Assignment 8 Instructions

1. Assignment 8: **37 - 40 points total**

   [35pts + 2 E.C. pts = 37 points] or [35pts + 2 e.c pts for being on time and 3 e.c pts for Ask and Answer = 40 points] or [35pts + 3 e.c points for Ask and Answer = 38 pts]

   Extra Credit points:

   2 E.C. points (for on-time work)

   3 E.C. points (for Ask and Answer forum participation),

2. Due Date & Time: **11/9/2020 at 11:59 PM**

WHAT TO SUBMIT

Submit 4 files to iLearn by the deadline and post a reflection on iLearn:

1. 3 Java File: Please submit 3 files to iLearn: **RectangleTest.java, EmployeeTest.java, CoffeeShopEmployee.java[30 points]**
2. 1 PDF File: Submit 1 Word/PDF file which is a filled-out, downloaded local copy of this Google page on your local computer, named “firstname-lastname-assignment-7-report.pdf”. Fill this out with screenshots then save it as Word or PDF

Then go to iLearn forum to submit:

3. 1 Reflection **[5 points]**

Then go to iLearn to ask a question:

4. 1 Optional Ask and Answer forum **[3 points]**

HOW TO SUBMIT

Please upload all files **separately** via iLearn Assignments Submission, unzipped

GUIDELINES FOR <u>ALL ASSIGNMENTS</u>:

1. Each assignment includes a code portion and a non-code portion. Please submit both 2 portions.
    a. Code portion: Your source code files, only the files which you create and edit.
    b. Non-code portion: Your assignment report, only 1 **Word** or **PDF** file.

2. Please submit all required files separately, un-zipped, via iLearn Assignments Submission

3. Always **<u>read through the entire assignment before starting and submitting any of it. Missing files or missing requirements will result in deducted points</u>**

4. **<u>a. Include a proper header at the top of every Java file. Figure 1</u>**

Header Format

```
/*
* Assignment <assignment number>
* Description: <program description>
* Name: <your name>
* ID: <your SFSU ID number>
* Class: CSC 210-<section number>
* Semester: <current semester>
*/
```

Replace each tag (such as **<assignment number>**) with the appropriate text.

You should adhere to this format as closely as possible. You do not need to include the **<>** symbols in your header fields.

<u>b. Only if you work with a Study Buddy, include your Buddy’s name in your header at the top of every Java file. Figure 1</u>

Header Format

```
/*
```

```
* Assignment <assignment number>

* Description: <program description>

* Name: <your name>

* Teammate: <Study Buddy name>

* ID: <your SFSU ID number>

* Class: CSC 210-<section number>

* Semester: <current semester>

*/
```

Assignment 8

PRACTICE WITH OBJECTS

❐Part 1: RectangleTest Class [8 points]

File Name: RectangleTest.Java

1. (The Rectangle class) Following the example of the Circle class in slides, design a class named Rectangle to represent a rectangle. This class contains:

a. Two double data fields named width and height that specify the width and height of the rectangle. The default values are 1 for both width and height.

b. A no-arg constructor that creates a default rectangle.

c. A constructor that creates a rectangle with the specified width and height.

d. A method named getArea() that returns the area of this rectangle.

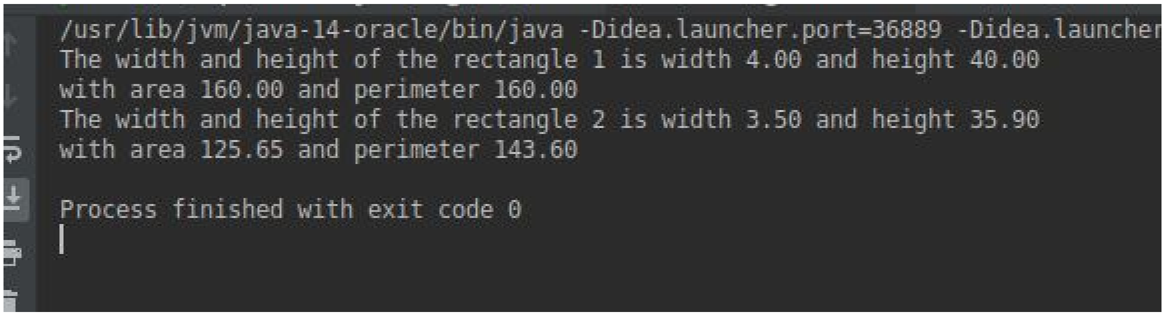
e. A method named getPerimeter() that returns the perimeter.

