

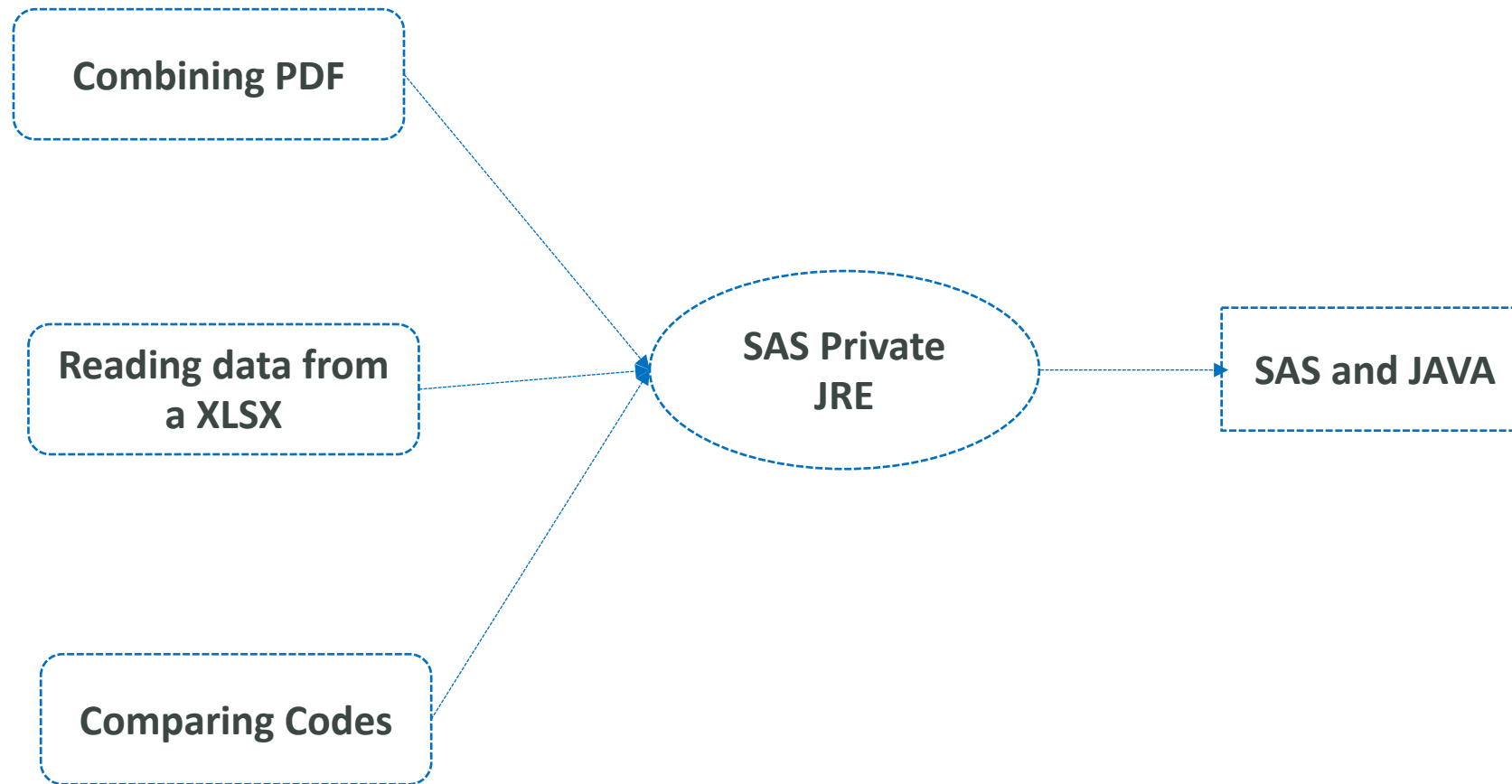
SAS is still a winner – with a little help from JAVA

- Praveen Kumar J
- Tech Lead

Introduction

- SAS provides a **one-stop-shop** solution for data analytics.
- But, there is always a break point for few process, where a SAS user depends on a third party tool or by manual proceedings.
- Process like combining PDF or RTF, comparing codes and also sometimes reading xlsx file.
- Does SAS support these process? Yes, with the help of SAS Private JRE.
- This session will explain how SAS is supporting for the above mentioned process without third party or manual Proceeding.

