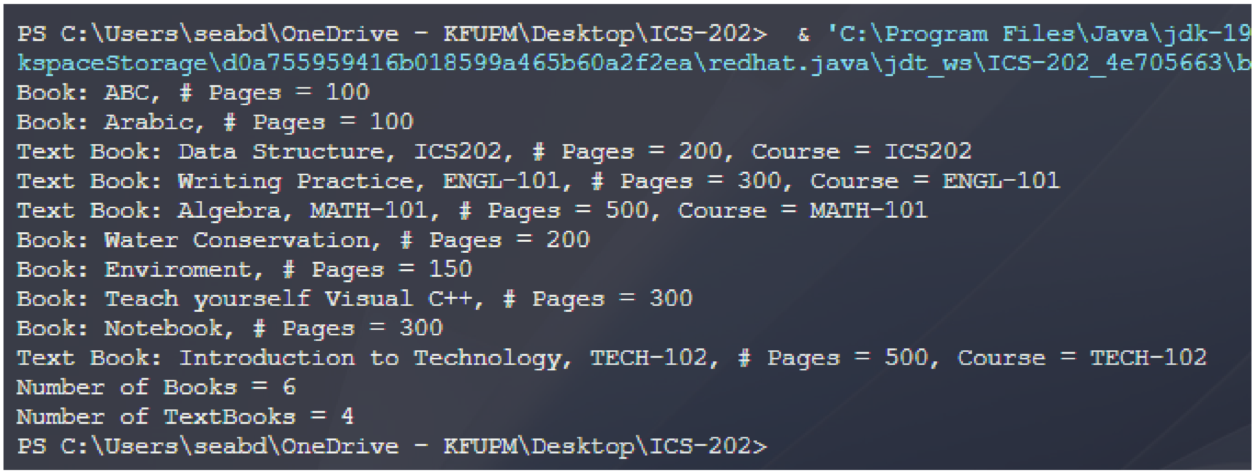
Name : Abdullah Al-Abbas | ID : 202156190 | Section : 54

Practice Exercise Solution:

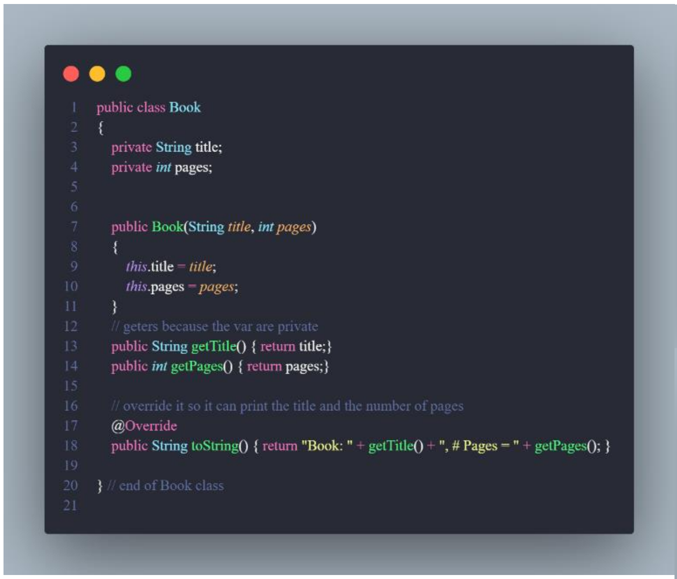
a. First, I write the instance variables that are in the question, then I created a getter for both of these variables since a declare them as a private, then I override the toString method an I make it return the name of the book and the title , then I created a subclass called Textbook and I added third variable called course with it's getter and I override the toString method again with the course in it , and finally in the main class (Q1) I have created an array with size 10 and it's type is Book so that I matched the sample output and I have created two variables to count the number for the book and textbooks then I created a for loop that prints all the objects and see if the type of the object is Textbook or book to count.

b.

