**CSC 4760 / 6760 Spring 2020**

**Big Data Programming**

**Classroom**: Langdale Hall 305 / On Webex meetings starting from March 30, 2020, because of COVID-19

**Date/Time**: Monday/Wednesday, 2:30 pm - 4:15 pm

**CRN:** 19219 (CSC 4760), 16099 (CSC 6760), 21265 (CSC 6760)

**Credits:** 4.0

**Website:** All course materials will be uploaded to iCollege; Starting from March 30, 2020, all lectures on webex meetings will be recorded and the videos will be uploaded in iCollege “Kaltura Course Gallery”.

**Instructor:** Dr. Yubao (Robert) Wu

**Office:** 1 Park Place, Room 638 (6 floor)

**Phone:** 404-413-6125 (office)

**E-mail:** [ywu28@gsu.edu](mailto:ywu28@gsu.edu) (Please contact me through this email, please do not send emails in iCollege)

**Office Hours:** 3:00 pm - 5:00 pm, Thursday, Webex Meetings; or by appointment

**TA**: Anil Ravuru

**TA’s E-mail**: Anil Ravuru <aravuru1@student.gsu.edu>

**TA Office**: Cubical 642E at the 6th floor of 25 Park Place

**TA Office Hours**: 12:00 pm - 1 pm, Tuesday and Thursday; or by appointment

**Textbooks**:

1. Recommended: **MapReduce Design Patterns**, by Donald Miner and Adam Shook, O'Reilly Media, 2012.

Repository for MapReduce Design Patterns (O'Reilly 2012) example source code:

<https://github.com/adamjshook/mapreducepatterns>

2. Recommended: **Learning Spark: Lightning-Fast Data Analysis**, by Holden Karau, Andy Konwinsky, Patrick Wendell, and Matei Zaharia, O'Reilly Media, 2015.

Example code from Learning Spark book

<https://github.com/databricks/learning-spark>

3. Other related books:

1. **Learning Apache Spark 2.0**. by Muhammad Asif Abbasi. 2017.
2. **Spark: The Definitive Guide,** By MateiZaharia, Bill Chambers. Publisher: [O'Reilly Media](https://www.safaribooksonline.com/library/publisher/oreilly-media-inc/?utm_medium=referral&utm_campaign=publisher&utm_source=oreilly&utm_content=catalog&utm_content=catalog). Release Date: February 2018
3. **Programming Pig**: Dataflow Scripting with Hadoop
4. **Programming Hive**: Data Warehouse and Query Language for Hadoop
5. **HBase**: The Definitive Guide: Random Access to Your Planet-Size Data
6. Data-Intensive Text Processing with MapReduce. Jimmy Liu and Chris Dyer.
7. Hadoop: The Definitive Guide, 4th Edition, by Tom White, O'Reilly Media, 2015.
8. Advanced Analytics with Spark: Patterns for Learning from Data at Scale 1st Edition
9. Data Analytics with Hadoop: An Introduction for Data Scientists 1st Edition
10. Big Data Analytics with Spark: A Practitioner's Guide to Using Spark for Large Scale Data Analysis
11. Data Algorithms: Recipes for Scaling Up with Hadoop and Spark

**Course Content**: This course will cover the frameworks, tools, and programming languages that are most commonly used in Big Data Programming, including MapReduce, Hadoop, Spark, Hive, Pig, HPCC Systems etc. Focus will be on algorithms for analyzing and mining massive datasets, graphs and social network data. Topics include the storage, management, processing, and analysis of massive datasets. Students will do hands-on assignments to practice programming in Hadoop and Spark.

**Prerequisite**:

Required Prerequisite(s): N/A

Suggested Prerequisite(s): 1) Java and Python Programming; 2) Data Structures; 3) Analysis of Algorithms; 4) Database Systems; 5) Machine Learning; 6) Data Mining.

We will use Java when discussing Hadoop and Python when discussing Spark. If you have not learnt these two programming languages, please read the suggested books and online courses in the Appendix.

The department will strictly enforce all prerequisites. Students without proper prerequisites will be dropped from the class, without any prior notice, at any time during the semester.

**Course Requirements**: All students should not only learn basic theoretical principles but also accumulate practical hands-on experience. All the students will do assignments and take exams. The graduate students will finish the course projects.

CSC 4760: The undergraduate students do not need to do a course project.

