Alexander J King  
Final Project

Comparative Programming Languages

4.28.20

**Introduction**

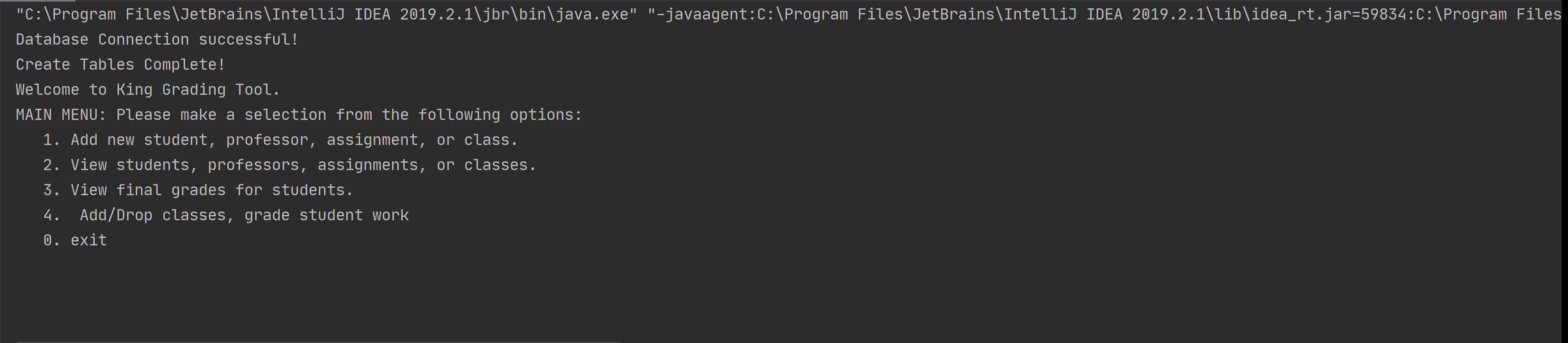
My program KingGradingTool utilizes an sql database to store values and call them back. It allows for adding and viewing students, professors, assignments, and classes. Classes are created and assigned a professor. Then students can sign up for the classes in one of their five open slots. After students are signed up for classes assignments can be created. When an assignment is created it is stored in a table of assignments t\_assignments and then distributed to every student signed up for the class with an entry in t\_strudentassignment table. After assignments are made the professor can go to the grading feature and tell the program what assignment they would like to grade. The professor is then presented with an iteration of the assignment for every student where he assigns the total number of points earned. Points earned cannot be less than 0 and cannot exceed the assigned points possible. After grading is complete final grades for students can be viewed by navigating to the final grades section and inputting the student id and class id. After these are input every assignment pointspossible and pointsearned are added and a final grade is displayed.

Students can also drop classes using the drop class feature. Some inputs that are asked for may not be obvious to the user but can be looked up. For example if a student forgets their student ID number or class number they can navigate to the view menu and will be presented with the information.

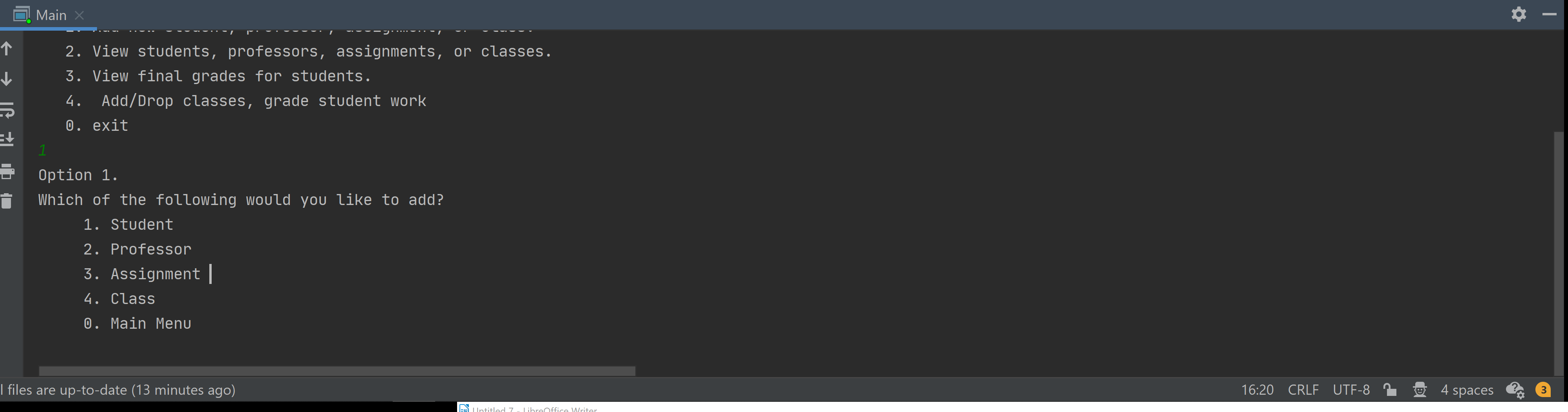
I’ve already created and added students, professors, classes, assignments, and grades so all of this can be tested without the need to create everything from scratch.

**Diving into King Grading Tool**

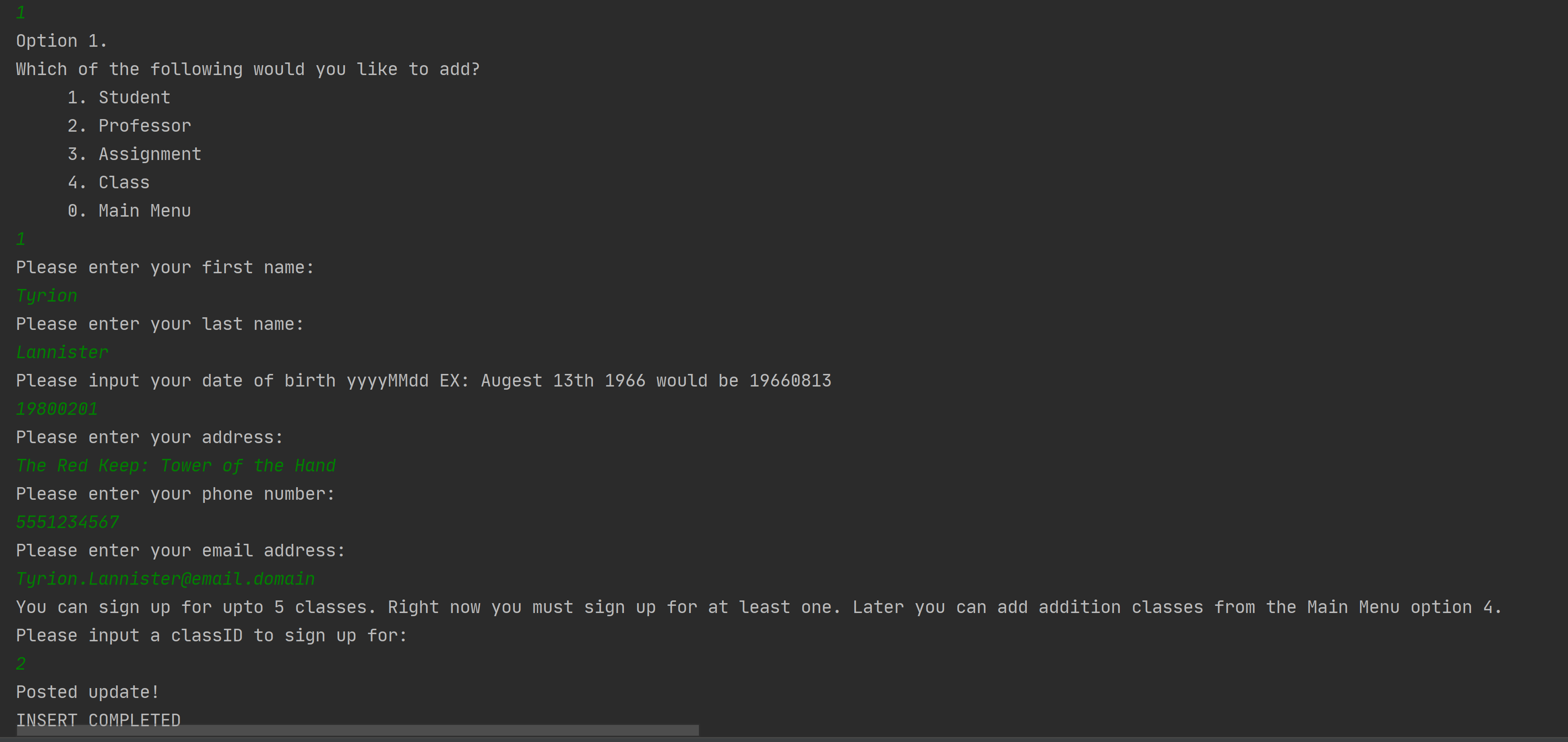
When you first launch the program it will build the tables it wants and check to ensure everything is going as expected and then present you with the main menu.



