

**ISIT312 Big Data Management**  
**SIM S4 2025**  
**Assignment 1**

---

**Scope**

The objectives of Assignment 1 include **implementation of HDFS applications, implementation of simple MapReduce applications, and describing an implementation of complex MapReduce applications.**

This assignment is due on **19 October 2025 by 9:00 pm** Singaporean Time (SGT).

This assignment is worth **10%** of the total evaluation in the subject.

The assignment consists of 3 tasks, and the specification of each task starts from a new page.

Only electronic submission through Moodle at:

<https://moodle.uowplatform.edu.au/login/index.php>

will be accepted. A submission procedure is explained at the end of Assignment 1 specification.

A policy regarding late submissions is included in the subject outline. Only one submission of Assignment 1 is allowed and only one submission per student is accepted.

A late submission penalty (25% of the total mark) will be applied for every 24 hours late.

A submission that contains an incorrect file attached is treated as a correct submission with all consequences coming from the evaluation of the file attached.

All files left on Moodle in a state "Draft (not submitted)" **will not be evaluated.**

An implementation that does not compile well due to one or more syntactical and/or run time errors scores no marks.

The first assignment is an **individual assignment**, and it is expected that all its tasks will be solved **individually without any cooperation** with the other students. However, it is allowed to declare in the submission comments that a particular component or task of this assignment has been implemented in cooperation with another student. In such a case evaluation of a task or component may be shared with another student. In all other cases plagiarism will result in a **FAIL** grade being recorded for entire assignment. If you have any doubts, questions, etc. please consult your lecturer or tutor during laboratory/tutorial classes or over e-mail.

---

### **Task 1 (3 marks)**

#### **Implementation of HDFS application**

Implement a HDFS application that merges two files located in HDFS into one file also located in HDFS.

The application must have the following parameters.

- (1) A path to, and a name of the first input file in HDFS.
- (2) A path to, and a name of the second input file in HDFS.
- (3) A path to, and a new name of an output file to be created in HDFS. The file is supposed to contain the contents of the first input file followed by the contents of the second input file.

Perform the following steps.

Implement the application and save its source code in a file `solution1.java`.

Upload two files to HDFS. The contents, the name, and the locations of the files in HDFS are up to you.

When ready, compile, create `jar` file, and process your application. Display the results created by the application.

Use Hadoop to provide a piece of evidence that two files uploaded into HDFS have been successful merged into one file in HDFS.

#### **Deliverables**

A file `solution1.java` with a source code of the application that merges two HDFS files.  
A file `solution1.pdf` that contains the contents of Terminal window with a report from compilation, creation of `jar` file, uploading to HDFS two small files for testing, processing of the application, and evidence that two files uploaded into HDFS has been successful merges in one file in HDFS with explanation of how the statements work.

---

## **Task 2 (4 marks)**

### **Implementation of MapReduce application**

Assume, that a speed camera records the speed of passing cars and saves the measurements in a text file. The speed of each car is measured in kilometres per hour. A single row in the file contains a car registration number, a location of the camera, a date when the speed has been measured, and the speed of a car with the recorded registration number. The values are always separated with a single blank.

For example, a sample file (*SpeedCamera.txt*) with the speed measurements contains the following lines:

```
PKR856 AYE 14-NOV-2024 80
UPS234 CTE 20-FEB-2025 110
PKR856 PIE 20-MAR-2025 90
PKR856 PIE 17-JUN-2025 100
UPS234 CTE 22-SEP-2024 100
UPS234 CTE 03-AUG-2025 90
```

Assume, that a speed limit in a location of the speed camera is 90 kilometres per hour.

Your task is to implement a MapReduce application, that finds an average speed of all cars, that exceeded a speed limit in the location of the speed camera.

Save your solution in a file `solution2.java`.

When ready, compile, create a `jar` file, and process your application. Display the results created by the application. The result of your application includes (1) the content of your input file, (2) the car registration number, the location of the camera, and the average speed that exceed the speed limit. When finished, Copy and Paste the messages from a Terminal screen into a file `solution2.pdf`.

A sample output of the application is as follows:

```
PKR856 - AYE      Speed limit exceeded!! Average Speed : 92
PKR856 - CTE      Speed limit exceeded!! Average Speed : 101
PKR856 - PIE      Speed limit exceeded!! Average Speed : 100
SLF123 - CTE      Speed limit exceeded!! Average Speed : 110
UPS234 - CTE      Speed limit exceeded!! Average Speed : 102
```

### **Deliverables**

A file `solution2.java` with a source code of the application that implement the functionality of the problem statement specified above. A file `solution2.pdf` with a report from the compilation of your code, the creation of the `jar` file, the processing of your application, the listing of your input file with the speed measurements, and the results of processing the `solution2.java`.

---

### **Task 3 (3 marks)**

#### **Implementation of MapReduce application**

An application MinMax described in an Exercise 2 has the functionality the same as the following SQL statement.

```
SELECT key, MIN(value), MAX(value)
FROM Sequence-of-key-value-pairs
GROUP BY key;
```

Extend Java code of the application such that it implements the functionality the same as the following SQL statement.

```
SELECT key, MAX(value), MIN(value), AVG(value), SUM(value)
FROM Sequence-of-key-value-pairs
GROUP BY key;
```

Save your solution in a file `solution3.java`.

When ready, compile, create the `jar` file, and process your application. To test your application, you can use a file `sales.txt` included in the zipped file of this specification. Display the results created by the application. When finished, Copy and Paste the messages from a Terminal screen into a file `solution3.pdf`.

A sample output of the application is as follows:

bolt	Max:51	Min:1	Avg:22	Total:137
drill	Max:14	Min:1	Avg:4	Total:19
nail	Max:50	Min:5	Avg:27	Total:55
screw	Max:78	Min:13	Avg:42	Total:257
washer	Max:56	Min:3	Avg:23	Total:139

#### **Deliverables**

A file `solution3.java` with a source code of the application that implement the functionality of `SELECT` statement given above. A file `solution3.pdf` with a report from compilation, creation of the `jar` file, processing of your application, and screen captures of the results of processing `solution3.java`.

---

## Submission of Assignment 1

**Note that you have only one submission. So, make absolutely sure that you submit the correct files with the correct contents. Please submit an Academic Consideration in SOLS if an extension (1 week maximally) is required.**

Please combine the files **solution1.pdf**, **solution2.pdf**, and **solution3.pdf** as a single pdf (**solutions.pdf**) first, then zip the files **solutions.pdf**, **solution1.java**, **solution2.java**, and **solution3.java** into a single zipped file (**A1-solutions.zip**). Please submit the zipped file through Moodle in the following way:

- (1) Access Moodle at **<http://moodle.uowplatform.edu.au/>**
- (2) To login use a **Login** link located in the right upper corner the Web page or in the middle of the bottom of the Web page
- (3) When logged select a site **ISIT312 (SP425) Big Data Management**
- (4) Scroll down to a section **SUBMISSIONS**
- (5) Click at **Assignment 1** link.
- (6) Click at a button **Add Submission**
- (7) Move the zipped file **A1-solutions.zip** into an area **You can drag and drop files here to add them**. You can also use a link **Add...**
- (9) Click at a button **Save changes**
- (10) Click at a button **Submit assignment**
- (11) Click at the checkbox with a text attached: **By checking this box, I confirm that this submission is my own work, ...** in order to confirm authorship of your submission.
- (12) Click at a button **Continue**

---

*End of specification*