CSC 6760: One or two graduate students need to form a group. Each group needs to do a project and writes a project report, which is in the research paper style. Each group will give a final presentation. If one group contains two graduate students, both of them need to present their parts of work in the final presentation.

Six assignments will be given. All of them are programming assignments. Java and Python programming are required. The source code needs to be submitted and should not contain any bugs. All the assignment reports need to be written in a text editor such as MS Word. Handwritten version is not acceptable. The students are not allowed to write the homework by hand, scan the papers or take pictures of the papers, and submit the images of handwritten contents. This is because the handwritten contents are hard to be recognized by the TA. All the assignments need to be submitted through iCollege.

The problems in Exam 1, Exam 2, and assignments may be different for CSC 4760 and CSC 6760.

**Grading Policy:**

**CSC 4760 Undergraduate Students**

|  |  |
| --- | --- |
| **Parts** | **Percentage** |
| Exam 1 | 15% |
| Exam 2 | 15% |
| Assignment 1 | 10% |
| Assignment 2 | 10% |
| Assignment 3 | 11% |
| Assignment 4 | 11% |
| Assignment 5 | 11% |
| Assignment 6 | 11% |
| Attendance | 6% |

The total percentage for Assignments is 64%.

**CSC 4760 Undergraduate Students**

|  |  |  |
| --- | --- | --- |
| A+ [97, 100] | A [93, 97) | A- [90, 93) |
| B+ [87, 90) | B [83, 87) | B- [80, 83) |
| C+ [77, 80) | C [70, 77) |  |
| D [60, 70) | F [0, 60) |  |

The intervals for the final grade may be changed depending on the distribution of the scores. For example, the interval for A might be changed to [95.2, 89.5) based on the distribution of the scores. The boundary score can only be decreased when we adjust the intervals. For example, the new boundary score for A+ may be 95.2, which should be less than 97. If one student’s score is no less than 97, an A+ will definitely be given.

**CSC 6760 Graduate Students**

|  |  |
| --- | --- |
| **Parts** | **Percentage** |
| Exam 1 | 10% |
| Exam 2 | 10% |
| Assignment 1 | 8% |
| Assignment 2 | 8% |
| Assignment 3 | 9% |
| Assignment 4 | 9% |
| Assignment 5 | 9% |
| Assignment 6 | 9% |
| Course Project | 22% |
| Attendance | 6% |

The total percentage for Assignments is 52%.

**CSC 6760 Graduate Students**

|  |  |  |
| --- | --- | --- |
| A+ [97, 100] | A [93, 97) | A- [90, 93) |
| B+ [87, 90) | B [80, 87) |  |
| C+ [77, 80) | C [73, 77) | C- [70, 73) |
| D [60, 70) | F [0, 60) |  |

Similar to the strategy described above, the intervals for the final grade may be changed depending on the distribution of the scores.

**Class Policy:**

* **Attendance**: Students are required to attend all classes. During each class, the student needs to sign the roster sheet. The roster sheets will be kept by the instructor. By the end of the semester, the TA will count the number of absence and give an attendance score for each student. The first two classes (Jan. 13 and Jan. 15) are not counted since they are before the end date of the “late registration and add/drop”.

For undergraduate and graduate students, the attendance score is judged by following the table.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Absence times | < 3 | 3 - 4 | 5 - 6 | 7 - 8 | 9 - 10 | 11 - 12 | > 12 |
| Attendance score | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

* **Academic honesty**: Plagiarism will result in a score of zero on the exam, assignment, or project. Plagiarism will be reported to the Dean of College of Arts and Sciences. The instructor has the right to make a decision.
* **Assignments and Projects**: They are needed to be handed in on time. Late submissions will get 20% penalty. Late submissions will not be accepted if they are submitted more than 7 days after the due date. Due to COVID-19, we will re-consider the late submissions if the students and their family are experiencing difficulties in finance or health, according to the university’s new policy.
* **Withdrawals**: March 3 Tuesday is the last day to withdraw and possibly receive a W. Due to COVID-19, the withdrawal date might have been extended. Please check the university’s official website.
* **Make-ups**: need the instructor's special permission.

**Programming Assignments**

This course will focus on writing code to solve various problems, so all assignments will be programming assignments.

Small datasets will be provided for each assignment so that you will not consume too much computing resource (time and space) while developing your solution. Some assignments will also offer a large dataset so that you can measure how your map-reduce solution scales with the dataset size and the computing resources available.

