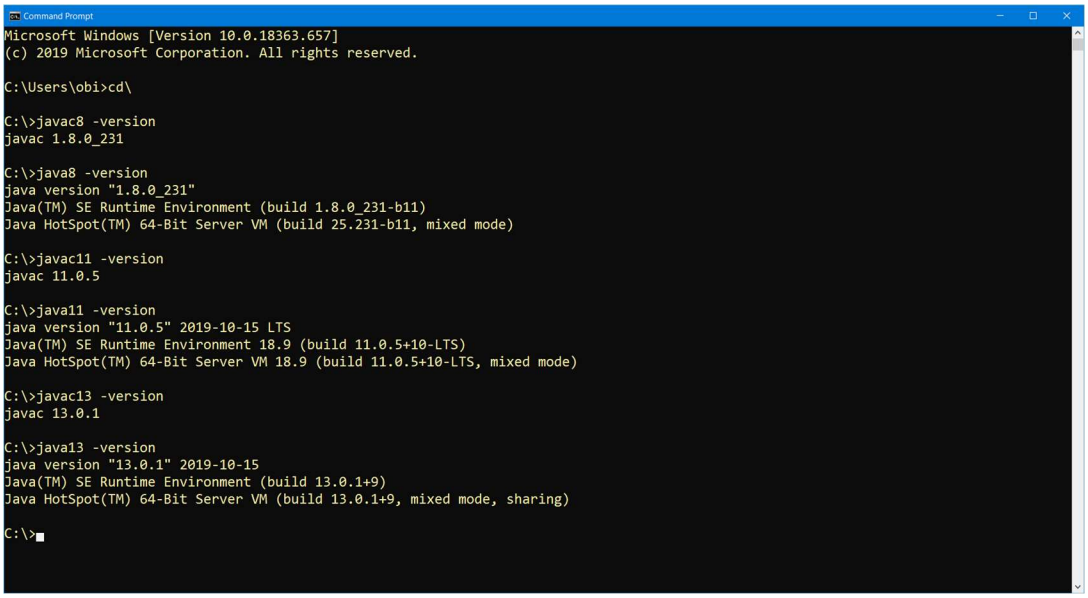
Setting-up your Java Software Development Environment and Tools

1. Java SE Development Kit (JDK):
   1.1. Go to the Oracle website at https://www.oracle.com/java/technologies/downloads/, obtain and install a suitable release of the JDK. JDK version 22 or 21, is recommended. You may follow the detailed step-by-step guide provided in the Setting-up document.
   1.2. Demonstrate and provide evidence of your having correctly installed a JDK, by taking screenshots of your Windows Command or OS terminal window, similar to the following:



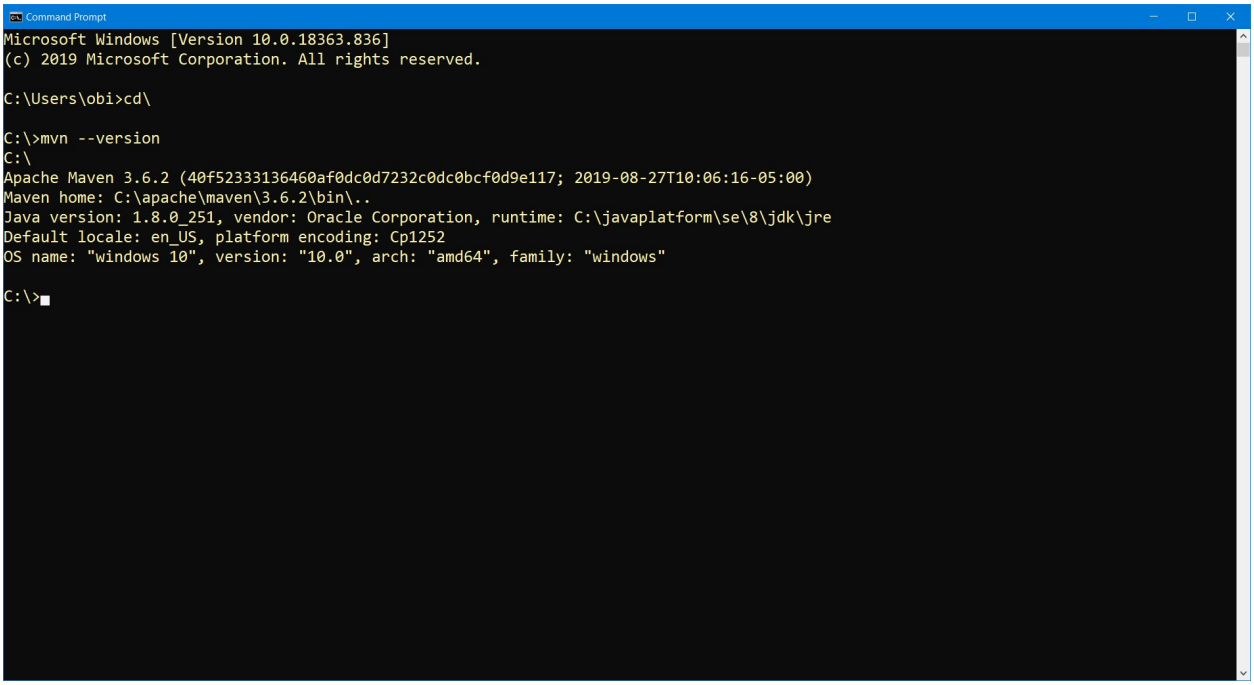
2. Integrated Development Environmnent (IDE) tool:
   2.1. Jetbrains IntelliJ IDEA – Obtain and install IntelliJ IDEA Ultimate Edition from Jetbrains website at https://www.jetbrains.com/idea/download/

   2.2. Alternatively, if you prefer coding your Java with an Eclipse IDE-based tool, then go to the Spring Tools webpage at https://spring.io/tools, obtain/download the "Spring Tools 4 for Eclipse" package appropriate for your OS. If you prefer some

other IDE, such as Netbeans, VS Code, or Eclipse for Enterprise Java etc., then you are welcome to use it.

3. Go to Apache Maven download webpage at http://maven.apache.org/download.cgi and obtain/download the latest released version of Apache Maven tool. Follow the Apache Maven setting-up tutorial (see pdf attached in Lab7a in Sakai Filename: 3_How to Set-up and Begin Using Apache Maven for your projects.pdf) to setup and start working with Maven.

4. Take a screenshot of your Apache Maven version information running in your terminal or Windows command prompt and include it in your submission.



5. **Coding Practice Exercises:** Using your IDE, implement code solution for the following, as a Maven or Gradle (Java Command line/Console application) project:

5.1. 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:08/05/1994

... 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 (as a Maven or Gradle project) 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.miu.cs.cs425.studentrecordsmgmtapp.model.

5.2. In the package named, edu.miu.cs.cs425.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.

5.3. 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 sorted in ascending order of the Students Names. **Note**: Call your printListOfStudents(...) method within your MyStudentRecordsMgmtApp class's main method, then execute it. And take/save a screenshot of the output as displayed in your IDE.

5.4. 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. And take/save a screenshot of the output in your IDE.

5.5. **Further CodingPractice Exercise Problems:**

5.5.1. 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".

5.5.2. 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).

5.6. 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.

5.7. Run – mvn package to produce a jar file of your finished project.

6. Create a git repository (or folder in your existing repository) for your work in the above tasks (call it say, "lab7a"), commit and push your finished assignment up to your github account.

7. Make a submission for this Lab assignment 7a in Sakai, by simply including/submitting the url/link of the repo on your github account.

//-- The End --//