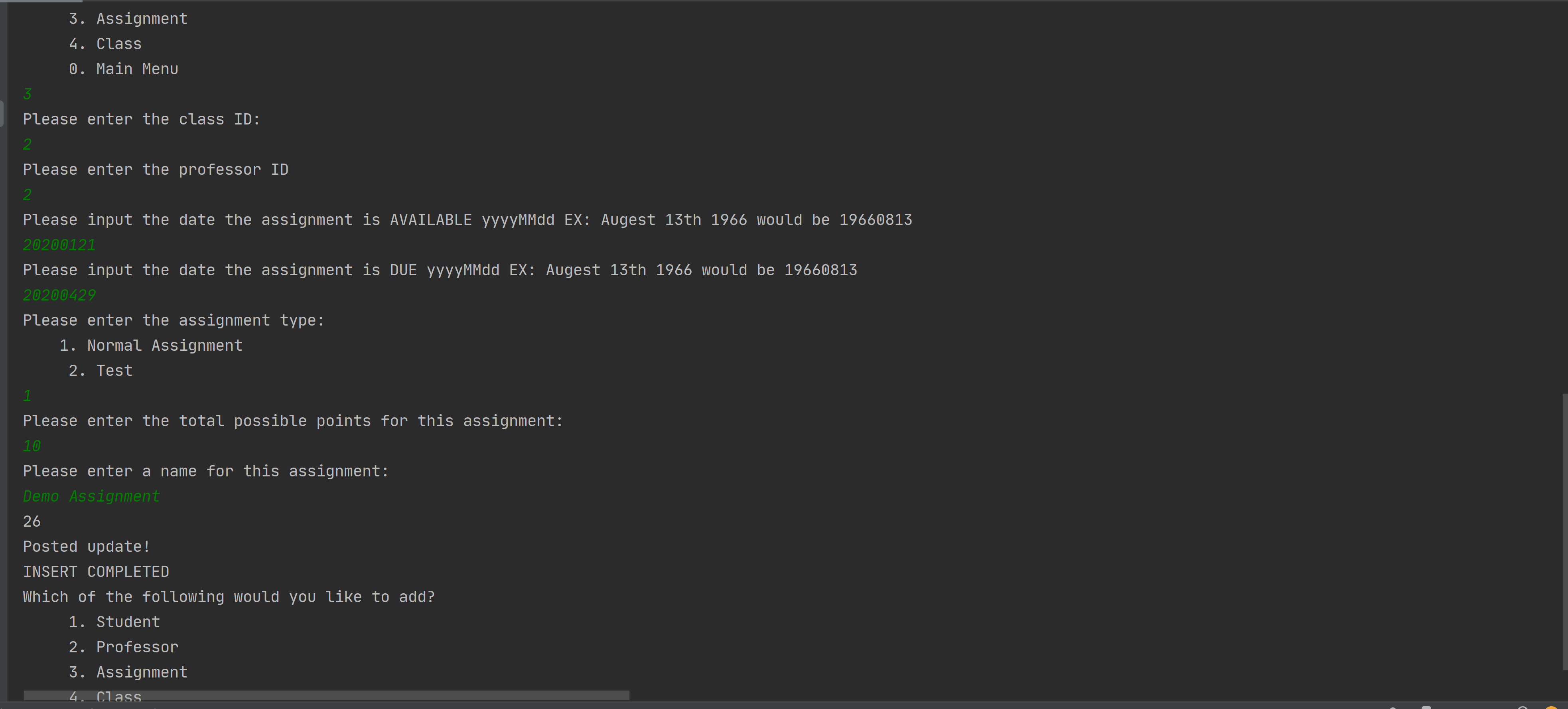
First we will explore the add new menu so we input option 1.



After we’re in the add new menu it will ask us what we would like to add. Lets add a new student.

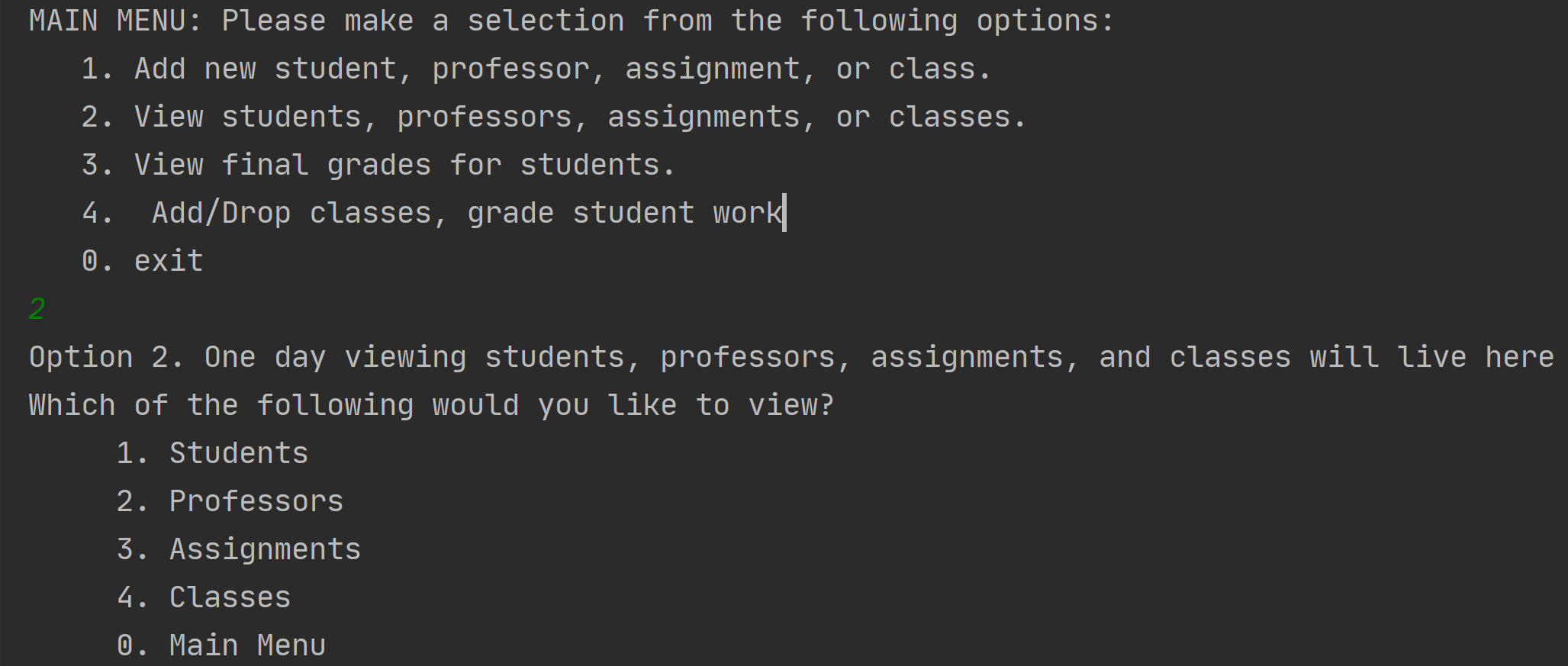
Above we see an example of adding a new student Tyrion Lannister and signing him up for a class. I decided to collect more information than I actually end up utilizing for the scope of the project with the idea of later implementations. We’re going to take note that we’re signed up for class 2.