You are free to discuss approaches to solving the assigned problems with your classmates, but each student is expected to write their own code. Source code must be submitted for each assignment, in addition to the results you obtained when running your program against the datasets provided. If duplicate work is detected, all parties involved will be penalized.

Each assignment is usually due TWO weeks after it is given. The TA usually takes one week to grade the homework. The scores and solutions will be posted in iCollege once the TA finishes the grading job.

**Exams**

We have two exams. The students should complete the exams individually and separately. The student can refer to textbooks and slides. The student can also use internet, online course materials, and computers during the exam. The students should not discuss the problems either online or in person in the exams. The students should not form groups to finish the exams. Plagiarism will result in a score of zero and will be reported to the Dean.

The problems in Exam 1 and Exam 2 may be different for CSC 4760 and CSC 6760.

Exam 1 will be happening during one class before mid-point. Exam 2 will be in the final exam week. The scores and solutions will be posted in iCollege once the TA finishes the grading job.

Late submissions of the exams will get 20% penalty. Late submissions for exams will not be accepted if they are submitted more than 3 days after the exam date.

**Course Project (Only for Graduate Students):**

In the course project, the students need to identify an interesting data processing problem, figure out the solution in a computer cluster, implement the algorithm/idea, and obtain some experimental results. The course project must be related to the course. That means the course project must be related to MapReduce, Hadoop, Spark, Pig, Hive, HBase, or other big data programming tools.

The course project is only for the graduate students.

Each graduate group/student submits one course project including:

1) Proposal (more than 2 pages);

2) Final report (more than 4 pages);

3) Software/source code, user manual, and sample dataset;

4) Slides (more than 12 slides);

The proposal and final report must strictly follow the ACM SIG Templates.

<https://www.acm.org/publications/proceedings-template-16dec2016>

The proposal should be more than 2 pages. The final report should be more than 4 pages. The proposal should focus more on the problem description, the motivation of your project, and the plan of next steps to finish the project. The proposal may also contain some preliminary results. The final report should extend the proposal and contains the details of the implementation of each step and the experimental results. They may contain the following sections.

1) Title

2) Abstract. Briefly describe the motivation, the novel idea, the technical overview, and the novel contributions of the project.

3) Introduction. This section may include the background, the application domain, the motivation of the project, the plan or overview of the project, and the summary of the novel contributions.

4) Related work. This section may provide a broad survey of the existing works that are related to this project. This section should summarize the existing works and point out the relationship between the existing works and this project. The limitations of the existing works may also be described.

5) Problem Formulation. This section may formulate the real-world problem in real-life applications into a computational problem. The input, the output, and the function or purpose of the algorithm may be described.

6) Algorithm Design. This section may provide a detailed description of the MapReduce/Spark/Pig/etc algorithms. For example, the flow-chart maybe used to describe the algorithm. This section may also provide the analysis of the algorithms such as the computational cost and the communication cost at each stage.

7) Experimental Results. In this section, you may describe the information of the input dataset. You may also describe how the algorithm in Section 6) is implemented. You need to provide the experimental results about the effectiveness/accuracy and efficiency/running-time evaluations. Figures or tables should be used to visualize the results. You also need to explain the results and describe what you observe from the results and what you expected to observe.

This is a sample structure of the proposal and report. The actual sections in your project may be significantly different than the above ones. You may design the overall logic of the presentation in your proposal and report.

All the materials need to be submitted to iCollege. Each presentation will be 12 - 20 minutes. This might vary depending on the total number of groups. The following table shows the percentage of each part in the final score for the course project.

|  |  |
| --- | --- |
| **Parts** | **Percentage** |
| Proposal | 15% |
| Final Report | 40% |
| Software/source code, user manual, and sample dataset | 20% |
| Slides and Presentation | 25% |

