

Assignment 9

Due Wednesday, June 5th, 9 pm

This is the last graded assignment of the quarter. The final exam for section 601 is scheduled for Friday, June 14th, 11:30 am - 1:45 pm. The final exam for section 610 must be taken between Thursday, June 13th and Saturday, June 15th and is scheduled by each student individually through D2L. Please ask if you have any questions.

Reading

Read Section 3.3, 3.4, 3.5, 3.6, 3.7 and 3.9 in **Core Java for the Impatient** by Cay S. Horstmann.

Logistics

In this class programming assignments may be completed in consultation with up to four other classmates. You must identify the classmates with whom you collaborate in a comment at the top of the assignment, and the total number of collaborators on any assignment **may not exceed four people**. You must also submit a comment at the top of each assignment that describes in detail how each collaborator contributed to the assignment. You may not under any circumstances discuss the assignments with classmates other than your identified collaborators. Working so closely with anyone other than your identified collaborators or Dr. Settle, so as to produce identical or near identical code is a violation of the Academic Integrity policy. This policy will be strictly enforced.

Please include the following with your assignment submission:

1. A comment at the top of your Java file identifying any classmates with whom you discussed or in any other way collaborated on the assignment. You may work with **no more than four** other people.
2. Add a comment at the top of your Java file that describes for each person what they contributed to the assignment. This must be at least 2-3 sentences and be very specific. Please make sure to read the Academic Integrity pledge for more details about the types of collaboration that are allowed.

A submission that does not include a list of collaborators and a comment describing each person's contributions when you collaborated with classmates will earn a 0. People who worked alone should put a comment at the top of their file indicating that or they will also earn a 0. There will be no exceptions to this rule.

Remember that **everyone** submitting the assignment must be able to explain **all** of the code submitted, regardless of the type of collaboration that occurred. Anyone submitting code they cannot explain is violating the Academic Integrity policy and will earn a 0 on the assignment.

Assignment

For this assignment you will do two things: **add to the Child class** we considered for [the seventh assignment](#) to include a new accessor and **write a driver program for the new Child class**. Begin the assignment by downloading the solution to the previous assignment in the Child.java file found on D2L.

1. First you should modify the Child class by **writing an accessor for the parent** of the Child object: **public String getParent()**
2. Next you need to write a driver program for the Child class. It will use the Arrays class for printing and sorting. In particular, it must:
 - a. **Have a static constant for the number of Child** objects in the array called MAX.
 - b. **Create an array of MAX Child** objects and fill them with appropriate test values using calls to the default or parameterized constructor of the Child class.
 - c. **Print the filled Child array** using the toString() method of the Arrays class. Do NOT use a loop.
 - d. **Sort the Child objects** using a call to Arrays.sort() that **uses the Comparable interface**.
 - e. **Print the newly sorted Child array** using the toString() method of the Arrays class. Do NOT use a loop.
 - f. **Sort the Child array using a call to Arrays.sort() with a lambda expression**. The lambda expression will sort based on the name of the parent of the Child. This lambda expression must use the parent accessor that you wrote in the first step of the assignment. Strings should be compared using the compareTo() method of the String class.
 - g. **Print the newly sorted Child array** using the toString() method of the Arrays class. Do NOT use a loop.

Below is the sample output that my completed solution produced. You are not required to use my particular Child objects in your solution, but I wanted to give you a concrete example to consider.

The Child array: [Name: Erin, birth year: 2004, parent: Amber Settle, Name: Joon, birth year: 2004, parent: Jeff Elam, Name: Prudence, birth year: 2007, parent: Andre Berthiaume]

The Child array, after sorting (using compareTo): [Name: Prudence, birth year: 2007, parent: Andre Berthiaume, Name: Erin, birth year: 2004, parent: Amber Settle, Name: Joon, birth year: 2004, parent: Jeff Elam]

The Child array, after sorting (using the lambda): [Name: Erin, birth year: 2004, parent: Amber Settle, Name: Prudence, birth year: 2007, parent: Andre Berthiaume, Name: Joon, birth year: 2004, parent: Jeff Elam]

Submitting the assignment

You must submit the assignment using the assignment 9 dropbox on [the D2L site](#). **Submit two Java files (Child.java and ChildDriver.java)** with your solutions found in them. Please don't submit a zip file since that makes my life difficult when grading. I've set up the dropbox to allow you to submit multiple files. Submissions after the deadline listed above will be automatically rejected by the system. See the syllabus for the grading policy.

Grading

The assignment is worth 100 points. The modifications to the Child class is worth 15 points and the Child driver program is worth 85 points.