After we add a student it takes us back to the add menu asking if we would like to add any others. I would now like to add an assignment so I select option 3.

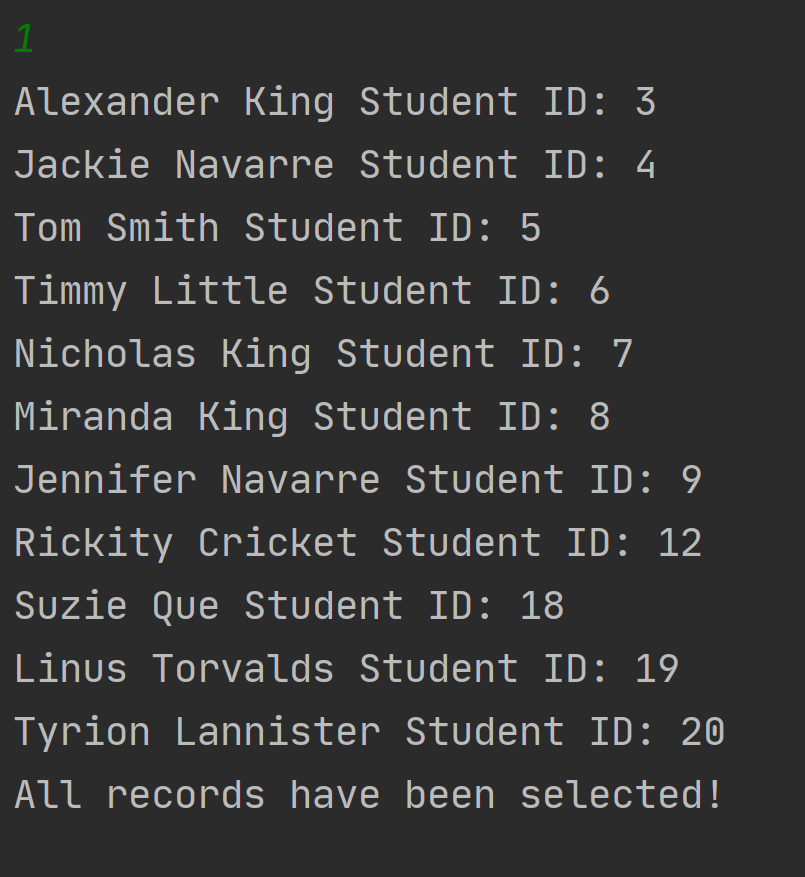


After selecting option three we are prompted for the class ID we would like the assignment to belong to, the date the assignment is available, date due, the type of assignment, the number of possible points, and a name. Currently I do not utilize the assign / due dates or the assignment type. Assignment types are created with the idea of weighing sections homework x% exams x% etc. After we name out assignment Demo Assignment we are presented with the max value of student ID it was assigned to so we know it was distributed to the entire class. Posted update and insert complete are letting us know the processes are done and they are distinct. You could input an improper thing and the insert process will still complete. EX: string where expected int.

Now lets check out the View menu. Here we’re going to make a mental (or physical) note of a couple important things. So we select 0 to go to the main menu and select option 2 to enter the View menu.