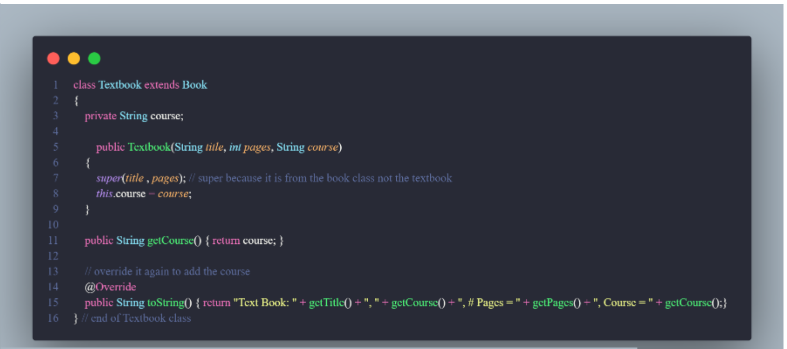


```
PS C:\Users\seabd\OneDrive - KFUPM\Desktop\ICS-202>  & 'C:\Program Files\Java\jdk-19
kspaceStorage\d0a755959416b018599a465b60a2f2ea\redhat.java\jdt_ws\ICS-202_4e705663\b
Book: ABC, # Pages = 100
Book: Arabic, # Pages = 100
Text Book: Data Structure, ICS202, # Pages = 200, Course = ICS202
Text Book: Writing Practice, ENGL-101, # Pages = 300, Course = ENGL-101
Text Book: Algebra, MATH-101, # Pages = 500, Course = MATH-101
Book: Water Conservation, # Pages = 200
Book: Enviroment, # Pages = 150
Book: Teach yourself Visual C++, # Pages = 300
Book: Notebook, # Pages = 300
Text Book: Introduction to Technology, TECH-102, # Pages = 500, Course = TECH-102
Number of Books = 6
Number of TextBooks = 4
PS C:\Users\seabd\OneDrive - KFUPM\Desktop\ICS-202>
```

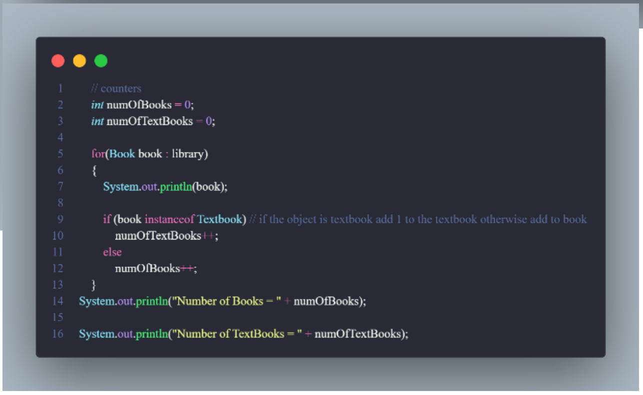
c.



```java
public class Book
{
   private String title;
   private int pages;


   public Book(String title, int pages)
   {
      this.title = title;
      this.pages = pages;
   }
   // geters because the var are private
   public String getTitle() { return title;}
   public int getPages() { return pages;}

   // override it so it can print the title and the number of pages
   @Override
   public String toString() { return "Book: " + getTitle() + ", # Pages = " + getPages(); }

} // end of Book class

```



```java
class Textbook extends Book
{
   private String course;

      public Textbook(String title, int pages, String course)
   {
      super(title , pages); // super because it is from the book class not the textbook
      this.course = course;
   }

   public String getCourse() { return course; }

   // override it again to add the course
   @Override
   public String toString() { return "Text Book: " + getTitle() + ", " + getCourse() + ", # Pages = " + getPages() + ", Course = " + getCourse();}
} // end of Textbook class
```

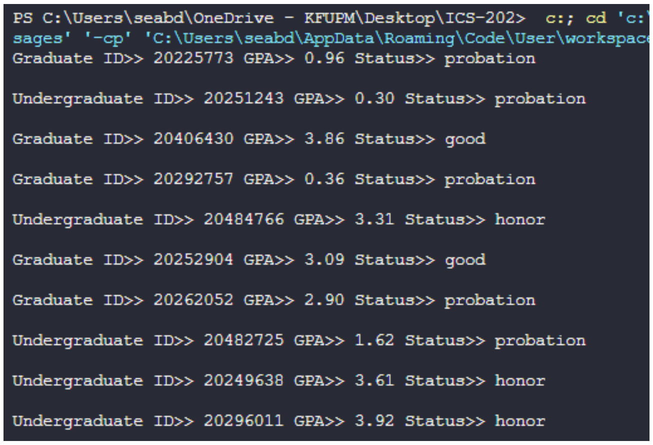


```java
      // counters
      int numOfBooks = 0;
      int numOfTextBooks = 0;

      for(Book book : library)
      {
         System.out.println(book);

         if (book instanceof Textbook) // if the object is textbook add 1 to the textbook otherwise add to book
            numOfTextBooks++;
         else
            numOfBooks++;
      }
   System.out.println("Number of Books = " + numOfBooks);

   System.out.println("Number of TextBooks = " + numOfTextBooks);
```