2. Implement the above class.

Write a test program **RectangleTest.java** that creates two Rectangle objects—one with width 4 and height 40 and the other with width 3.5 and height 35.9. Display the width, height, area, and perimeter of each rectangle in this. Note: Rectangle class should be in this same class

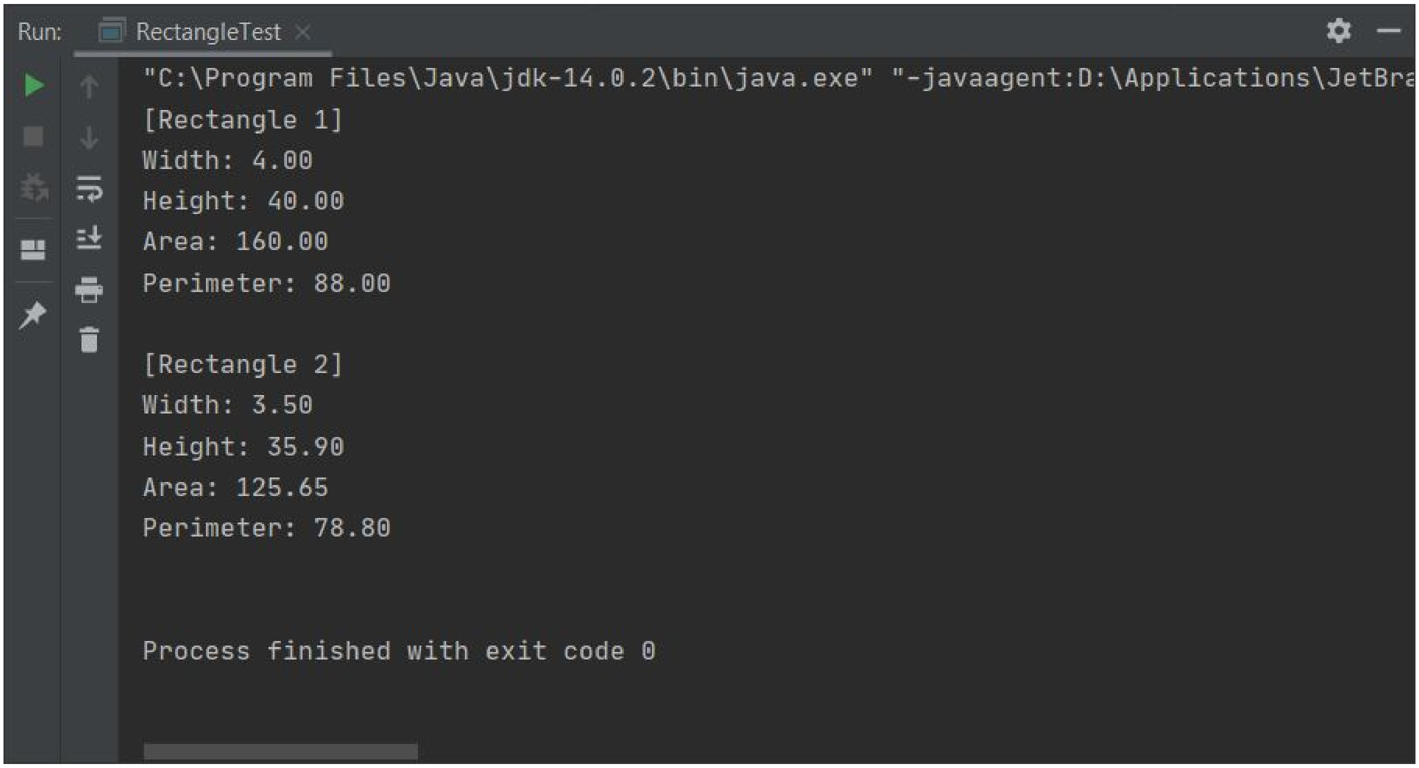
order. (10 points)

Sample output:



```
/usr/lib/jvm/java-14-oracle/bin/java -Didea.launcher.port=36889 -Didea.launcher
The width and height of the rectangle 1 is width 4.00 and height 40.00
with area 160.00 and perimeter 160.00
The width and height of the rectangle 2 is width 3.50 and height 35.90
with area 125.65 and perimeter 143.60

Process finished with exit code 0
```

