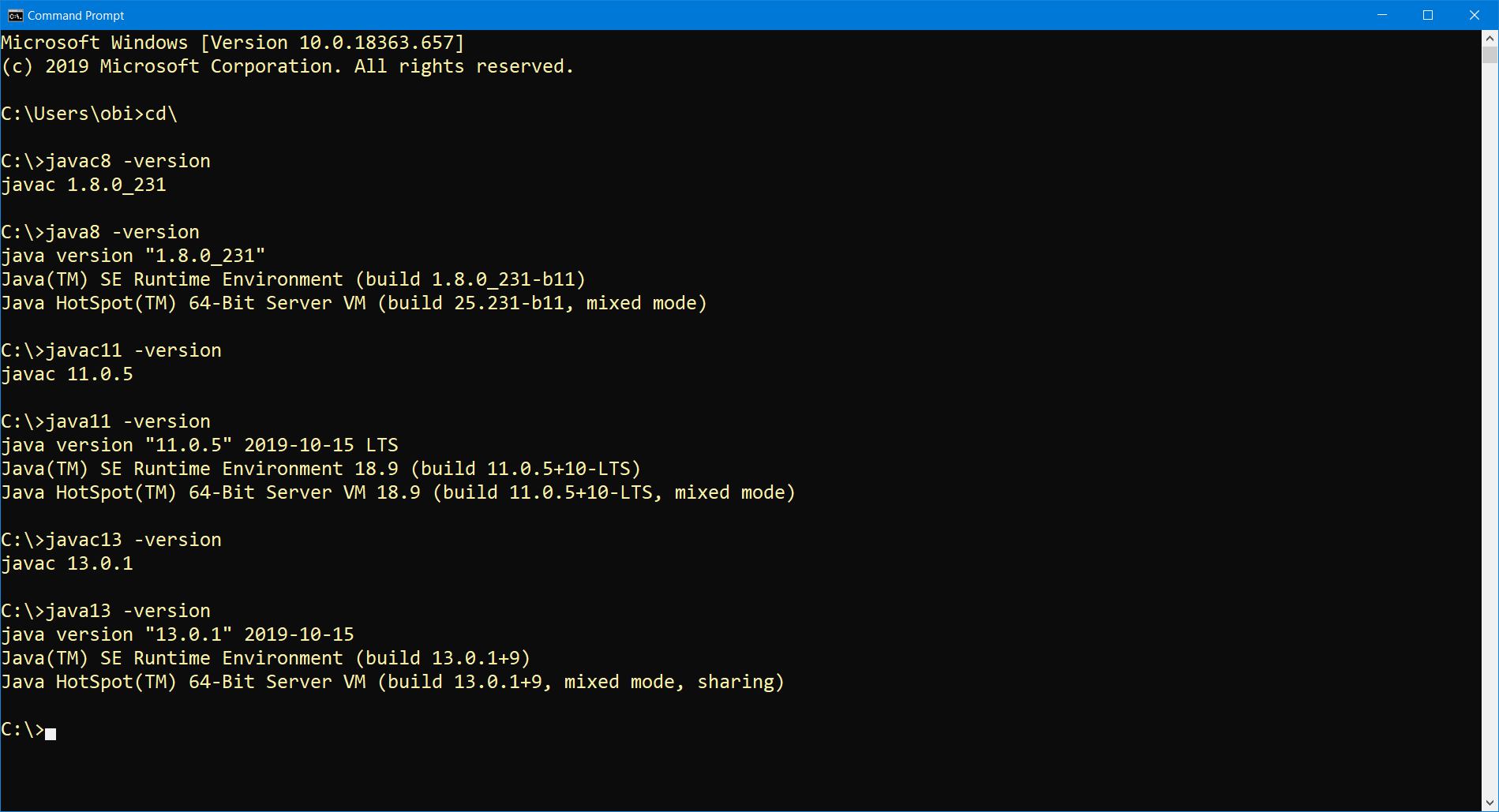
**Setting-up Java Software Development Tools and Environment for CS425-SWE-202003**

