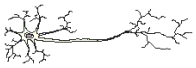

HMW #1

1. Read Lesson 19, Chapter 20.
2. Do check point exercises 20.1-20.22, 20.26-20.34.
3. Do programming exercises 20.1, 20.3, 20.6, 20.20.
4. Visit and evaluate: https://www.geeksforgeeks.org/collections-in-java-2/. Worth it?
5. Visit and evaluate: http://tutorials.jenkov.com/java-collections/queue.html. What did you find?

HMW #2

1. Read Lesson 19, Read Chapter 24.
2. Do check point exercises 24.1-24.23.
3. Do programming exercises 24.1, 24.4, 24.14
4. Visit and evaluate: https://www.callicoder.com/java-stack/ Anything interesting?
5. Visit and evaluate: https://www.softwaretestinghelp.com/java-priority-queue-tutorial/. Learn anything?

HMW #3

1. Read Lesson 20, Read Chapter 21.
2. Do check point exercises 21.1-21.25
3. Do programming exercises 21.1, 21.4, 21.12,
4. Visit and evaluate: https://www.edureka.co/blog/sets-in-java/. Why are sets important?
5. Visit and evaluate: https://www.youtube.com/watch?v=PeFyhRr42ac. Too simple video?

HMW #4

1. Read Lesson 20, Read Chapter 25.
2. Do check point exercises 25.1-25.25.
3. Do programming exercises 25.1, 25.4, 25.12,
4. Visit and evaluate: https://www.java67.com/2014/01/when-to-use-linkedhashset-vs-treeset-vs-hashset-java.html. Why are sets important?
5. Visit and evaluate: https://www.youtube.com/watch?v=fZgO3R3V_jw. Good video?

HMW #5

1. Read Lesson 13, Read Chapter 12.
2. Do check point exercises 12.1-12.16.
3. Do programming exercises 12.1-12.2, 12.5.
4. Visit and evaluate: http://docs.oracle.com/javase/tutorial/essential/io/index.html. Anything exciting?
5. Visit and evaluate: https://www.tutorialspoint.com/java/java_files_io.htm. What did you find?

***12.7** (*NumberFormatException*) Write the `bin2Dec(String binaryString)` method to convert a binary string into a decimal number. Implement the `bin2Dec` method to throw a `NumberFormatException` if the string is not a binary string.

***12.8** (*HexFormatException*) Exercise 12.6 implements the `hex2Dec` method to throw a `NumberFormatException` if the string is not a hex string. Define a custom exception called `HexFormatException`. Implement the `hex2Dec` method to throw a `HexFormatException` if the string is not a hex string.

***12.9** (*BinaryFormatException*) Exercise 12.7 implements the `bin2Dec` method to throw a `BinaryFormatException` if the string is not a binary string. Define a custom exception called `BinaryFormatException`. Implement the `bin2Dec` method to throw a `BinaryFormatException` if the string is not a binary string.

HMW #6

1. Read Lesson 13, Read Chapter 12.
2. Do check point exercises 12.17-12.25.
3. Do programming exercises 12.8, 12.12, 12.14.
4. Visit and evaluate: http://www.tutorialspoint.com/java/java_files_io.htm. Worth the visit?
5. Visit and evaluate: https://examples.javacodegeeks.com/java-io-ioexception/. What's there?

12.36 Suppose you enter `45 57.8 789`, then press the *Enter* key. Show the contents of the variables after the following code is executed.

```
Scanner input = new Scanner(System.in);
int intValue = input.nextInt();
double doubleValue = input.nextDouble();
String line = input.nextLine();
```

HMW #7

1. Read Lesson 13, Read Chapter 12.
2. Do programming exercises 12.19, 12.24, 12.30.
3. Visit and evaluate: http://en. https://www.infoworld.com/article/2078654/java-se-five-ways-to-maximize-java-nio-and-nio-2.html. What did you find?
4. Visit and evaluate: https://www.marcobehler.com/guides/java-files. What did you learn about Path and Path.of?

