



# Hortonworks Data Platform Developer: Java H7G67S

This 4-day course provides a deep dive into Hadoop application development. The Students will explore how to design and develop efficient and effective MapReduce applications for Hadoop 2.0 using the Hortonworks Data Platform. The course will focus on harnessing the power of Hadoop 2.0 to manipulate, analyze and perform computations on their Big Data. This course is 50% lecture and 50% hands-on labs.

## Hortonworks Data Platform Developer: Java

**Price** USD \$2,795

**Links to local  
schedules,  
pricing and  
registration** [US/Canada](#)  
[Mexico/Latin America](#)  
[Brazil](#)

**HP course #** H7G67S

**Category** Big Data

**Duration** 4 days

## Audience

- Students should be experienced Java software engineers who need to design and develop Java MapReduce applications for Hadoop 2.0

## Prerequisites

- This course assumes students have experience developing Java applications and using a Java IDE. Labs are completed using the Eclipse IDE and Maven. No prior Hadoop knowledge is required

## Course objectives

At the completion of the course students will be able to:

- Explain Hadoop 2.0 and the Hadoop Distributed File System
- Explain the new YARN framework in Hadoop 2.0
- Develop a Java MapReduce application
- Run a MapReduce application on YARN
- Use combiners and in-map aggregation to improve the performance of a MapReduce job
- Write a custom partitioner to avoid data skew on reducers
- Perform a secondary sort by writing custom key and group comparator classes
- Recognize use cases for the various built-in input and output formats
- Write a custom input and output format for a MapReduce job
- Optimize a MapReduce job by following best practices
- Configure various aspects of a MapReduce job to optimize mappers and reducers

- Develop a custom RawComparator class
- Use the Distributed Cache
- Describe the various join techniques in Hadoop
- Perform a map-side join
- Use a Bloom filter to join two large datasets
- Perform unit tests using the UnitMR API
- Describe the basic architecture of HBase
- Write an HBase MapReduce application
- Describe use cases for Pig and Hive
- Write a simple Pig script to explore and transform big data
- Write a Pig UDF (User-Defined Function) in Java
- Execute a Hive query
- Write a Hive UDF in Java
- Use the JobControl class to create a workflow of MapReduce jobs
- Use Oozie to define and schedule workflows

## Benefits to you

- This course will provide in depth explanation on how to perform Hadoop 2.0 application development

## Course outline

### Day 1

- Understanding Hadoop 2.0 and HDFS
- Writing MapReduce Applications
- Map aggregation

### Day 2

- Partitioning and Sorting
- Input and Output Formats
- Optimizing MapReduce Jobs

### Day 3

- Advanced MapReduce Features
- Unit Testing
- HBase Programming

### Day 4

- Pig Programming
- Hive Programming
- Defining Workflow

**Lab Content**

- Configuring a Hadoop 2.0 Development Environment
- Putting data into HDFS using Java
- Write a distributed grep MapReduce application
- Write an inverted index MapReduce application
- Configure and use a combiner
- Writing a custom combiner
- Writing a custom partitioner
- Globally sort output using the TotalOrderPartitioner
- Writing a MapReduce job whose data is sorted using a composite key
- Writing a custom InputFormat class
- Writing a custom OutputFormat class
- Compute a simple moving average of historical stock price data
- Use data compression
- Define a RawComparator
- Perform a map-side join
- Using a Bloom filter
- Unit testing a MapReduce job
- Import data into HBase
- Writing an HBase MapReduce job
- Writing a User-Defined Pig-Function
- Writing a User-Defined Hive Function
- Defining an Oozie workflow

Learn more at

**[hpe.com/us/training/bigdata](http://hpe.com/us/training/bigdata)**