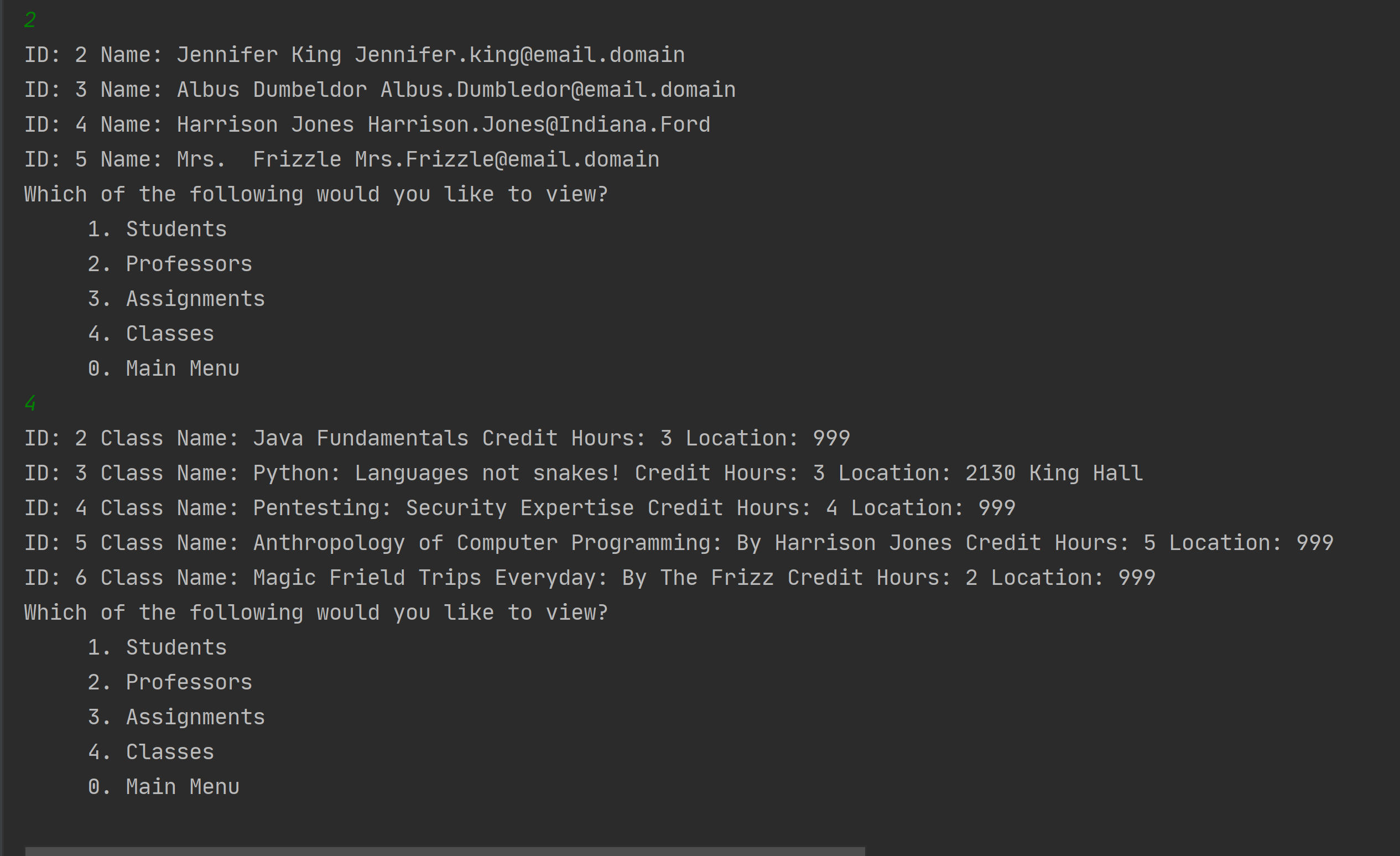


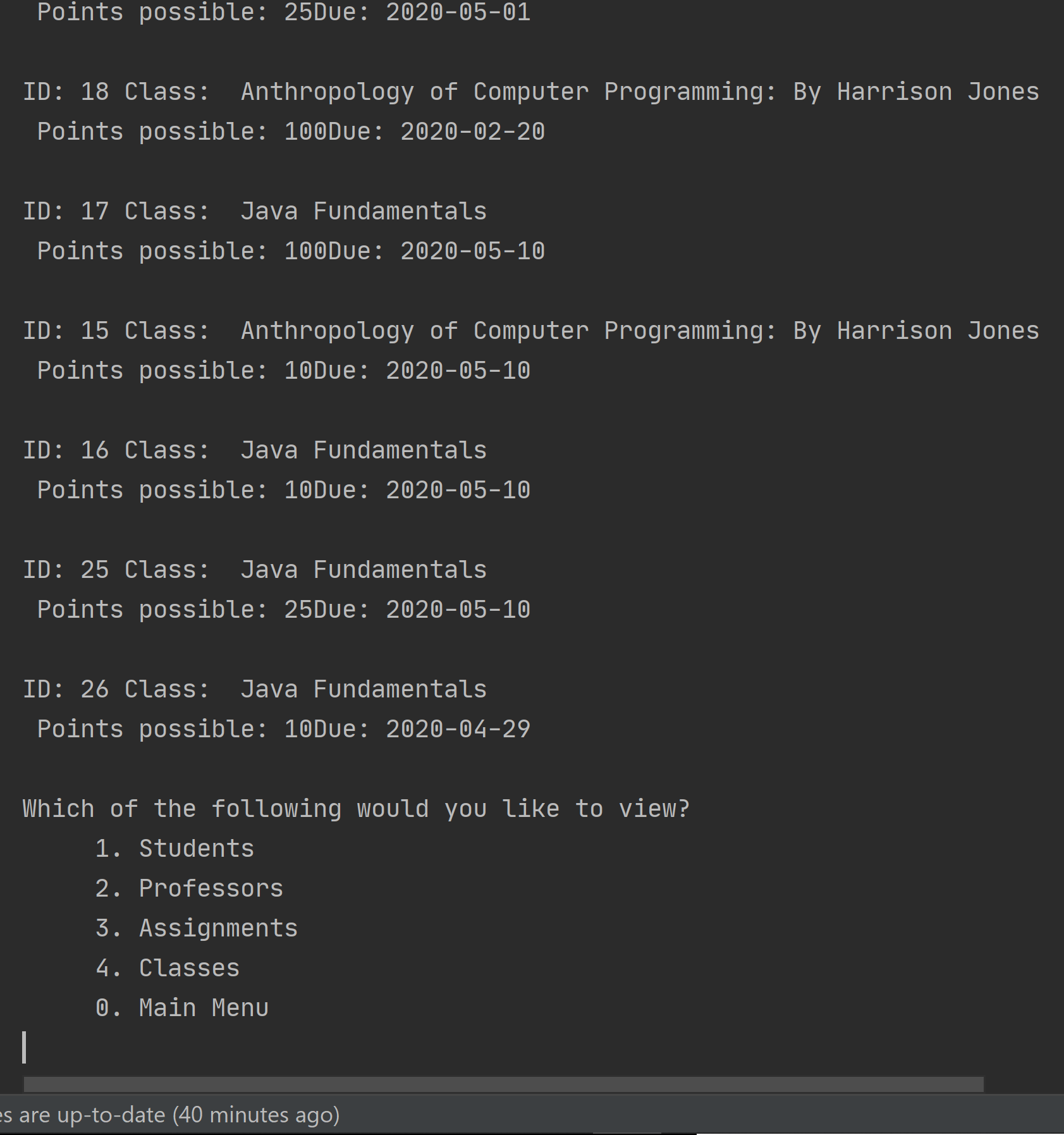
We want to view students, professors, and classes. We’re going to take note of Tyrions student id, any classID he might be signed up for, and his professor(s) id.We will select students first.



Here we are presented with all of the students names and IDs. We’re going to remember that Tyrions id is 20 and we recall from earlier he signed up for class ID 2.

Now we do the same for classes, professors, and assignments:



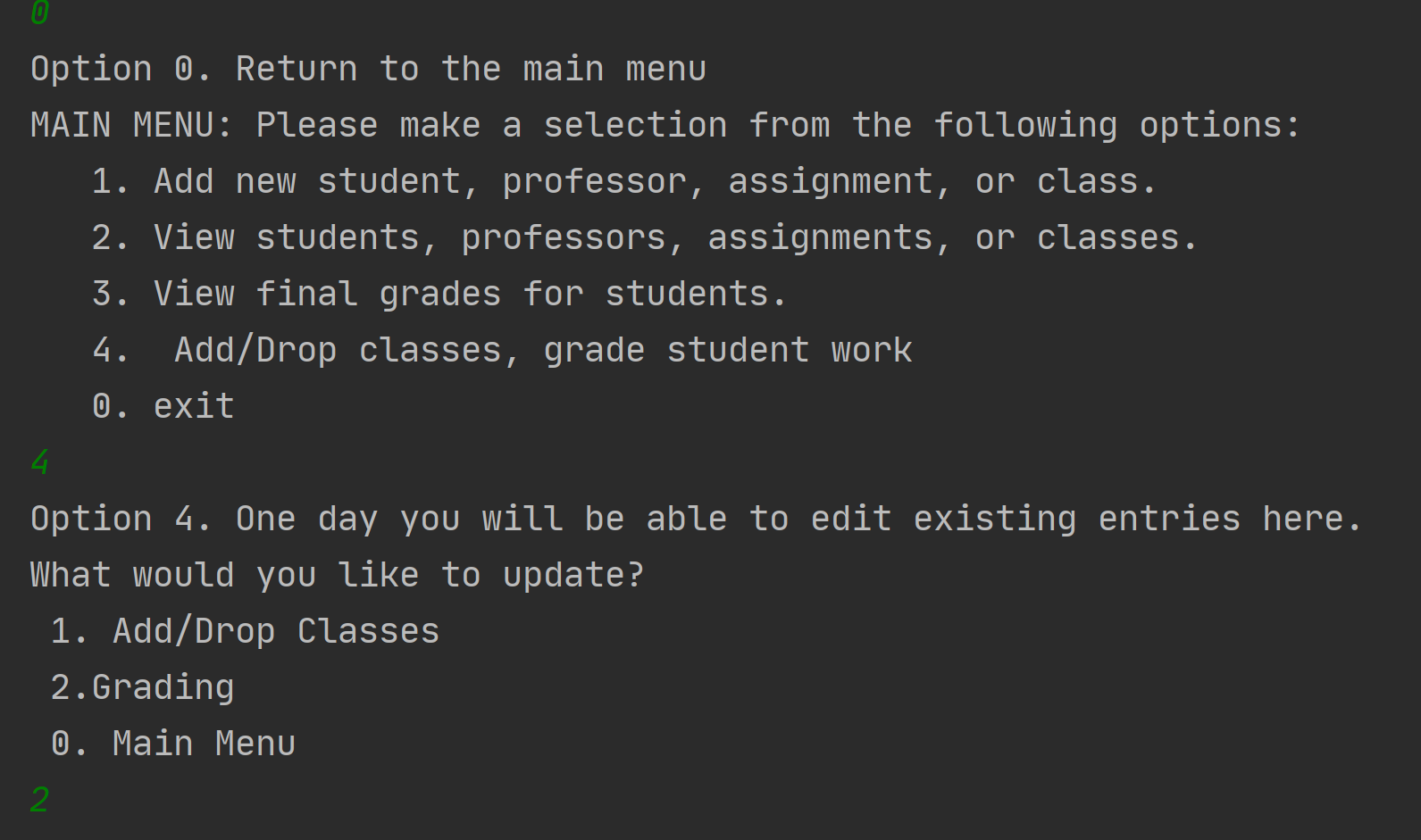


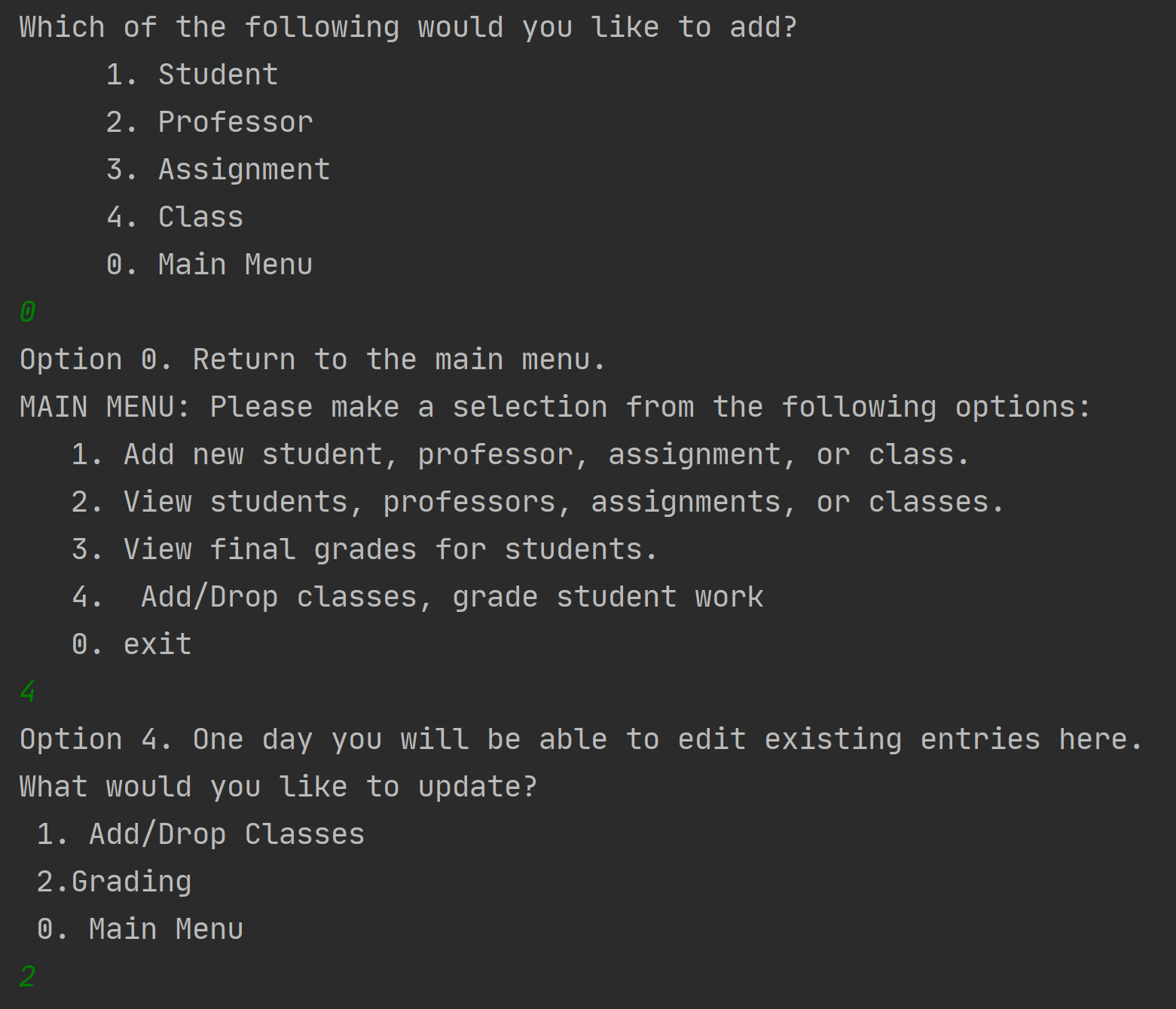
We need to add to our notes here the professor ID who matches with his class id. Here we can see Jennifer King teaches classid2. So we’re going to add her prof id to our notes. As well as the most recent added homework assignment id.   
Student id: 20