**Tentative Course Outline and Schedule:**

| **Dates** | **Classes** | **Contents** |
| --- | --- | --- |
| Jan. 13 | Class 1 | Syllabus, Introduction to Big Data (Classes Begin) |
| Jan. 15 | Class 2 | MapReduce Design Patterns, Ch. 1 MapReduce/WordCount |
| Jan. 17 | -- | End of Student late registration and add/drop |
| Jan. 20 | No Class | Holiday (MLK) |
| Jan. 22 | Class 3 | Introduction to Linux commands; Installing Hadoop in Ubuntu OS |
| Jan. 27 | Class 4 | MapReduce Design Patterns, Ch. 2 Summarization Patterns, Min Max Count, Average, Inverted Index |
| Jan. 29 | Class 5 | MapReduce Design Patterns, Ch. 3 Filtering Patterns, Distributed Grep, Top-K Pattern, Distinct Pattern |
| Feb. 3 | Class 6 | MapReduce Design Patterns, Ch. 4 Data Organization Patterns, MapReduce Types, Partitioner, Total Order Sorting, Two Mappers, Binning, Shuffling |
| Feb. 5 | Class 7 | MapReduce Design Patterns, Ch. 5 Join Patterns, Reduce Side Join, Replicated Join, Composite Join |
| Feb. 10 | Class 8 | MapReduce Design Patterns, Ch. 6 Metapatterns |
| Feb. 12 | Class 9 | MapReduce Design Patterns, Ch. 7 Input and Output Patterns |
| Feb. 16 | -- | Project Proposal Due (11:59 pm eastern time, Feb. 16) |
| Feb. 17 | Class 10 | Invited Talk: HPCC Systems – Big Data Platform and Use Cases  by LexisNexis |
| Feb. 19 | Class 11 | Installing Spark in Ubuntu OS  Learning Spark, Ch. 1 Introduction to Spark |
| Feb. 24 | Class 12 | Learning Spark, Ch. 2 Getting Started |
| Feb. 26 | Class 13 | Exam 1 |
| March 2 | Class 14 | Learning Spark, Ch. 3 Programming with RDDs |
| March 3 | -- | Midpoint (the last day to withdraw) |
| March 4 | Class 15 | Learning Spark, Ch. 4 Key Value Pairs |
| March 9 | Class 16 | Learning Spark, Ch. 5 Loading and Saving Data |
| March 11 | Class 17 | Learning Spark, Ch. 6 Advanced Spark Programming |
| March 16 | No Class | Spring Break |
| March 18 | No Class | Spring Break |
| March 23 | No Class | Class cancelled because of COVID-19 |
| March 25 | No Class | Class cancelled because of COVID-19 |
| March 30 | Class 18 | Learning Spark, Ch. 7 Running on a Cluster |
| April 1 | Class 19 | Learning Spark, Ch. 8 Tuning and Debugging Spark |
| April 6 | Class 20 | Selected Topics: Spark Machine Learning Library (spam email detection) |
| April 8 | Class 21 | Selected Topics: Spark Streaming (tweet streaming) |
| April 13 | Class 22 | Selected Topics: Spark GraphX (PageRank, connected components, triangle counting, property graph) |
| April 15 | Class 23 | Selected Topics: Pig |
| April 20 | Class 24 | Selected Topics: Hive, HBase |
| April 22 | Class 25 | Project Presentations 1 (6 groups) |
| April 27 | Class 26 | Project Presentations 2 (6 groups)  (Classes End) |
| April 27 | -- | Course Project Due (11:59 pm eastern time, April 27) |
| April 28 – May 5 | -- | Final Exams Period. Final Exam schedule:  https://registrar.gsu.edu/atlanta-campus-spring-2020-final-exam-schedule/  We will have the final exam on 13:30-16:00, May 4. |
| May 4 | 13:30-16:00 | Exam 2 (Open slides, textbook, and internet; but no chats)  We will release the word doc version of Exam 2 in iCollege by 2:30pm, 05/04/2020. Students need to type in the solutions into the word doc file and submit the solution into iCollege. As a backup, we will create a Google Drive folder. If you cannot submit it into iCollege, you can upload the file into the Google Drive folder. |
| May 7 | -- | Grades Due at 5 pm eastern time, May 7 |

This is a tentative course schedule. The contents taught in classes may be significantly different from those listed in the table. Research papers may be presented in the classes. Thus the contents in the research papers may be significantly different from those in the two textbooks.

**Statement**: This course syllabus provides a general plan for the course; deviations may be necessary. All the website links are successfully accessed by Jan. 12, 2020.

**Appendix**

**Online Resources:**

1. Apache Hadoop: <http://hadoop.apache.org/>

2. Apache Spark: <https://spark.apache.org/>

3. Mastering Apache® Spark™ 2.0. http://go.databricks.com/mastering-apache-spark-2.0

4. HPCC Systems from LexisNexis: <https://hpccsystems.com/>

**Related Courses:**

1. Coursera (<https://www.coursera.org>):

Data Manipulation at Scale: Systems and Algorithms. Bill Howe. University of Washington.