There was no box here… I'm assuming you want a screenshot of the output here?



```
"C:\Program Files\Java\jdk-14.0.2\bin\java.exe" "-javaagent:D:\Applications\JetBra
[Rectangle 1]
Width: 4.00
Height: 40.00
Area: 160.00
Perimeter: 88.00

[Rectangle 2]
Width: 3.50
Height: 35.90
Area: 125.65
Perimeter: 78.80


Process finished with exit code 0
```

❐Part 2: Employee Class [8 points]

File Name: EmployeeTest.Java

1. (The Employee class) Design a class named Employee. This class contains:

a. A string data field named employeeId for the employee's id

b. A string data field named name for employee's name

c. A string data field named email for email

d. A int data field named deptId for the department id

e. A constructor with no parameter

f. A constructor that creates an employee with the specified employeeId, name, email, and deptId. Use the **same variable names and assign it to the clas data field using the keyword "this"**

g. Create getters and setters for each variable:

i). Create setter methods named setEmployeeName() that sets the employee's name, setEmployeeId() that sets the employeeId, setEmployeeEmail() that sets the employee email and setEmployeeDeptId() that sets the department ID.

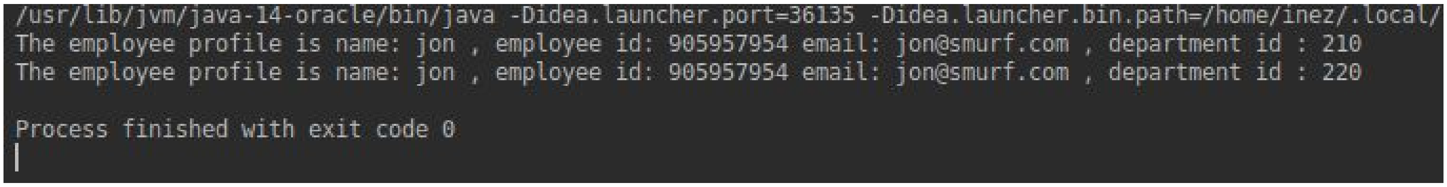
ii). Create getter methods modeled like above i.e. getEmployeeName, getEmployeeId, etc. for all the variables and return the values

2. Implement the above class.

Write a **EmployeeTest.Java** test program that creates an Employee object with your name as name, and employeeId is initialized with your student ID, your student email as email, and 210 for department ID. Print out this initial profile by using the getter methods.

Then use setEmployeeDeptId() method, to change the department ID to 220. Print this new profile with the getter method for Employee Department Id.

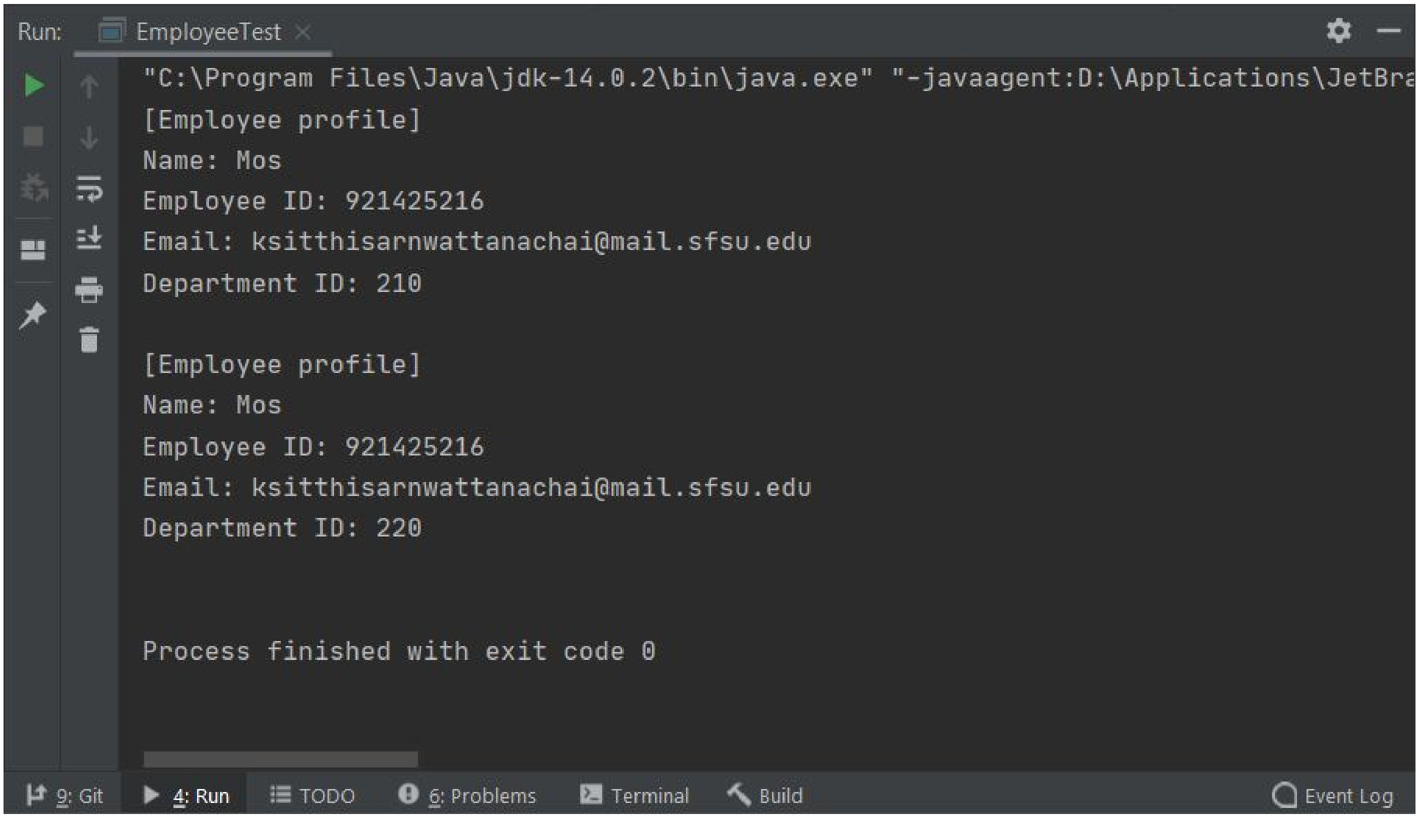
Example output



```
/usr/lib/jvm/java-14-oracle/bin/java -Didea.launcher.port=36135 -Didea.launcher.bin.path=/home/inez/.local/
The employee profile is name: jon , employee id: 905957954 email: jon@smurf.com , department id : 210
The employee profile is name: jon , employee id: 905957954 email: jon@smurf.com , department id : 220

Process finished with exit code 0
```