Solved Problems



JAVA and JAVA Class file

- JAVA – A front end programming Language
- Uses:
 - Object oriented programming Language
 - Platform independent – Class files
 - Robust
 - Code-reusable

SAS Private JRE

- SAS Private JRE – JAVA Runtime Environment – is a third party tool which comes in-built in SAS
- SAS Private JRE started with SAS 9.1.3 and currently in SAS 9.4 there are advancement to have few additional features in Private JRE.
- Private JRE: Key feature
 - Change of folder structure
 - Secure and Safe
 - Easily accessible

Inter-linking - *SAS and JAVA / JAVA Class File*

- Steps to create and execute Java program
 - Create a JAVA Program
 - Compile and obtain the JAVA Class file
 - Use the SAS Macro from SAS Support to perform the link
 - Write a SAS Program that call the Class file
 - Execute and output is ready

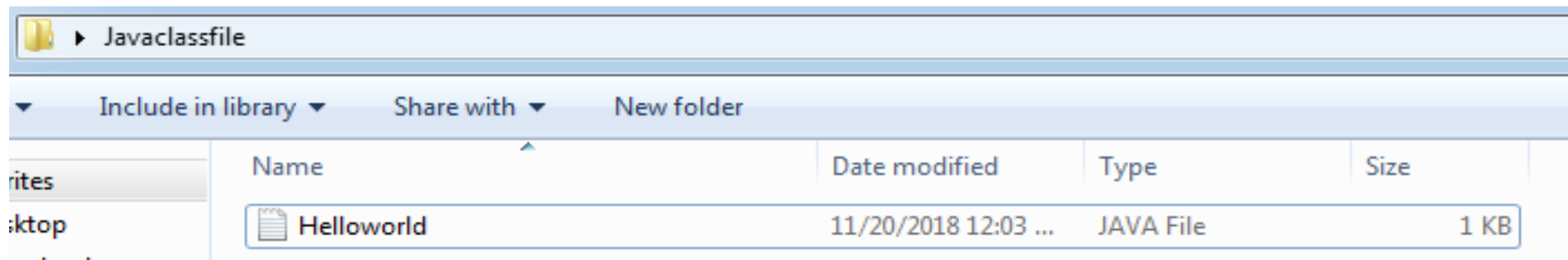
Example 1: *Simple Program – Hello World*

- JAVA Program to print “Hello World Test”

```
public class Helloworld {  
  
    public static String disp() {  
        return "Hello World: The value is from JAVA and Print in SAS Window";  
    }  
  
    public static void main(String args[]) {  
        System.out.println(disp());  
    }  
}
```

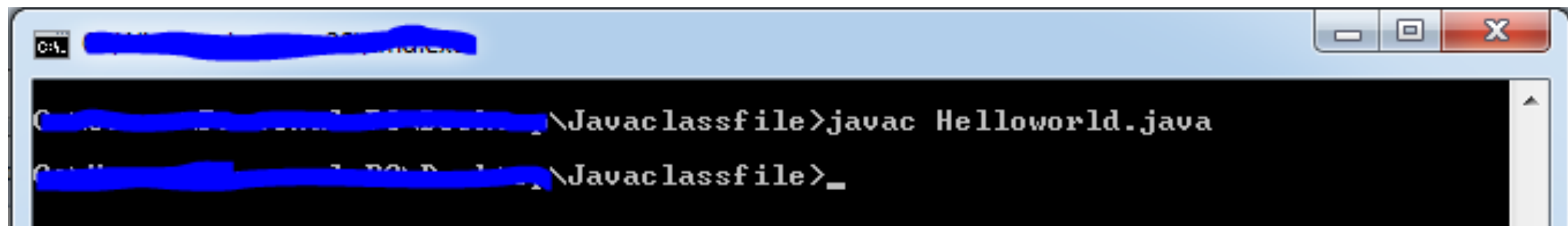
Example 1: Simple Program – Hello World (contd.)