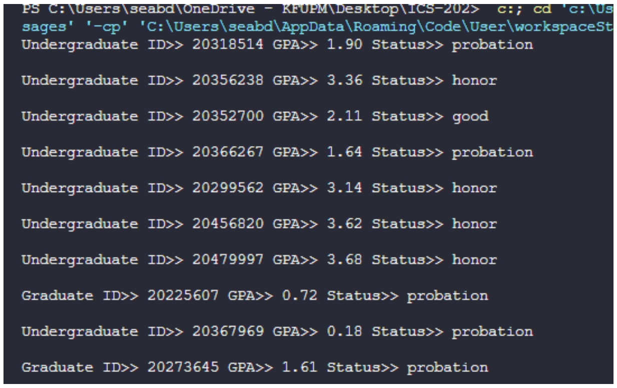
Lab Exercise

a. First, I created an abstract class called student that have two variables and their getters , then I created the abstract method called getStatus that will be overwritten later, then I created another method called display students that display the information, then I created a subclass called Graduate that have the getStatus method only with the the condition that have been provided in the question and same thing happen to the second subclass ‘Undergraduate’ and in the main class ‘Q2’ I created a random id that is close to the kfupm id and a random gpa out of 4 and I printed some random student object

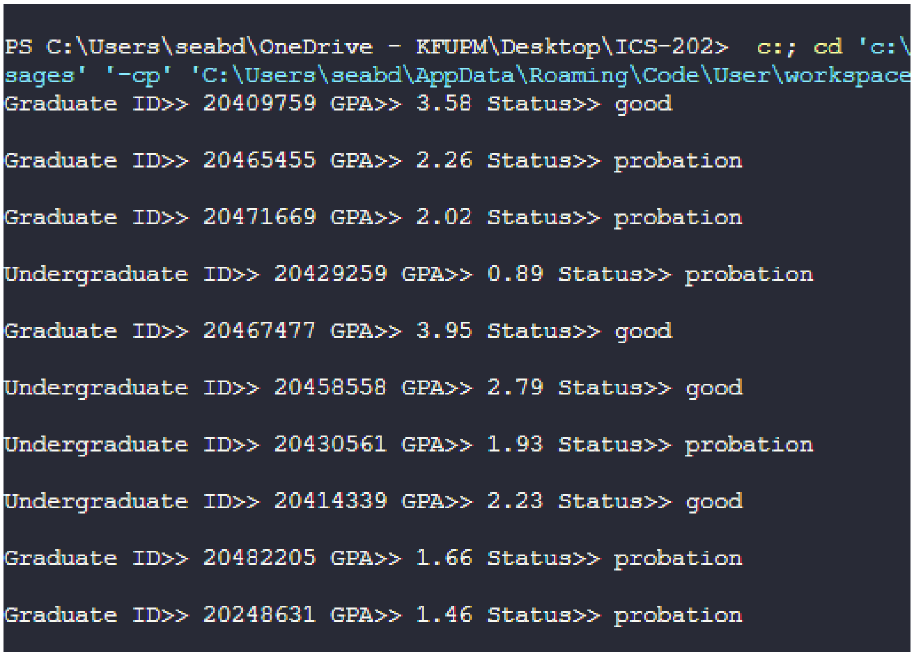
b.



```
PS C:\Users\seabd\OneDrive - KFUPM\Desktop\ICS-202>  c:; cd 'c:\
sages' '-cp' 'C:\Users\seabd\AppData\Roaming\Code\User\workspace
Graduate ID>> 20225773 GPA>> 0.96 Status>> probation

Undergraduate ID>> 20251243 GPA>> 0.30 Status>> probation

Graduate ID>> 20406430 GPA>> 3.86 Status>> good

Graduate ID>> 20292757 GPA>> 0.36 Status>> probation

Undergraduate ID>> 20484766 GPA>> 3.31 Status>> honor

Graduate ID>> 20252904 GPA>> 3.09 Status>> good

Graduate ID>> 20262052 GPA>> 2.90 Status>> probation

Undergraduate ID>> 20482725 GPA>> 1.62 Status>> probation

Undergraduate ID>> 20249638 GPA>> 3.61 Status>> honor

Undergraduate ID>> 20296011 GPA>> 3.92 Status>> honor
```