1. Java SE Development Kit (JDK):
   1. Go to the Oracle website at <http://www.oracle.com/technetwork/java/javase/downloads/index.html>, obtain and install the latest releases of JDKs for Java SE 8, 11 and 13. You may follow the detailed step-by-step guide provided in the [Setting-up document here](https://docs.google.com/document/d/1BRKW44SlId19otqlIYu7yJbArqn3qqzKfsCA9qq7gD4/edit).
   2. Demonstrate and provide evidence of your having installed the 3 versions of the JDK, by taking screenshots of your Windows Command or OS terminal window, similar to the following:



1. IDE tool:
   1. Go to the Eclipse.org website at <https://www.eclipse.org/downloads/packages/>, obtain the "Eclipse IDE for Enterprise Java Developers" package. If you prefer some other IDE, such as Jetbrains IntelliJ IDEA, Netbeans, Oracle JDeveloper etc., then you are welcome to use it.
2. Using your IDE, implement code solutions for the following:
   1. Assume you have been tasked to build an application to be used in managing the Student records for a University and an excerpt of the students' data is given as follows:

s1: studentId:110001, name:Dave, dateOfAdmission:11/18/1951

s2: studentId:110002, name:Anna, dateOfAdmission:12/07/1990

s3: studentId:110003, name:Erica, dateOfAdmission:01/31/1974

s4: studentId:110004, name:Carlos, dateOfAdmission:08/22/2009

s5: studentId:110005, name:Bob, dateOfAdmission:03/05/1990

... etc.

To accomplish this, you are required to code a class named Student, which has the following data fields (i.e. instance variables) - studentId, name, dateOfAdmission.

Create a Command-Line Java Application and write code for the Student class, including the following:

- Each of the data fields,

- Any 3 constructors including the default constructor, and

- Getter (accessor) and Setter (mutator) methods for the data fields.

- Make the class be inside a package named, edu.mum.cs.cs425.demos.studentrecordsmgmtapp.model.

* 1. In the package named, edu.mum.cs.cs425.demos.studentrecordsmgmtapp, add an executable Java class named, MyStudentRecordsMgmtApp. In the class's main method, write code that creates an array of Students, using the sample data provided above.
  2. In the MyStudentRecordsMgmtApp class, also add a method named, printListOfStudents, which takes as input, the array of students and it iterates through the objects and prints out all the students data to the console/screen. This print-out should be in ascending order of the Students Names. **Note**: Call your printListOfStudents(...) method within your MyStudentRecordsMgmtApp class's main method, then execute it.
  3. Also, in the MyStudentRecordsMgmtApp class, add another method named, getListOfPlatinumAlumniStudents, which takes as input, the array of all students and it returns a List of only PlatinumAlumni students. A PlatinumAlumni student is a student who gained admission into the University at least 30 years ago. **Note**: Call your getListOfPlatinumAlumniStudents(...) method within your MyStudentRecordsMgmtApp class's main method, print the list of the platinum-alumni students, in descending order of their dates of admission, then execute it.
  4. **Further CodingPractice Exercise Problems:**
     1. Write a function (or method) named, printHelloWorld, that takes as input, an array of integers and iterates through them, and it prints the text, "Hello", if the integer is a multiple of 5. It prints the text, "World", if the integer is a multiple of 7. And when it encounters an integer that is a multiple of both 5 and 7, it prints the text, "HelloWorld".
     2. Write code for a method named, findSecondBiggest, which takes as input, an array of integers and finds and returns the second biggest of the integers. For example, findSecondBiggest([1,2,3,4,5]) should return 4. And findSecondBiggest([19,9,11,0,12]) should return 12. (**Note**: Do not use sorting).
  5. Take a screenshot of each of your results as shown within your IDE (or in a command/terminal window) and include it in your submission.

1. Create a git repository for your work in the above tasks, commit and push your finished assignment up to your github account.
2. Make a submission for this Lab assignment 6 in Sakai, by simply including/submitting the url/link of the repo on your github account.