Please paste your screenshot for this scenario here:

Output :



```
"C:\Program Files\Java\jdk-14.0.2\bin\java.exe" "-javaagent:D:\Applications\JetBra
[Employee profile]
Name: Mos
Employee ID: 921425216
Email: ksitthisarnwattanachai@mail.sfsu.edu
Department ID: 210

[Employee profile]
Name: Mos
Employee ID: 921425216
Email: ksitthisarnwattanachai@mail.sfsu.edu
Department ID: 220


Process finished with exit code 0
```

❐Part 3: CoffeeShopEmployee Class [14 points]

File Name: CoffeeShopEmployee.Java

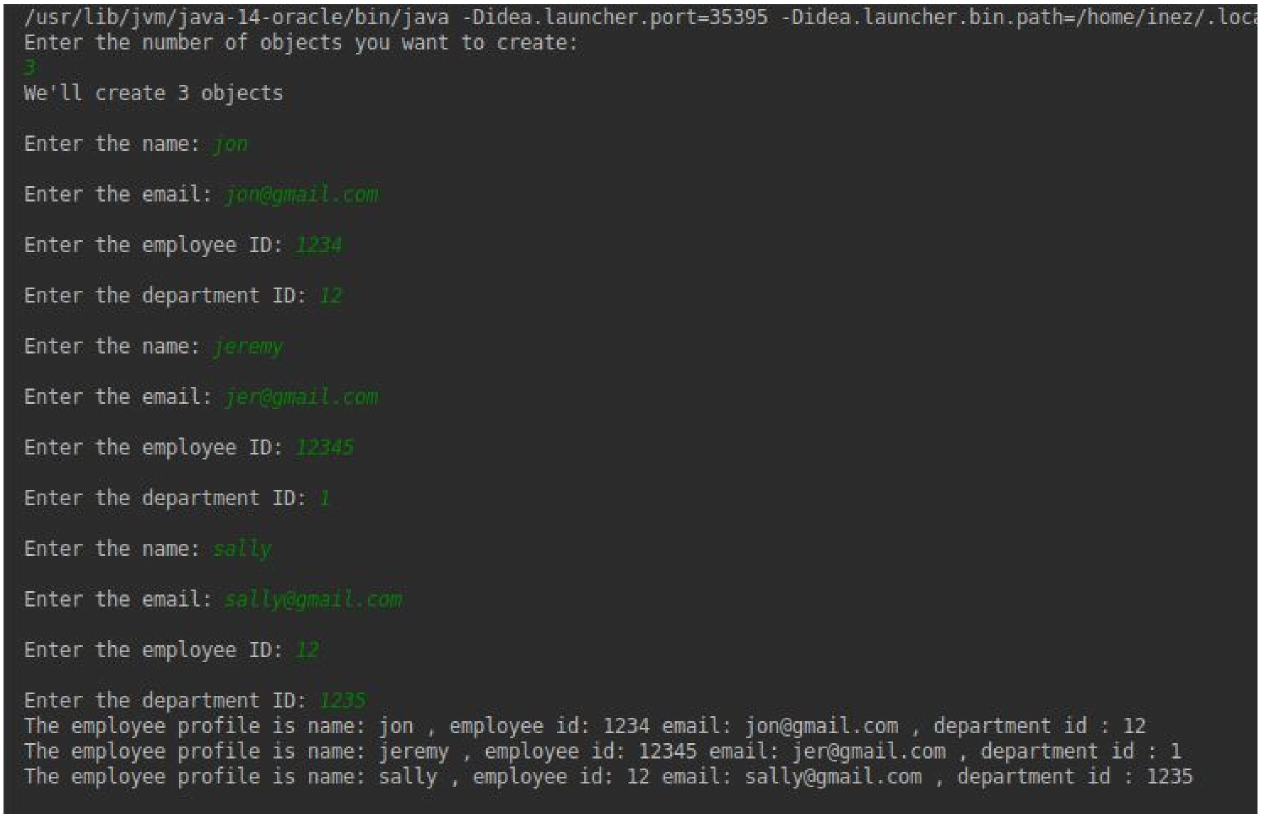
Steps to follow for main method

1. Prompt the user to create 3 or more Employee objects from the Employee class, no parameter is needed

2. Use a loop and an Array of to create these objects and assign them as each element in the array

3. Use a loop to set the name, email, employee id, department Id properties of each objects, for all 4 variables values using setter methods of each object.