12.37 Suppose you enter `45`, press the *Enter* key, `57.8`, press the *Enter* key, `789`, and press the *Enter* key. Show the contents of the variables after the following code is executed.

```
Scanner input = new Scanner(System.in);
int intValue = input.nextInt();
double doubleValue = input.nextDouble();
String line = input.nextLine();
```

HMW #8

1. Read Lesson 13, Read Chapter 12 & 19.
2. Do check point exercises 19.1-19.13.
3. Do programming exercises 19.1, 19.4, 19.6.
4. Visit and evaluate: https://docs.oracle.com/javase/tutorial/java/generics/index.html. Anything there?
5. Visit and evaluate: https://www.youtube.com/watch?v=XMvznsY02Mk. Worth the visit?

HMW #9

1. Read Lesson 13, Read Chapter 12 & 19.
2. Do check point exercises 19.14-19.20.
3. Do programming exercises 19.2, 19.5, 19.9.
4. Visit and evaluate: https://objectcomputing.com/resources/publications/sett/july-2003-generics-in-java. Interesting?
5. Visit and evaluate: https://www.programiz.com/java-programming/generics. What's there?

HMW #10

1. Read Lesson 14, Read Chapter 17.
2. Do check point exercises 17.1-17.13.
3. Do programming exercises 17.1, 17.5, 17.8.
4. Visit and evaluate: https://www.codejava.net/java-se/file-io/how-to-read-and-write-binary-files-in-java. What did you find?
5. Visit and evaluate: http://www.javapractices.com/topic/TopicAction.do?Id=245. Worth the visit?

HMW #11

1. Read Lesson 14, Read Chapter 17.
2. Do check point exercises 17.14-17.23.
3. Do programming exercises 17.2, 17.4, 17.9.
4. Visit and evaluate: https://www.informit.com/articles/article.aspx?p=2955144&seqNum=2. Fruitful?
5. Visit and evaluate: https://www.youtube.com/watch?v=-if77wPQKhg. Can you stomach the video?

HMW #12

1. Read Lesson 14, Read Chapter 17.
2. Do programming exercises 17.14-17.15.
3. Visit and evaluate: https://docs.oracle.com/javase/tutorial/essential/io/rafs.html. Worth it?
4. Visit and evaluate: https://www.codejava.net/java-se/file-io/java-io-how-to-use-randomaccess-file-java-io-package. What did you find?

HMW #13

1. Read Lesson 18 and Chapter 27.
2. Do check point exercises 27.1-27.14.
3. Do programming exercises 27.1, 27.2, 27.4.
4. Visit and evaluate: https://beginnersbook.com/2014/07/hashtable-in-java-with-example/. Did you learn anything from it?
5. Visit and evaluate: https://www.youtube.com/watch?v=LTAAn97QBH8. What's there?

HMW #14

1. Read Lesson 18 and Chapter 27.
2. Do check point exercises 27.15-27.20.
3. Do programming exercises 27.3, 27.5, 27.7.
4. Visit and evaluate: https://www.geeksforgeeks.org/hashing-in-java/. Anything interesting?
5. Visit and evaluate: https://algs4.cs.princeton.edu/34hash/. What's there?

HMW #15

1. Read Lesson 18 and Chapter 32.
2. Do check point exercises 32.1-32.9.
3. Do programming exercises 32.1-32.3.
4. Visit and evaluate: https://www.javatpoint.com/java-jdbc#:~:text=JDBC%20stands%20for%20Java%20Database,to%20connect%20with%20the%20database. How was it?
5. Visit and evaluate: https://docs.oracle.com/javase/tutorial/jdbc/basics/index.html. Learn anything?

HMW #16

1. Read Lesson 18 and Chapter 32.
2. Do check point exercises 32.10-32.15.
3. Do programming exercises 32.4-32.5.
4. Visit and evaluate: https://www.youtube.com/watch?v=3OrEsC-QjUA. Interesting?
5. Visit and evaluate: https://www.geeksforgeeks.org/establishing-jdbc-connection-in-java/. Anything new?

HMW #17

1. Read Lesson 18 and Chapter 32.
2. Do check point exercises 32.16-32.20.
3. Do programming exercises 32.6-32.7.
4. Visit and evaluate: https://www.youtube.com/watch?v=Cq4lwVE2Fzk. What's there?

5. Visit and evaluate: https://dev.mysql.com/doc/connector-j/5.1/en/connector-j-usagenotes-connect-drivermanager.html. Worth the visit?

