(pdf\_file)**

**page = reader.getPage(5)**

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

**The number of pages is stored in numPages attribute of 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?

**File has to be decrypted:**

**pdf\_file = open(‘encrypted.pdf’, ‘rb’)**

**reader = PyPDF2.pdfFileReader(pdf\_file)**

**reader.decrypt(‘swordfish’)**

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

**rotateClockwise() and rotateCounterClockwise() with specified number of degrees to rotate.**

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

**Paragraph object is represented by paragraphs in the document and separated by return or enter. Run object represents parts of text in the paragraph that has same font style continuously.**

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

**doc.paragraphs**

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

**Run object**

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

**If set to true for bold, Run will be bold, for False – Run will be not bold. For None Run will be default to style’s boldness.**

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

**doc = docx.Document()**

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

**doc.add\_paragraph(‘Hello, there!’)**

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

**0, 1, 2, 3, 4**