### 1. In what modes should the `PdfFileReader()` and `PdfFileWriter()` file objects be opened?

- \*\*PdfFileReader\*\*: The file object should be opened in binary read mode (`'rb'`).

- \*\*PdfFileWriter\*\*: The file object should be opened in binary write mode (`'wb'`).

#### Example:

```python

from PyPDF2 import PdfFileReader, PdfFileWriter

# PdfFileReader

with open('example.pdf', 'rb') as file:

reader = PdfFileReader(file)

# PdfFileWriter

with open('output.pdf', 'wb') as file:

writer = PdfFileWriter()

writer.write(file)

```

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

You can get a `Page` object for page 5 using the `getPage()` method. Note that page numbering starts from 0, so page 5 is at index 4.

#### Example:

```python

page = reader.getPage(4)

```

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

The `numPages` attribute of a `PdfFileReader` object stores the number of pages in the PDF document.

#### Example:

```python

num\_pages = reader.numPages

```

### 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?

You must call the `decrypt()` method with the password before obtaining `Page` objects.

#### Example:

```python

reader.decrypt('swordfish')

page = reader.getPage(0)

```

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

You use the `rotateClockwise()` or `rotateCounterClockwise()` methods to rotate a page.

#### Example:

```python

page = reader.getPage(0)

page.rotateClockwise(90) # Rotates the page 90 degrees clockwise

```

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

- \*\*Paragraph object\*\*: Represents a single paragraph of text, which can contain multiple runs.

- \*\*Run object\*\*: Represents a contiguous run of text within a paragraph that shares the same formatting.

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

You can obtain a list of `Paragraph` objects using the `paragraphs` attribute.

#### Example:

```python

paragraphs = doc.paragraphs

```

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

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

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

- \*\*`False`\*\*: Explicitly sets the text to not be bold.

- \*\*`True`\*\*: Explicitly sets the text to be bold.

- \*\*`None`\*\*: Inherits the bold setting from the style hierarchy or defaults if no explicit setting is applied.

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

You can create a `Document` object using the `Document` class from the `python-docx` library.

#### Example:

```python

from docx import Document

doc = Document()

```

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

You use the `add\_paragraph()` method.

#### Example:

```python

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

```

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

The integers 0 through 9 represent the levels of headings available in Word documents, where 0 is the title, 1 is the main heading, and 2 through 9 are sub-headings of increasing granularity.

#### Example:

```python

doc.add\_heading('Main Heading', level=1)

doc.add\_heading('Sub-heading', level=2)

```