HMW #18

1. Read Lesson 18 and Chapter 32.
2. Do programming exercises 32.8-32.9.
3. Visit and evaluate: https://www.studytonight.com/java/introduction-to-jdbc.php. How was it?
4. Visit and evaluate: https://techvidvan.com/tutorials/jdbc-tutorial/. Learn anything?

HMW #19

1. Read Lesson 18 and Chapter 32.
2. Do programming exercises 32.10-32.11.
3. Visit and evaluate: https://alvinalexander.com/java/edu/pj/jdbc/jdbc0003/. How was it?
4. Visit and evaluate: https://www.codejava.net/java-se/jdbc/connect-to-mysql-database-via-jdbc. What's there?

HMW #20

1. Read Lesson 28 and Chapter 28.
2. Do check point exercises 28.1-28.9.
3. Do programming exercises 28.1-28.3.
4. Visit and evaluate: https://www.tutorialspoint.com/jdbc/jdbc-create-database.htm. Any good?
5. Visit and evaluate: https://www3.ntu.edu.sg/home/ehchua/programming/java/JDBC_Basic.html .What's there?

HMW #21

1. Read Lesson 28 and Chapter 28.
2. Do check point exercises 28.10-28.15.
3. Do programming exercises 28.4-28.6.
4. Visit and evaluate: https://www.baeldung.com/java-graphs. Who developed MariaDB?
5. Visit and evaluate: https://www.geeksforgeeks.org/implementing-generic-graph-in-java/. What did you learn?

HMW #22

1. Read Lesson 28 and Chapter 28.
2. Do check point exercises 28.16-28.20.
3. Do programming exercises 28.7-28.9.
4. Visit and evaluate: https://www.geeksforgeeks.org/implementing-generic-graph-in-java/. How was it?
5. Visit and evaluate: https://medium.com/@mithratalluri/basic-graph-implementation-in-java-9ed12e328c57. What was there?

HMW #23

1. Read Lesson 28 and Chapter 28.
2. Do check point exercises 28.21-28.25.
3. Do programming exercises 28.10-28.12.
4. Visit and evaluate: https://algorithms.tutorialhorizon.com/implement-graph-using-map-java/. Any good?
5. Visit and evaluate: http://web.cecs.pdx.edu/~sheard/course/Cs163/Doc/Graphs.html. Worth the visit?

HMW #24

1. Read Lesson 28 and Chapter 28.
2. Do programming exercises 28.13-28.15.
3. Visit and evaluate: https://www.hackerearth.com/practice/algorithms/graphs/graph-representation/tutorial/. Interesting?
4. Visit and evaluate: https://www.journaldev.com/44036/adjacency-list. How was it?

HMW #25

1. Read Lesson 28 and Chapter 28.
2. Do programming exercises 28.16-28.18.
3. Visit and evaluate: https://algorithms.tutorialhorizon.com/graph-implementation-adjacency-list-better-set-2/. What's there?
4. Visit and evaluate: https://www.genuinecoder.com/add-javafx-charts-graphs-tutorial-html/. Have you done any JavaFX?

HMW #26

1. Read Lesson 28 and Chapter 28.
2. Do programming exercises 28.19-28.21
3. Visit and evaluate: https://javatutorial.net/graphs-java-example. Learned anything?
4. Visit and evaluate: https://opendatastructures.org/ods-java/12_1_AdjacencyMatrix_Repres.html. How was it?

HMW #27

1. Read Lesson 29 and Chapter 29.
2. Do check point exercises 29.1-29.8.
3. Do programming exercises 29.1-29.3.
4. Visit and evaluate: http://www.mathcs.emory.edu/~cheung/Courses/171/Syllabus/11-Graph/weighted.html. Did you learn anything from it?
5. Visit and evaluate: https://algorithms.tutorialhorizon.com/weighted-graph-implementation-java/. What did you find?

HMW #28

1. Read Lesson 29 and Chapter 29.
2. Do check point exercises 29.9-29.15.
3. Do programming exercises 29.4-24.6.
4. Visit and evaluate: https://stackabuse.com/graphs-in-java-representing-graphs-in-code/. What is there?
5. Visit and evaluate: http://users.monash.edu/~lloyd/tildeAlgDS/Graph/. Interesting?