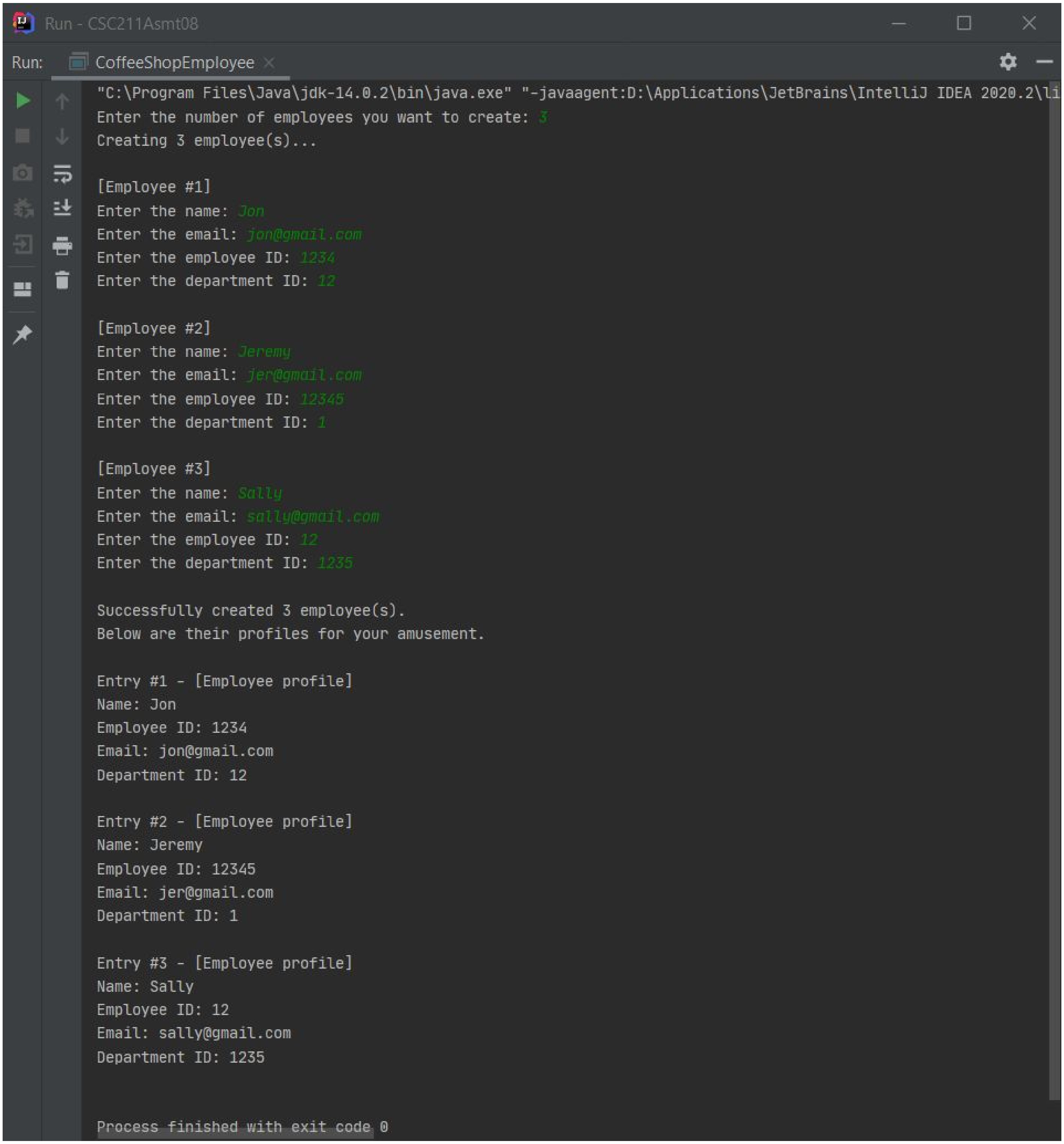
3. Create a method to print each Employee object in the Objects Array. Print out all employee details.



```
/usr/lib/jvm/java-14-oracle/bin/java -Didea.launcher.port=35395 -Didea.launcher.bin.path=/home/inez/.loca
Enter the number of objects you want to create:
3
We'll create 3 objects

Enter the name: jon

Enter the email: jon@gmail.com

Enter the employee ID: 1234

Enter the department ID: 12

Enter the name: jeremy

Enter the email: jer@gmail.com

Enter the employee ID: 12345

Enter the department ID: 1

Enter the name: sally

Enter the email: sally@gmail.com

Enter the employee ID: 12

Enter the department ID: 1235
The employee profile is name: jon , employee id: 1234 email: jon@gmail.com , department id : 12
The employee profile is name: jeremy , employee id: 12345 email: jer@gmail.com , department id : 1
The employee profile is name: sally , employee id: 12 email: sally@gmail.com , department id : 1235
```