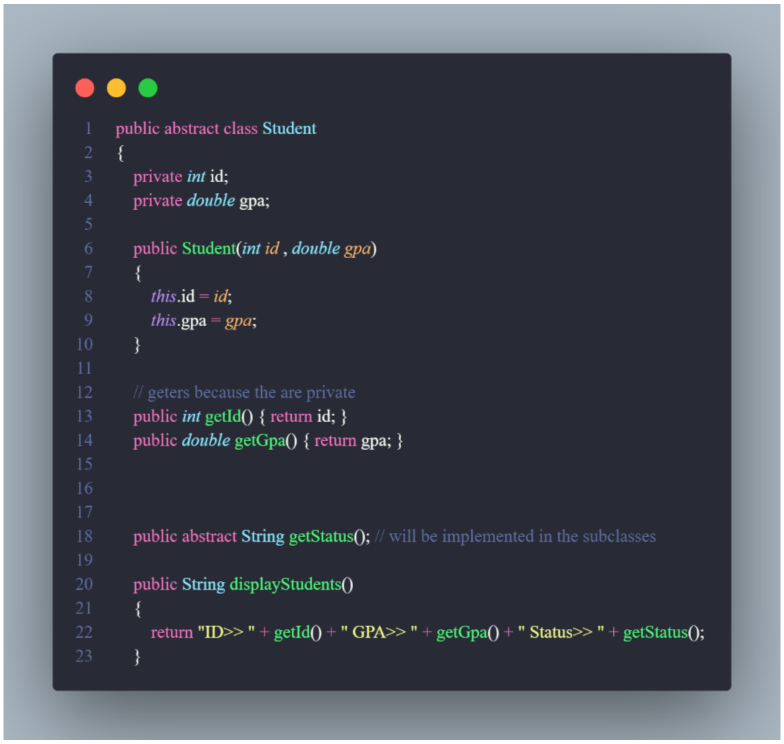


```
PS C:\Users\seabd\OneDrive - KFUPM\Desktop\ICS-202>  c:; cd 'c:\Us
sages' '-cp' 'C:\Users\seabd\AppData\Roaming\Code\User\workspaceSt
Undergraduate ID>> 20318514 GPA>> 1.90 Status>> probation

Undergraduate ID>> 20356238 GPA>> 3.36 Status>> honor

Undergraduate ID>> 20352700 GPA>> 2.11 Status>> good

Undergraduate ID>> 20366267 GPA>> 1.64 Status>> probation

Undergraduate ID>> 20299562 GPA>> 3.14 Status>> honor

Undergraduate ID>> 20456820 GPA>> 3.62 Status>> honor

Undergraduate ID>> 20479997 GPA>> 3.68 Status>> honor

Graduate ID>> 20225607 GPA>> 0.72 Status>> probation

Undergraduate ID>> 20367969 GPA>> 0.18 Status>> probation

Graduate ID>> 20273645 GPA>> 1.61 Status>> probation
```