- Compiling JAVA and Obtaining a Class file



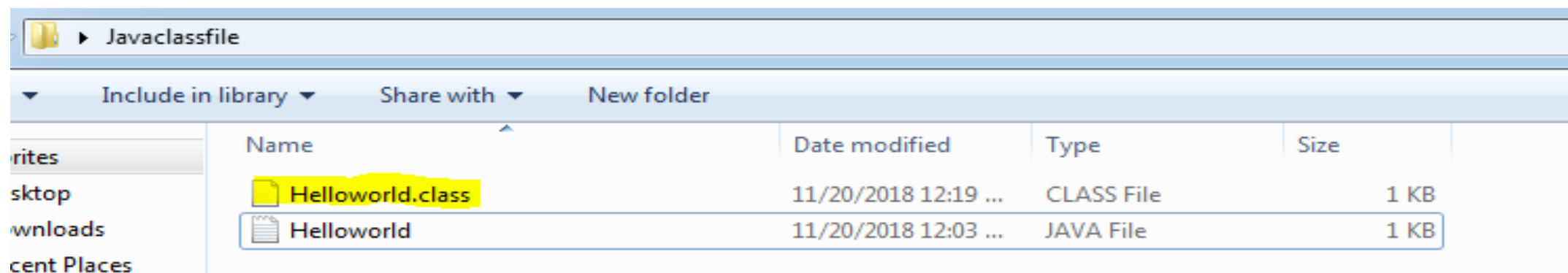
- Open CMD and navigate to folder where JAVA file resides for compiling and to obtain the CLASS file
- Ensure the navigate place is the JAVA CLASSPATH

Example 1: Simple Program – Hello World (contd.)



```
C:\Users\user> cd C:\Java\src\Javaclassfile
C:\Java\src\Javaclassfile> javac Helloworld.java
C:\Java\src\Javaclassfile>
```

- Once after submitting the code in same folder we could see CLASS file create as below



Name	Date modified	Type	Size
Helloworld.class	11/20/2018 12:19 ...	CLASS File	1 KB
Helloworld	11/20/2018 12:03 ...	JAVA File	1 KB

Example 1: *Simple Program – Hello World* (contd.)

- Linking CLASS file to SAS Program and obtain the output.
- Below three macro a user can find it in SAS SUPPORT website (<http://support.sas.com/kb/38/518.html>)

```
%init_classpath_update;
```

```
%add_to_classpath(...\Javaclassfile) ;
```

```
%reset_classpath;
```

Example 1: Simple Program – Hello World *(contd.)*

SAS Program and Final Output

```
%init_classpath_update;  
%add_to_classpath(....\Javaclassfile);  
DATA out;  
    length output $200;  
    DECLARE JavaObj df("Helloworld");  
    df.callStaticStringMethod("disp",output);  
    put output=;  
  
RUN;  
  
%reset_classpath;
```

Example 1: Simple Program – Hello World (contd.)

- Final output:
 - Log Window

```
output=Hello World: The value is from JAVA and Print in SAS Window
NOTE: The data set WORK.OUT has 1 observations and 1 variables.
NOTE: DATA statement used (Total process time):
      real time           0.03 seconds
      cpu time            0.01 seconds
```

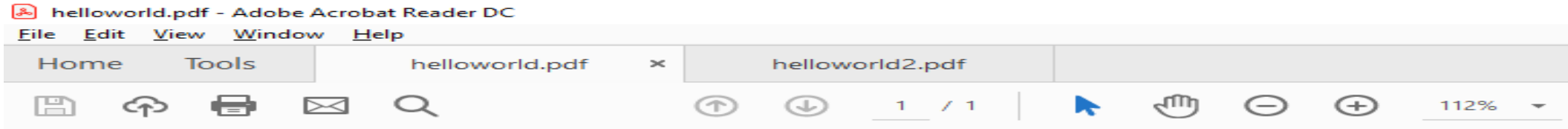
- Dataset

	output
1	Hello World: The value is from JAVA and Print in SAS Window

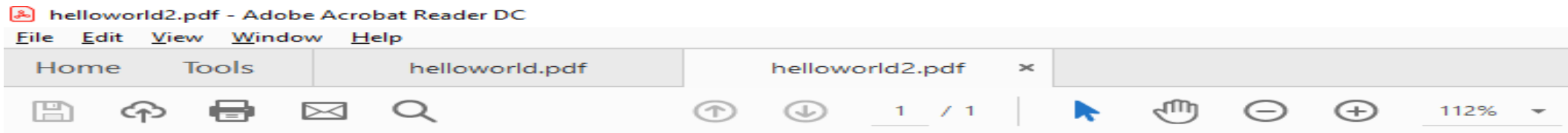
Combining PDF's into Single PDF

- JAVA Program requirement:
 - itextpdf package
 - PDF Reader
 - PDF writer
 - List
 - Input filestream
 - Ouptut filestream
- There is a open source code available in below website “<https://examples.javacodegeeks.com/core-java/itext-merge-pdf-example/>”

