



Hortonworks Data Platform Developer: Java H7G67S (EDU-PRIV-DEV-JAVA-200)

HPE course number	H7G67S
Course length	4 days
Delivery mode	ILT
View schedule, local pricing, and register	View now
View related courses	View now

This advanced four-day course provides Java programmers a deep-dive into Hadoop 2.0 application development. Students will learn how to design and develop efficient and effective MapReduce applications for Hadoop 2.0 using the Hortonworks Data Platform. Students who attend this course will learn how to harness the power of Hadoop 2.0 to manipulate, analyze and perform computations on their Big Data.

Why HPE Education Services?

- IDC MarketScape leader 4 years running for IT education and training*
- Recognized by IDC for leading with global coverage, unmatched technical expertise, and targeted education consulting services*
- Key partnerships with industry leaders OpenStack®, VMware®, Linux®, Microsoft®, ITIL, PMI, CSA, and (ISC)²
- Complete continuum of training delivery options—self-paced eLearning, custom education consulting, traditional classroom, video on-demand instruction, live virtual instructor-led with hands-on lab, dedicated onsite training
- Simplified purchase option with HPE Training Credits

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

Detailed course outline

Day 1	<ul style="list-style-type: none"> • Understanding Hadoop 2.0 and HDFS • Writing MapReduce Applications • Map aggregation
Day 2	<ul style="list-style-type: none"> • Partitioning and Sorting • Input and Output Formats • Optimizing MapReduce Jobs
Day 3	<ul style="list-style-type: none"> • Advanced MapReduce Features • Unit Testing • HBase Programming
Day 4	<ul style="list-style-type: none"> • Pig Programming • Hive Programming • Defining Workflow
Lab Content	<p>Students will work through the following lab exercises using Eclipse, Maven, and the Hortonworks Data Platform 2.0:</p> <ul style="list-style-type: none"> • 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

Learn more at
hpe.com/ww/learnbigdata

Follow us:



© Copyright 2015–2016 Hewlett Packard Enterprise Development LP. The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Microsoft is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries. Java is a registered trademark of Oracle and/or its affiliates. The OpenStack Word Mark is either a registered trademark/service mark or trademark/service mark of the OpenStack Foundation, in the United States and other countries and is used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation or the OpenStack community. Pivotal and Cloud Foundry are trademarks and/or registered trademarks of Pivotal Software, Inc. in the United States and/or other countries. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other third-party trademark(s) is/are property of their respective owner(s).