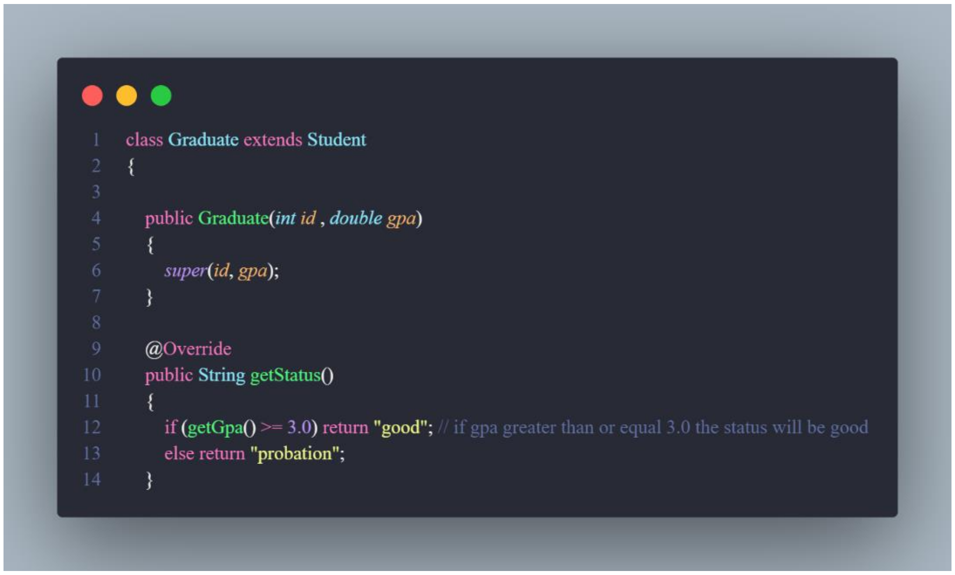


```
PS C:\Users\seabd\OneDrive - KFUPM\Desktop\ICS-202>  c:; cd 'c:\
sages' '-cp' 'C:\Users\seabd\AppData\Roaming\Code\User\workspace
Graduate ID>> 20409759 GPA>> 3.58 Status>> good

Graduate ID>> 20465455 GPA>> 2.26 Status>> probation

Graduate ID>> 20471669 GPA>> 2.02 Status>> probation

Undergraduate ID>> 20429259 GPA>> 0.89 Status>> probation

Graduate ID>> 20467477 GPA>> 3.95 Status>> good

Undergraduate ID>> 20458558 GPA>> 2.79 Status>> good

Undergraduate ID>> 20430561 GPA>> 1.93 Status>> probation

Undergraduate ID>> 20414339 GPA>> 2.23 Status>> good

Graduate ID>> 20482205 GPA>> 1.66 Status>> probation

Graduate ID>> 20248631 GPA>> 1.46 Status>> probation
```