Two Separate PDF

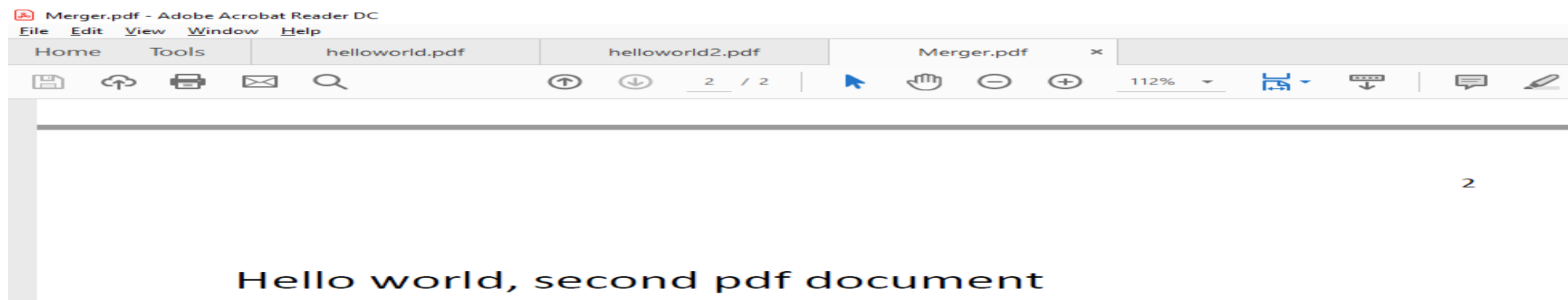
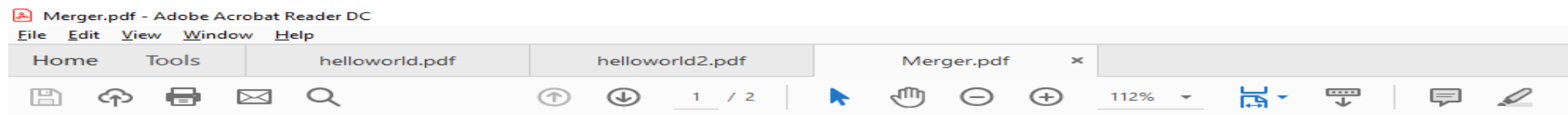


Hello world, first pdf document



Hello world, second pdf document

Merged PDFs into a single file - *Output*



Pros and Cons

Pros:

Platform independent, works like a macro.

No need to purchase additional software or Package.

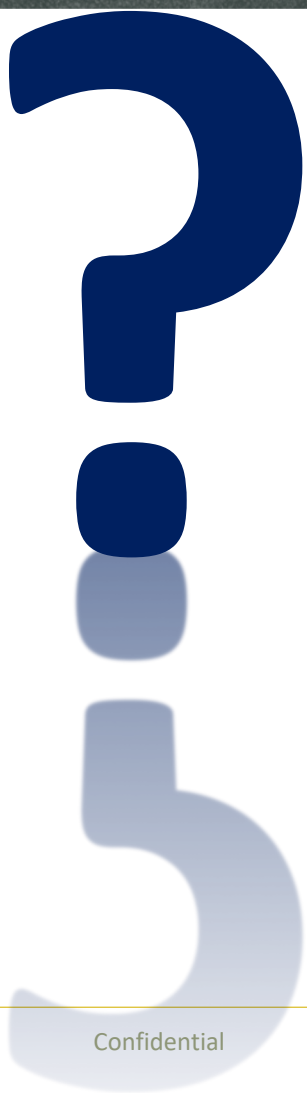
Less time consuming.

No cost.

Can be updated in future as required.

Cons:

Needs a JAVA Programmer



Thank You!