classid: 2

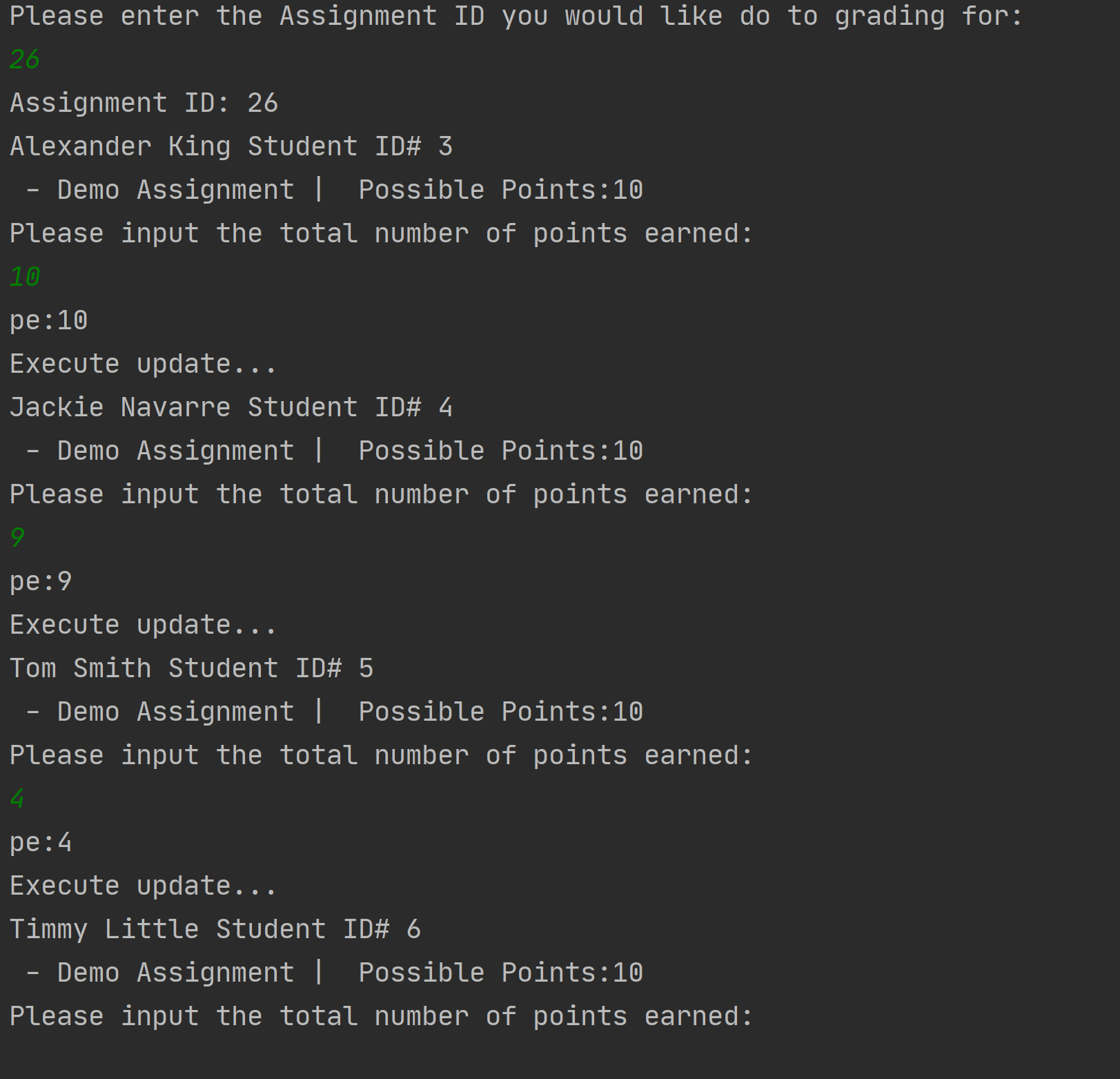
Assignmet id: 26

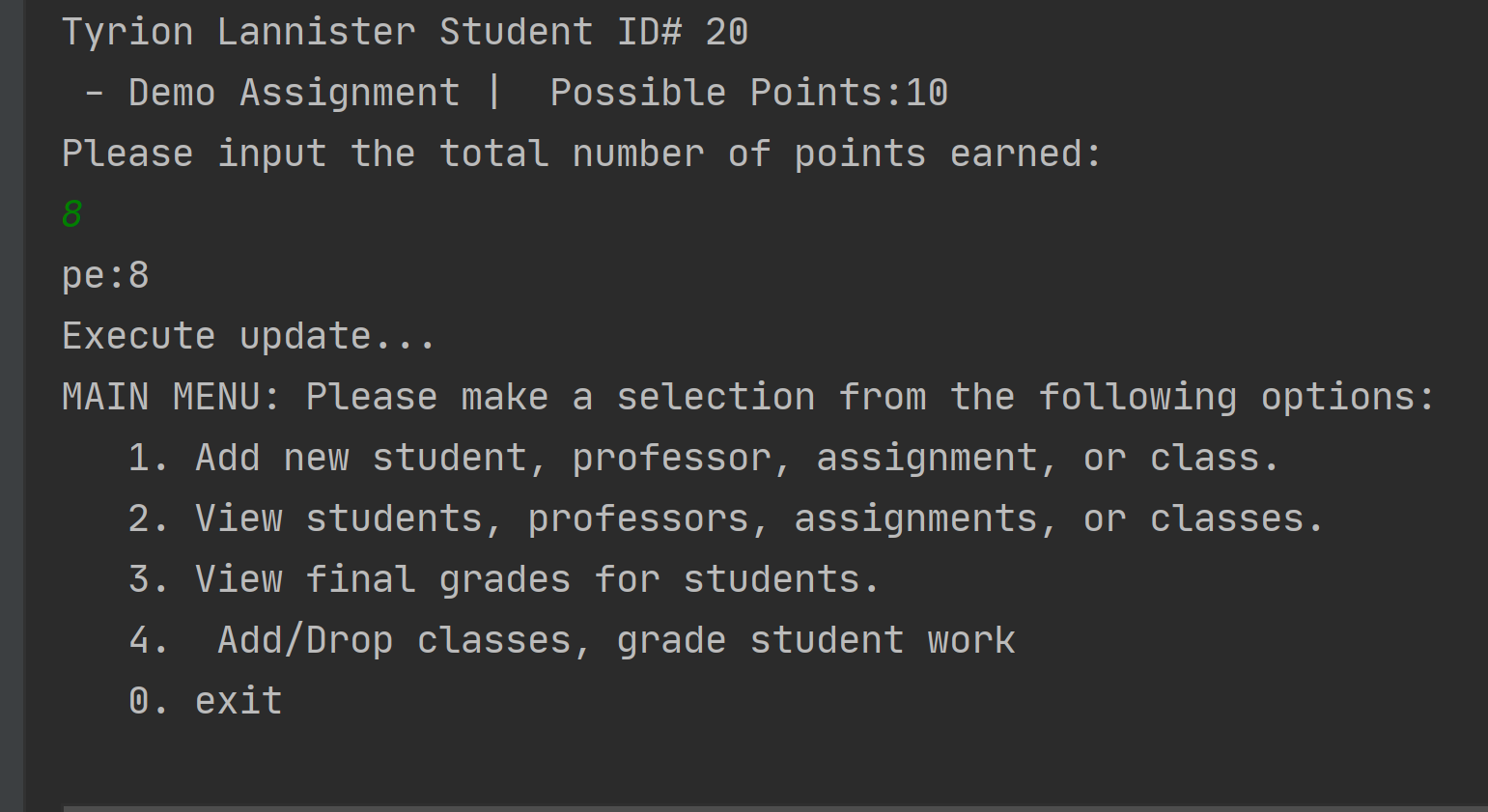
With these know id numbers we’re ready to grade Tyrions work. From the main menu we selection option 4 and then option 2 for grading.