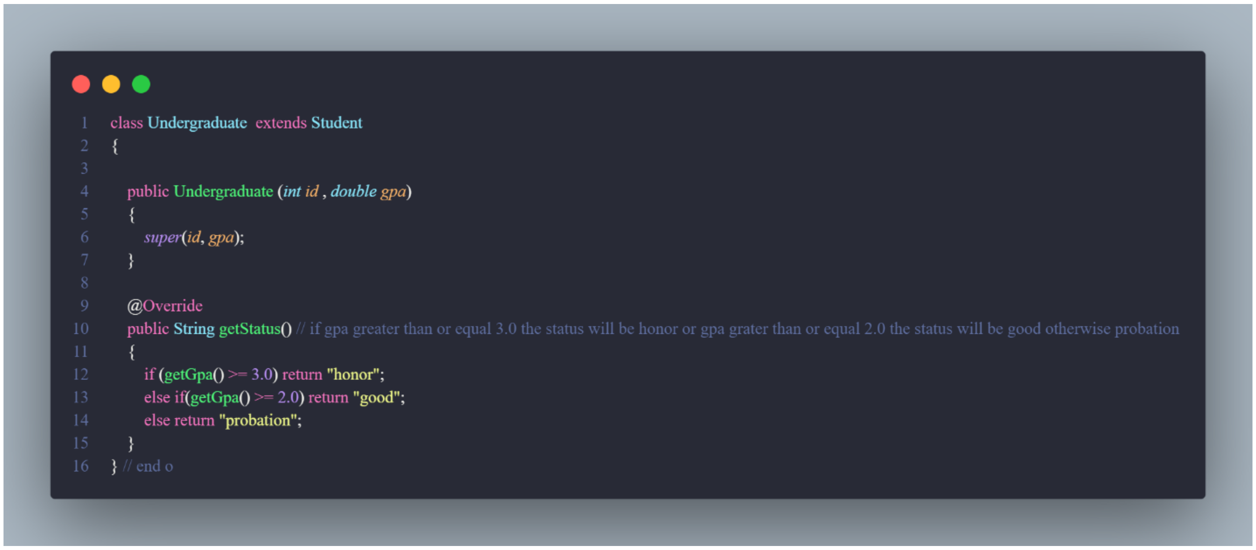
C



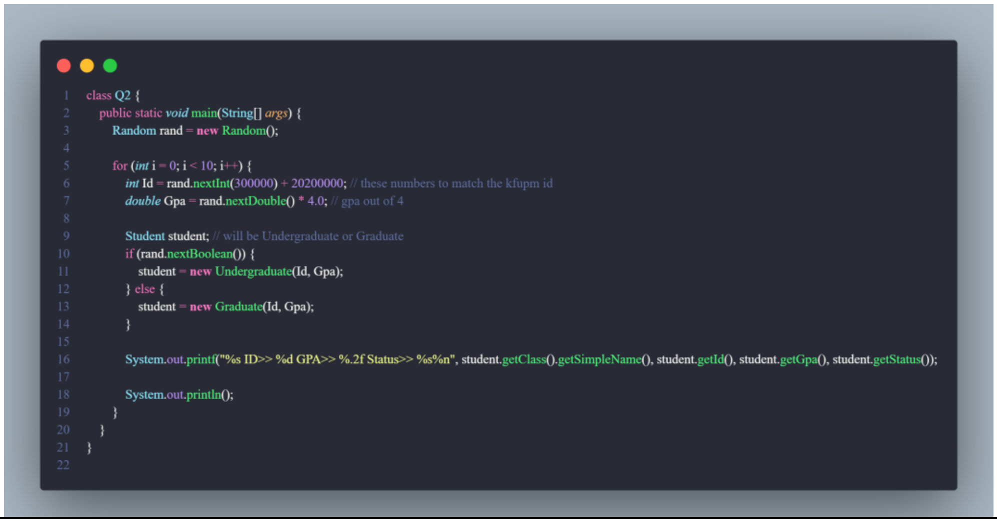
```java
public abstract class Student
{
   private int id;
   private double gpa;

   public Student(int id , double gpa)
   {
      this.id = id;
      this.gpa = gpa;
   }

   // geters because the are private
   public int getId() { return id; }
   public double getGpa() { return gpa; }



   public abstract String getStatus(); // will be implemented in the subclasses

   public String displayStudents()
   {
      return "ID>> " + getId() + " GPA>> " + getGpa() + " Status>> " + getStatus();
   }
```



```java
class Graduate extends Student
{

    public Graduate(int id , double gpa)
    {
        super(id, gpa);
    }

    @Override
    public String getStatus()
    {
        if (getGpa() >= 3.0) return "good"; // if gpa greater than or equal 3.0 the status will be good
        else return "probation";
    }
```



```java
class Undergraduate  extends Student
{

    public Undergraduate (int id , double gpa)
    {
        super(id, gpa);
    }

    @Override
    public String getStatus() // if gpa greater than or equal 3.0 the status will be honor or gpa grater than or equal 2.0 the status will be good otherwise probation
    {
        if (getGpa() >= 3.0) return "honor";
        else if(getGpa() >= 2.0) return "good";
        else return "probation";
    }
} // end o
```



```java
class Q2 {
  public static void main(String[] args) {
    Random rand = new Random();

    for (int i = 0; i < 10; i++) {
      int Id = rand.nextInt(300000) + 20200000; // these numbers to match the kfupm id
      double Gpa = rand.nextDouble() * 4.0; // gpa out of 4

      Student student; // will be Undergraduate or Graduate
      if (rand.nextBoolean()) {
        student = new Undergraduate(Id, Gpa);
      } else {
        student = new Graduate(Id, Gpa);
      }

      System.out.printf("%s ID>> %d GPA>> %.2f Status>> %s%n", student.getClass().getSimpleName(), student.getId(), student.getGpa(), student.getStatus());

      System.out.println();
    }
  }
}

```