HMW #29

1. Read Lesson 29 and Chapter 29.
2. Do check point exercises 29.16-29.28.
3. Do programming exercises 29.7-29.9.
4. Visit and evaluate: https://stackabuse.com/graphs-in-java-representing-graphs-in-code/. How was it?
5. Visit and evaluate: https://support.csis.pace.edu/CSISWeb/docs/techReports/techReport225.pdf. Any good?
6. For the code WeightedEdge edge = new WeightedEdge(1, 2, 3.5); what is edge.u, edge.v, and edge.w?
7. What is the output of the following code?

```
List<WeightedEdge> list = new ArrayList<>();
List.add(new WeightedEdge(1, 2, 3.5));
List.add(new WeightedEdge(2, 3, 4.5));
WeightedEdge e = java.util,Collections.max(list);
System.out.println(e.u);
System.out.println(e.v);
System.out.println(e.weight);
```

HMW #30

1. Read Lesson 29 and Chapter 29.
2. Do programming exercises 29.10-29.12.
3. Visit and evaluate: https://www.programcreek.com/java-api-examples/?api=org.jgrapht.WeightedGraph. Did you learn anything from it?
4. Visit and evaluate: https://hellokoding.com/graph-data-structure/. Worth the visit?
5. If a priority queue is used to store weighted edges, what is the output of the following code?

```
PriorityQueue<WeightedEdge> q = new PriorityQueue<>();
q.offer(new WeightedEdge(1, 2, 3.5));
q.offer(new WeightedEdge(1, 6, 6.5));
q.offer(new WeightedEdge(1, 7, 1.5));
System.out.println(q.poll().weight);
System.out.println(q.poll().weight);
System.out.println(q.poll().weight);
```

6. Is a minimum spanning tree unique if all edges have different weights?

HMW #31

1. Read Lesson 29 and Chapter 29.
2. Do programming exercises 29.13-29.16.

3. Visit and evaluate: https://www.tutorialspoint.com/java/java_multithreading.htm. What's there?
4. Visit and evaluate: https://www.geeksforgeeks.org/multithreading-in-java/. What did you learn?

HMW #32

1. Read Lesson 15 and Chapter 30.
2. Do check point exercises 30.1-30.8.
3. Do programming exercises 30.1-30.3.
4. Visit and evaluate: https://beginnersbook.com/2013/03/multithreading-in-java/. Learned anything?
5. Visit and evaluate: https://www.youtube.com/watch?v=BWjsk2S_sZ0. How did it go?

HMW #33

1. Read Lesson 15 and Chapter 30.
2. Do check point exercises 30.9-30.15.
3. Do programming exercises 30.4-30.6.
4. Visit and evaluate: https://dzone.com/articles/java-thread-tutorial-creating-threads-and-multithr. Did you learn anything from it?
5. Visit and evaluate: https://www3.ntu.edu.sg/home/ehchua/programming/java/j5e_multithreading.html. Any good?

HMW #34

1. Read Lesson 15 and Chapter 30.
2. Do check point exercises 30.15-30.25.
3. Do programming exercises 30.7-30.8.
4. Visit and evaluate: https://www.softwaretestinghelp.com/multithreading-in-java/. What's there?
5. Visit and evaluate: . Worth the visit?

HMW #35

1. Read Lesson 15 and Chapter 30.
2. Do programming exercises 30.9-30.10.
3. Visit and evaluate: https://www.techbeamers.com/java-multithreading-with-examples/. How was it?
4. Visit and evaluate: https://www.c-sharpcorner.com/article/a-complete-multithreading-tutorial-in-java/. What did you find?

HMW #36

1. Read Lesson 15 and Chapter 30.
2. Do programming exercises 30.11-30.12.
3. Visit and evaluate: https://cse.iitkgp.ac.in/~dsamanta/java/ch6.htm. Did you learn anything from it?
4. Visit and evaluate: https://www.javaprogramto.com/2020/01/java-matrix-multiplication-threads.html. Any good?

HMW #37

1. Read Lesson 15 and Chapter 30.
2. Do programming exercises 30.13-30.14.
3. Visit and evaluate: https://ducmanhphan.github.io/2019-01-10-How-to-use-collections-in-Java-multithreading/. What did you find?
4. Visit and evaluate: https://javahungry.blogspot.com/2017/10/java-multithreading-interview-questions-and-answers.html. What's there?