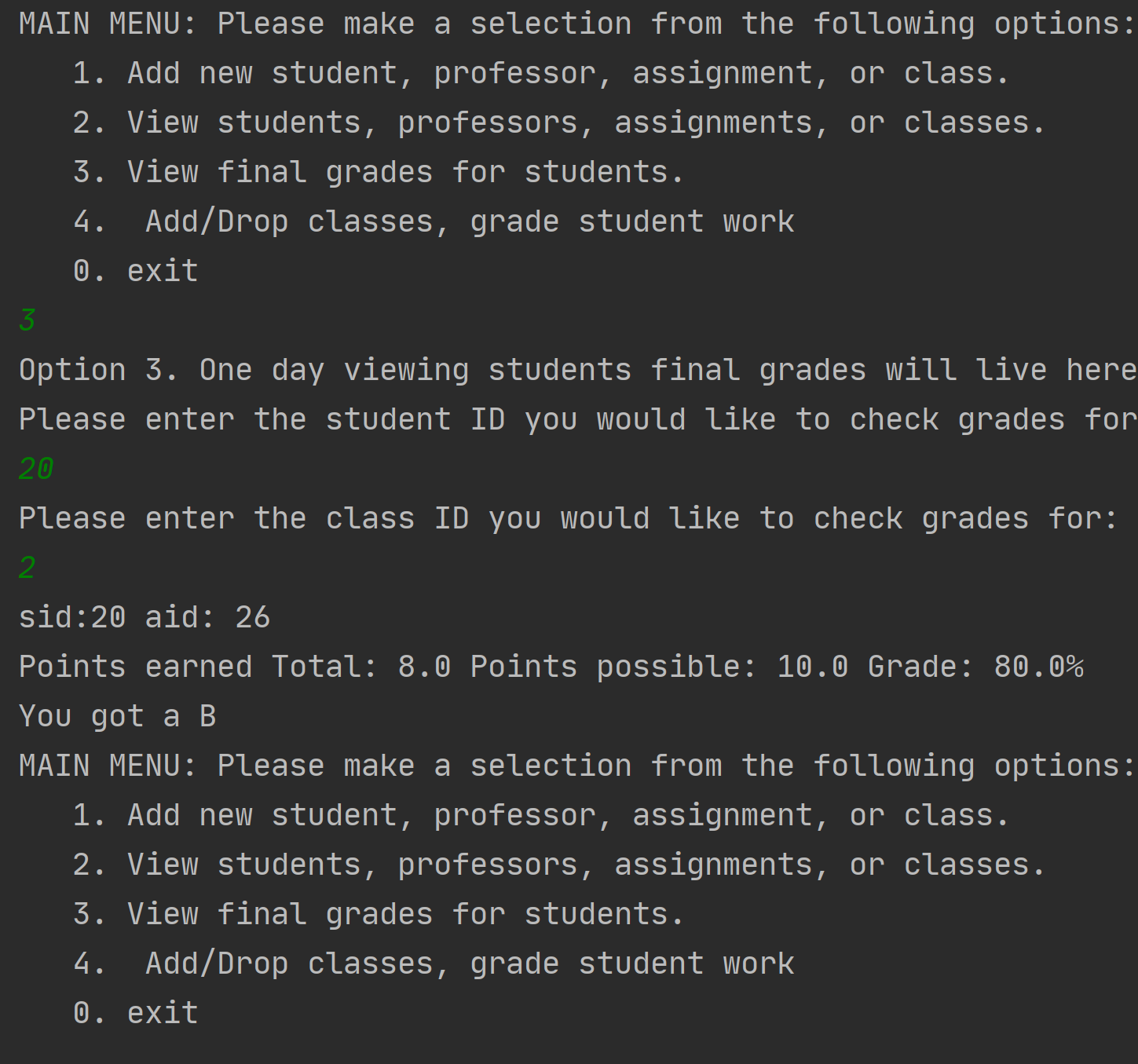


Once we are in the grading section we will be asked for the assignment ID. If we do not know these off the top of our head we can look them up in the view menu. We took note that the one we want is 26. After inputting the assignment it will tell you the total possible points and ask you for total points earned for every student in the class. After inputting the points earned it will echo it back to you and then move on to the next student until we have completed them all.





So now we can view final grades for Tyrion. We need to tell the program our student id (20) and the class we want grades for (2). We expect him to get a B on our grading scale since he scored 8/10 on his only assignment. 100-90A, 89-80 B, 79-70 C, 69-60 D, 59-0 F. And that is exactly what we see.



**History and overview of Java and SQL**

**Java**

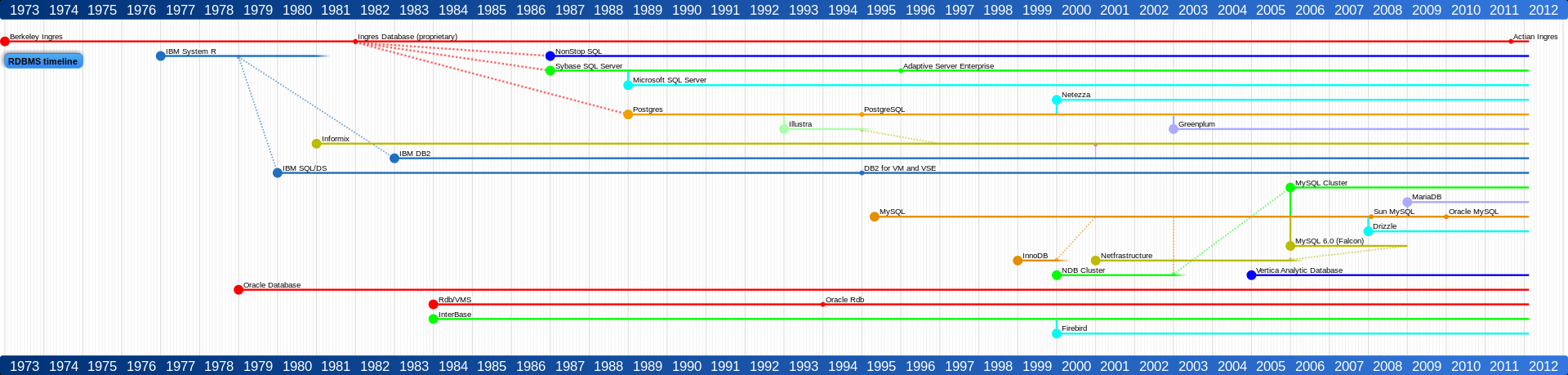
James Gosling, Mike Sheridan, and Patrick Naughton started the Java language project in June 1991 under the name Oak. In 1995 Oak was renamed Java. In 1995 Time magazine called Java one of the 10 best products of 1995. In January 1996 jdk 1.0 released and Java is being used in Windows apps, web apps, enterprize apps, mobile apps, cards, etc. Javas release schedule seems to be erratic and inconsistent. JDK beta 1995 – Java SE 10 released in 2018.

Java was originally designed for interactive TV but it was too advanced for the digital cable of the time.

<Source for all info in Java: <https://www.javatpoint.com/history-of-java> >

**SQL**

SQL stands for Structured Query Language and it is used for databases.

  
Figure 1: SQL timeline https://sqlwithmanoj.files.wordpress.com/2015/05/rdbms\_timeline-svg.png

