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

Question 2.
From a PdfFileReader object, how do you get a Page object for page 5?
Answer 2:
Calling getPage(4) will return a page object for page 5, since page 0 is the first page.

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

Question 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?
Answer 4:
To read an encrypted PDF, call the decrypt() function and pass the password as a string, in our case it is swordfish. After we call decrypt() with the correct password, you'll see that calling getPage() no longer causes an error. Call decrypt('swordfish').

Question 5.
What methods do you use to rotate a page?
Answer 5:
The pages of a PDF can be rotated in 90-degree increments with the rotateClockwise() and rotateCounterClockwise() methods.

Question 6.
What is the difference between a Run object and a Paragraph object?
Answer 6:
A document contains multiple paragraphs. A paragraph begins on a new line and contains multiple runs. Runs are contiguous groups of characters within a paragraph.

Question 7.
How do you obtain a list of Paragraph objects for a Document object that's stored in a variable named doc?
Answer 7:
len(doc.paragraphs)

Question 8.

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

Answer 8:

Run Object

Question 9.

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

Answer 9:

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

Question 10.

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

Answer 10:

mydoc = docx.Document()

Question 11.

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

Answer 11:

doc.add_paragraph('Hello, there!')

Question 12.

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

Answer 12:

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. The add_heading() function returns a Paragraph object to save you the step of extracting it from the Document object as a separate step.