Please paste your screenshot for this scenario here:

Output :



```
"C:\Program Files\Java\jdk-14.0.2\bin\java.exe" "-javaagent:D:\Applications\JetBrains\IntelliJ IDEA 2020.2\li
Enter the number of employees you want to create: 3
Creating 3 employee(s)...

[Employee #1]
Enter the name: Jon
Enter the email: jon@gmail.com
Enter the employee ID: 1234
Enter the department ID: 12

[Employee #2]
Enter the name: Jeremy
Enter the email: jer@gmail.com
Enter the employee ID: 12345
Enter the department ID: 1

[Employee #3]
Enter the name: Sally
Enter the email: sally@gmail.com
Enter the employee ID: 12
Enter the department ID: 1235

Successfully created 3 employee(s).
Below are their profiles for your amusement.

Entry #1 - [Employee profile]
Name: Jon
Employee ID: 1234
Email: jon@gmail.com
Department ID: 12

Entry #2 - [Employee profile]
Name: Jeremy
Employee ID: 12345
Email: jer@gmail.com
Department ID: 1

Entry #3 - [Employee profile]
Name: Sally
Employee ID: 12
Email: sally@gmail.com
Department ID: 1235


Process finished with exit code 0
```

❐Part 4: Reflect 50 words [5 points]

Let’s reflect on what we’ve learned so far.

The following reflection is identical to the one posted on iLearn.

This assignment was generally simple and straightforward. There were a couple of ambiguities on where each class needed to be defined and which files to submit. But was later clarified with a mentor on Slack.

The requirement of separately instantiating the Employee object and using setters to provide value seemed a bit confusing and redundant. One could simply instantiate Employee with the constructor that was created as part of an earlier section, and then add that to the array. If the purpose was to make use of the setter, the prompt could’ve asked the user to tweak a particular employee *after* it has been set up.

👏 Assignment 8 Part 4: Your own Reflection

❐Part 5 Optional Extra Credit : Ask and Answer [3 e.c points]

Ask or Answer a question posted in this form

👏 Assignment 8 Part 5: Ask and Answer