<https://www.coursera.org/learn/data-manipulation/home/welcome>

Pig stuff in the above course: <https://www.coursera.org/learn/data-manipulation/lecture/2bKBv/almost-sql-pig>

Introduction to Apache Spark and AWS

<https://www.coursera.org/learn/bigdata-cluster-apache-spark-and-aws>

Big Data Specialization

<https://www.coursera.org/specializations/big-data>

Big Data Analysis with Scala and Spark

<https://www.coursera.org/learn/scala-spark-big-data>

Big Data Essentials: HDFS, MapReduce and Spark RDD

<https://www.coursera.org/learn/big-data-essentials>

Big Data Analysis: Hive, Spark SQL, DataFrames and GraphFrames

<https://www.coursera.org/learn/big-data-analysis>

Google Cloud Platform Big Data and Machine Learning Fundamentals

<https://www.coursera.org/learn/gcp-big-data-ml-fundamentals>

2. Cognitive Class: (<https://cognitiveclass.ai/>)

Big Data 101

<https://cognitiveclass.ai/courses/what-is-big-data/>

Data Science 101

<https://cognitiveclass.ai/courses/data-science-101/>

MapReduce and YARN

<https://cognitiveclass.ai/courses/mapreduce-and-yarn/>

Hadoop 101

<https://cognitiveclass.ai/courses/introduction-to-hadoop/>

Spark Fundamentals I

<https://cognitiveclass.ai/courses/what-is-spark/>

Spark Fundamentals II

<https://cognitiveclass.ai/courses/spark-rdd/>

Spark MLlib

<https://cognitiveclass.ai/courses/spark-mllib/>

Exploring Spark’s GraphX

<https://cognitiveclass.ai/courses/spark-graphx/>

Apache Pig 101. Warren Pettit, Daniel Tran, Leons Petrazickis. IBM. Big Data University.

<https://cognitiveclass.ai/courses/introduction-to-pig/>

3. Introduction to MapReduce Programming (A free five-hour crash-course on MapReduce) :

<https://archive.bigdatauniversity.com/courses/introduction-to-mapreduce-programming/>

4. Intro to Hadoop and MapReduce (by Cloudera). Another (longer) introduction to Hadoop and Map Reduce (FREE):

<https://www.udacity.com/course/intro-to-hadoop-and-mapreduce--ud617>

5. Hadoop Tutorial: Developing Big-Data Applications with Apache Hadoop

<http://www.coreservlets.com/hadoop-tutorial/>

This is a great Hadoop tutorial.

6. Big Data Analytics with HDInsight: Hadoop on Azure

<https://mva.microsoft.com/en-US/training-courses/big-data-analytics-with-hdinsight-hadoop-on-azure-10551?l=KBcyOu97_7604984382>

7. Learning Apache Hadoop EcoSystem - Hive (Free Hive course):

<https://www.udemy.com/learning-apache-hive/>

8. Free big data tutorials: <https://www.dezyre.com/tutorial>

9. CS378 - Big Data Programming - Fall 2017 - David Franke. All lecture notes are publicly available.

<https://www.cs.utexas.edu/~dfranke/courses/2017fall/cs378-BDP.htm>

10. HPCC Systems Online Courses: <https://hpccsystems.com/training#Classes> . Each student can get a coupon for free access of the online courses.

**Cloud Service Providers:**

1. Amazon EMR (Elastic MapReduce): <https://aws.amazon.com/emr/>

2. Google Cloud Platform: <https://cloud.google.com/>

3. Microsoft Azure Cloud Computing Platform & Services: <https://azure.microsoft.com/en-us/>

4. IBM Cloud (BlueMix): <https://www.ibm.com/cloud/>

**Further Reading**

Here are references to further reading. Graduate students are encouraged to read them.

