A **Reader** is used to open a .doc/.docx/.pdf/.epub/.djvu files. Also, sometimes, they can be used to edit those files too.

A **DocReader** is a kind of **Reader** which is used to read ./doc/.docx files and edit them.
It prints “<DocReaderName> is reading <FileName>” when it is passed a file to read and “<DocReaderName> is editing <FileName>” when it is passed a file to edit. Here, **DocReaderName** is the name of the **DocReader** object/software and FileName is a string.
Example: “Microsoft word is reading manifesto.docx”, “Google Docs is editing Assignment.doc”

A **PDFReader** is a kind of **Reader** which is used to read .pdf/.epub/.djvu files and edit them.
It prints “<PDFReaderName> is reading <FileName>” when it is passed a file to read and “<PDFReaderName> is editing <FileName>” when it is passed a file to edit. Here,**PDFReaderName** is the name of the **PDFReader** object/software and FileName is a string.
Example: “Foxit Reader is reading manifesto.pdf”, “WinDJVU is editing ThePrience.djvu”

If any file is passed to these readers which they don’t support (.doc/.docx for **PDFReaders** or .epub/.djvu/.pdf **for DocReaders**) they throw an exception showing the message “**<ReaderName> cannot open <fileextension> format files**”.
Example: “Microsoft Word cannot open .djvu format files”

Also, if any file name does not end with any of .doc/.docx/.pdf/.epub/.djvu, they throw an exception showing the message “<FileName> is not a readeable file”.
Example: “abc.jpg is not a readable file”.

(i) design and develop some classes with inheritance/abstraction relationship among them and appropriate constructors. [8]
(ii) implement two exceptions. [8]
(iii) write a demo/main function to demonstrate the functionalities. In demo, develop a **DocReader** object “**MicrosoftReader”** and a **PDFReader** object “**FoxitReader**” to show the demo.[4]
You need to use String endswith() method. Here is a demo code.

```
public class JavaExample {
   public static void main(String[] args) {
       String str = "Java String tutorial";
       if(str.endsWith("tutorial")) {
              System.out.println("The Given String ends with tutorial");
       }
    }
}
```

Also, while declaring the prototype of a function in abstract class/interface, you can write it like this

```
public void doThisJob() throws XException, YException;
```

And in implementation,

```
public void doThisJob() throws XException, YException
{
…..
….}
```