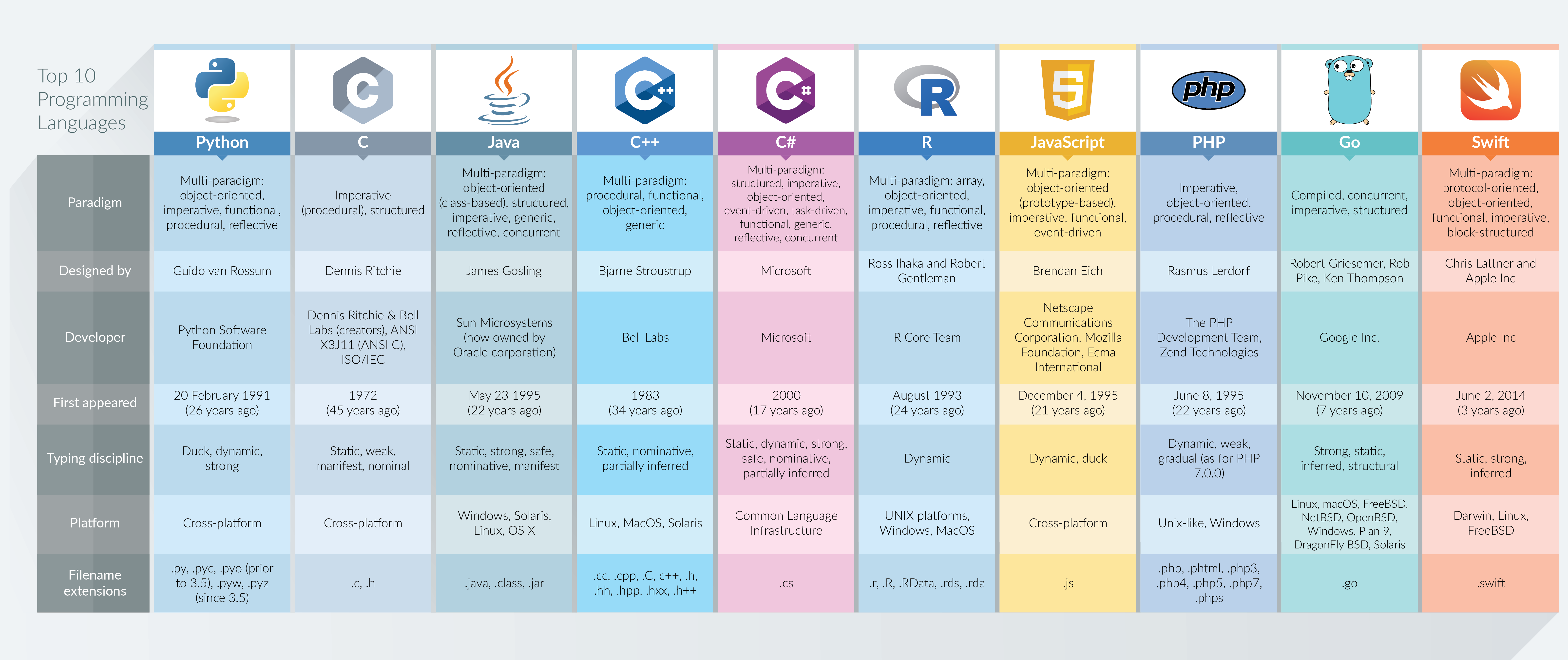
(high enough resolution to zoom in if needed)

A brief history explanation of SQL by Oracle.

“Dr. E. F. Codd published the paper, "A Relational Model of Data for Large Shared Data Banks", in June 1970 in the Association of Computer Machinery (ACM) journal, Communications of the ACM. Codd's model is now accepted as the definitive model for relational database management systems (RDBMS). The language, Structured English Query Language (SEQUEL) was developed by IBM Corporation, Inc., to use Codd's model. SEQUEL later became SQL (still pronounced "sequel"). In 1979, Relational Software, Inc. (now Oracle) introduced the first commercially available implementation of SQL. Today, SQL is accepted as the standard RDBMS language.”

< <https://docs.oracle.com/cd/B19306_01/server.102/b14200/intro001.htm>>

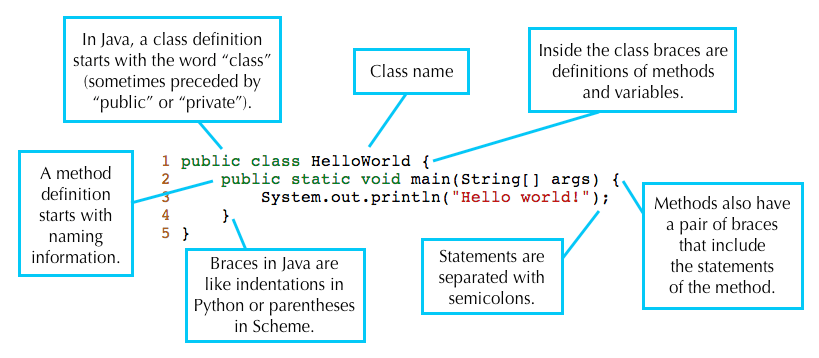
A chart that quickly comparison of programming languages. I choose Java.

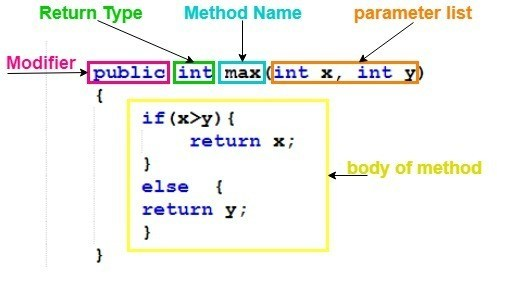
  
Figure 2: DZone > Web Dev Zone > Top 10 Programming Languages in 2017Top 10 Programming Languages in 2017 Anton Shaleynikov < https://dzone.com/articles/top-10-programming-languages-in-2017 >

I picked Java because I know it the best and it has been one of my favorite languages I’ve used.

I find it highly read and writable thus the efficiency of cost is maximized for me. Java is considered to be one of the most reliable languages.

Format of a Java program:

  
Figure 3: Format of a Java Program <https://inst.eecs.berkeley.edu/~cs61b/fa15/hw/lab1/>

  
Figure 4: Explanation of Methods < https://simplesnippets.tech/java-methods-detailed-explanation-with-program-example/ >

Explanation of methods:

Control structures are:

in-sequence – The most basic run the program line by line start to end.

Branching – Calling sections of the program.

Looping – doing and repeating a section of code if the requirements of the loop are met

If statement – if x then do this

else if – same as above but iterated.

Else – if none of the if / else if are satisfied do this.

Java has lots and lots of data structures.

I used arrays, lists, ResultSets, ints, doubles, strings, etc. These all serve their own purpose.

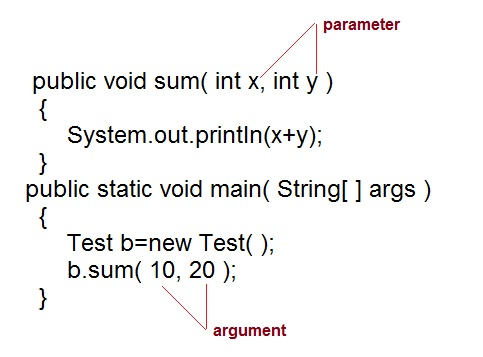
Int – whole number integer

double – 64 bit number with decimal.

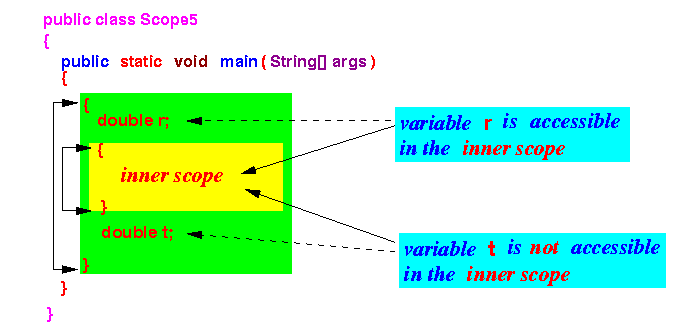
String - “A set of characters in a particular order”

ResultSets – JDBC Java Database Connectivity result set to get my SQL quires.

Parameter passing:

  
Figure 5: Paramerter Passing < https://foundjava.blogspot.com/2017/05/method-overloading-foundjava.html >

Scope: Variables are available within their curly brackets. If you want something in-scope declare it somewhere within the brackets of where you want it.

  
Figure 6: Scope < http://www.mathcs.emory.edu/~cheung/Courses/170/Syllabus/08/scope.html >