* *MapReduce: Simplified Data Processing on Large Clusters*, by Jeffry Dean and Sanjay Ghemawat, 2004.
* *A Comparison of Join Algorithms for Log Processing in MapReduce*, by Spyros Blanas, Jignesh M. Patel, Vuk Ercegovac, Jun Rao, Eugene J. Shekita, and Yuanyuan Tian, 2010.
* *The Family of MapReduce and Large-Scale Data Processing Systems*, by Sherif Sakr, Anna Liu, and Ayman G. Fayoumi, 2013.
* MapReduce Algorithms for Big Data Analysis, by Kyuseok Shim, 2012.
* The Google file system, By Sanjay Ghemawat, et al., 2003.
* *Spark: Cluster Computing with Working Sets*, by Matei Zaharia, Mosharaf Chowdhury, Micheal J. Franklin, Scott Shenker, and Ion Stoica, 2010.
* SparkR: Scaling R Programs with Spark, Shivaram Venkataraman, Zongheng Yang, Davies Liu, Eric Liang, Hossein Falaki, Xiangrui Meng, Reynold Xin, Ali Ghodsi, Michael Franklin, Ion Stoica, and Matei Zaharia. SIGMOD 2016. June 2016.
* MLlib: Machine Learning in Apache Spark, Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, Doris Xin, Reynold Xin, Michael J. Franklin, Reza Zadeh, Matei Zaharia, and Ameet Talwalkar. Journal of Machine Learning Research (JMLR). 2016.
* Spark SQL: Relational Data Processing in Spark. Michael Armbrust, Reynold S. Xin, Cheng Lian, Yin Huai, Davies Liu, Joseph K. Bradley, Xiangrui Meng, Tomer Kaftan, Michael J. Franklin, Ali Ghodsi, Matei Zaharia. SIGMOD 2015.
* GraphX: Unifying Data-Parallel and Graph-Parallel Analytics. Reynold S. Xin, Daniel Crankshaw, Ankur Dave, Joseph E. Gonzalez, Michael J. Franklin, Ion Stoica. OSDI 2014.
* Discretized Streams: Fault-Tolerant Streaming Computation at Scale. Matei Zaharia, Tathagata Das, Haoyuan Li, Timothy Hunter, Scott Shenker, Ion Stoica. SOSP 2013.
* Shark: SQL and Rich Analytics at Scale. Reynold S. Xin, Joshua Rosen, Matei Zaharia, Michael J. Franklin, Scott Shenker, Ion Stoica. SIGMOD 2013.
* Discretized Streams: An Efficient and Fault-Tolerant Model for Stream Processing on Large Clusters. Matei Zaharia, Tathagata Das, Haoyuan Li, Scott Shenker, Ion Stoica. HotCloud 2012.
* Shark: Fast Data Analysis Using Coarse-grained Distributed Memory (demo). Cliff Engle, Antonio Lupher, Reynold S. Xin, Matei Zaharia, Haoyuan Li, Scott Shenker, Ion Stoica. SIGMOD 2012. Best Demo Award.
* Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, Ion Stoica. NSDI 2012. Best Paper Award.

**Java and Python Programming Courses at GSU:**

1) CSc 1302 Principles of Computer Science II (Java)

2) CSc 2302/7352 Computer Programming for Non-Majors II; Data Structure for Bioinformatics (Python)

**Recommended books for Java and Python Programming:**

1) Building Java Programs: A Back to Basics Approach, 4th edition. by Stuart Reges and Marty Stepp. There are slides and other learning materials available online. <http://www.buildingjavaprograms.com/>

2) Java: A Beginner's Guide, Seventh Edition. By Herbert Schildt. 2017.

3) Introduction to Computing Using Python: An Application Development Focus. Ljubomir Perkovic, 2011.

4) Python Crash Course: A Hands-On, Project-Based Introduction to Programming. By Eric Matthes. 2015.

**Suggested online courses for Java and Python Programming:**

1) Java Tutorial for Complete Beginners. <https://www.udemy.com/java-tutorial/>

2) Learn Java. <https://www.codecademy.com/learn/learn-java>

3) Programming Foundations with Python. <https://www.udacity.com/course/programming-foundations-with-python--ud036>

4) Learn Python. <https://www.codecademy.com/learn/learn-python>

**Suggested online courses for Machine Learning and Deep Learning:**

1) Intel Online Courses – Machine Learning 101

<https://software.intel.com/en-us/ai-academy/students/kits/machine-learning-101>

2) Intel Online Courses – Deep Learning 101

<https://software.intel.com/en-us/ai-academy/students/kits/deep-learning-101>

3) Introduction to BigDL on Apache Spark\* Part1

<https://software.intel.com/en-us/videos/introduction-to-bigdl-on-apache-spark-part1>