Course Code: CSE 2104
Course Title: Object-oriented Programming Lab
Lab Work 2

SUBMITTED TO :

Shakib Mahamud Dipto
CSE
lecturer
University of Liberal Arts

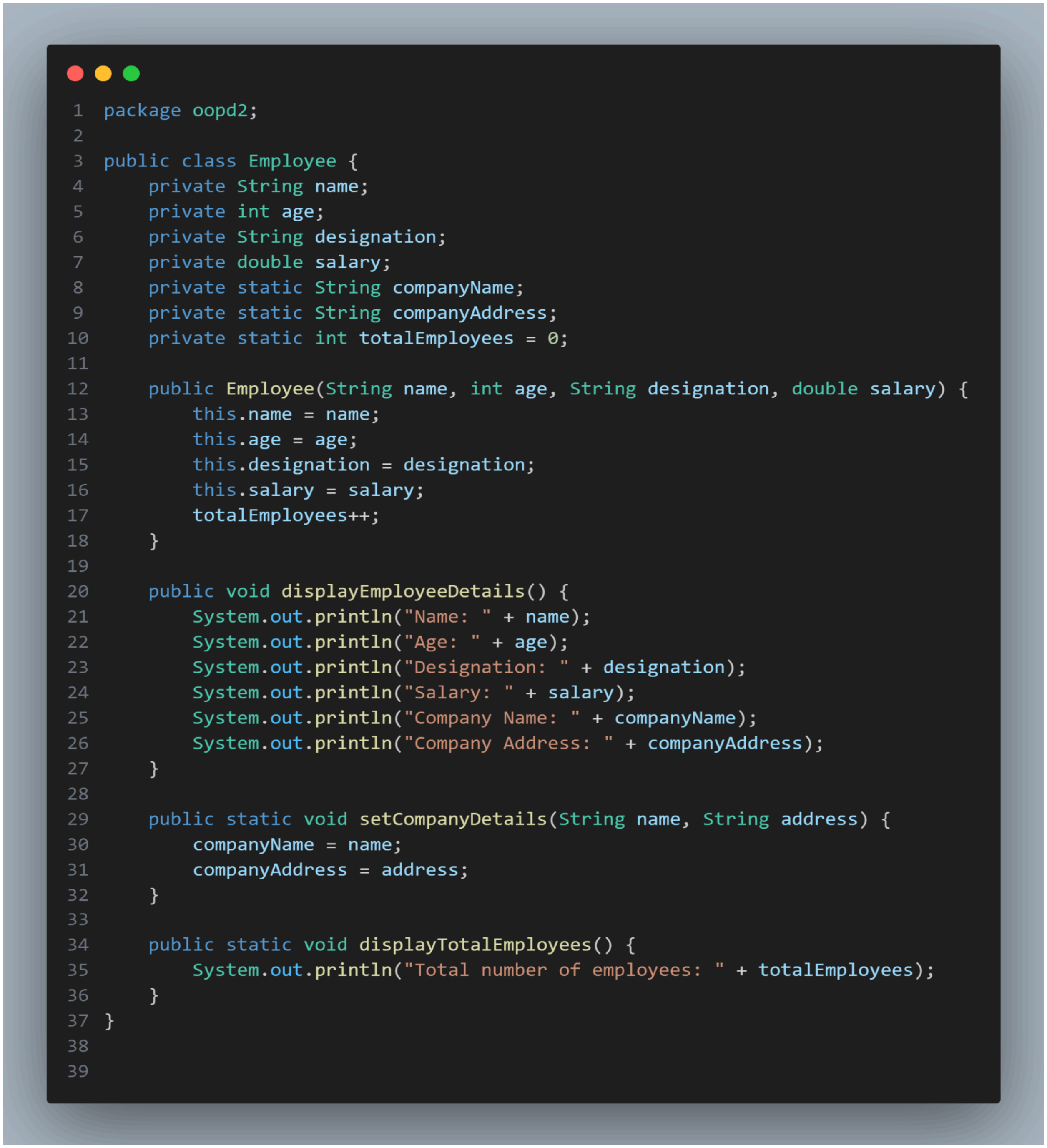
SUBMITTED BY :

Piyas sarkar
223014024
Section 4

Question1:

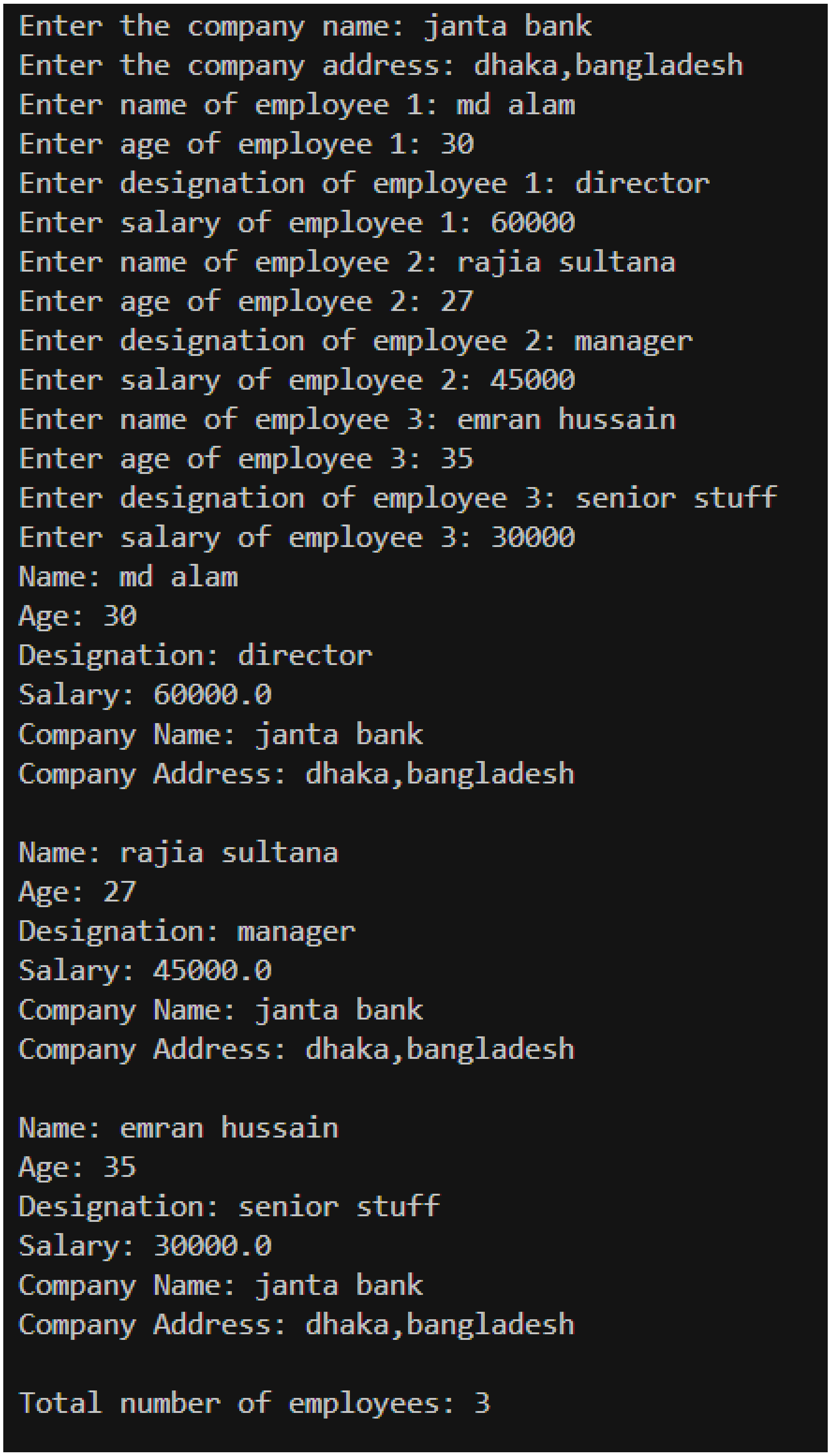
Create an Employee class with properties such as name, age, designation, salary (object variables) and company name, company address (class variables). Implement a parameterized constructor to initialize 3 objects. Include an object method to display the employee details and a class method to display the total number of employees.

Solution Code:



```java
package oopd2;

public class Employee {
    private String name;
    private int age;
    private String designation;
    private double salary;
    private static String companyName;
    private static String companyAddress;
    private static int totalEmployees = 0;

    public Employee(String name, int age, String designation, double salary) {
        this.name = name;
        this.age = age;
        this.designation = designation;
        this.salary = salary;
        totalEmployees++;
    }

    public void displayEmployeeDetails() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Designation: " + designation);
        System.out.println("Salary: " + salary);
        System.out.println("Company Name: " + companyName);
        System.out.println("Company Address: " + companyAddress);
    }

    public static void setCompanyDetails(String name, String address) {
        companyName = name;
        companyAddress = address;
    }

    public static void displayTotalEmployees() {
        System.out.println("Total number of employees: " + totalEmployees);
    }
}
```

Output:



```
Enter the company name: janta bank
Enter the company address: dhaka,bangladesh
Enter name of employee 1: md alam
Enter age of employee 1: 30
Enter designation of employee 1: director
Enter salary of employee 1: 60000
Enter name of employee 2: rajia sultana
Enter age of employee 2: 27
Enter designation of employee 2: manager
Enter salary of employee 2: 45000
Enter name of employee 3: emran hussain
Enter age of employee 3: 35
Enter designation of employee 3: senior stuff
Enter salary of employee 3: 30000
Name: md alam
Age: 30
Designation: director
Salary: 60000.0
Company Name: janta bank
Company Address: dhaka,bangladesh

Name: rajia sultana
Age: 27
Designation: manager
Salary: 45000.0
Company Name: janta bank
Company Address: dhaka,bangladesh

Name: emran hussain
Age: 35
Designation: senior stuff
Salary: 30000.0
Company Name: janta bank
Company Address: dhaka,bangladesh

Total number of employees: 3
```

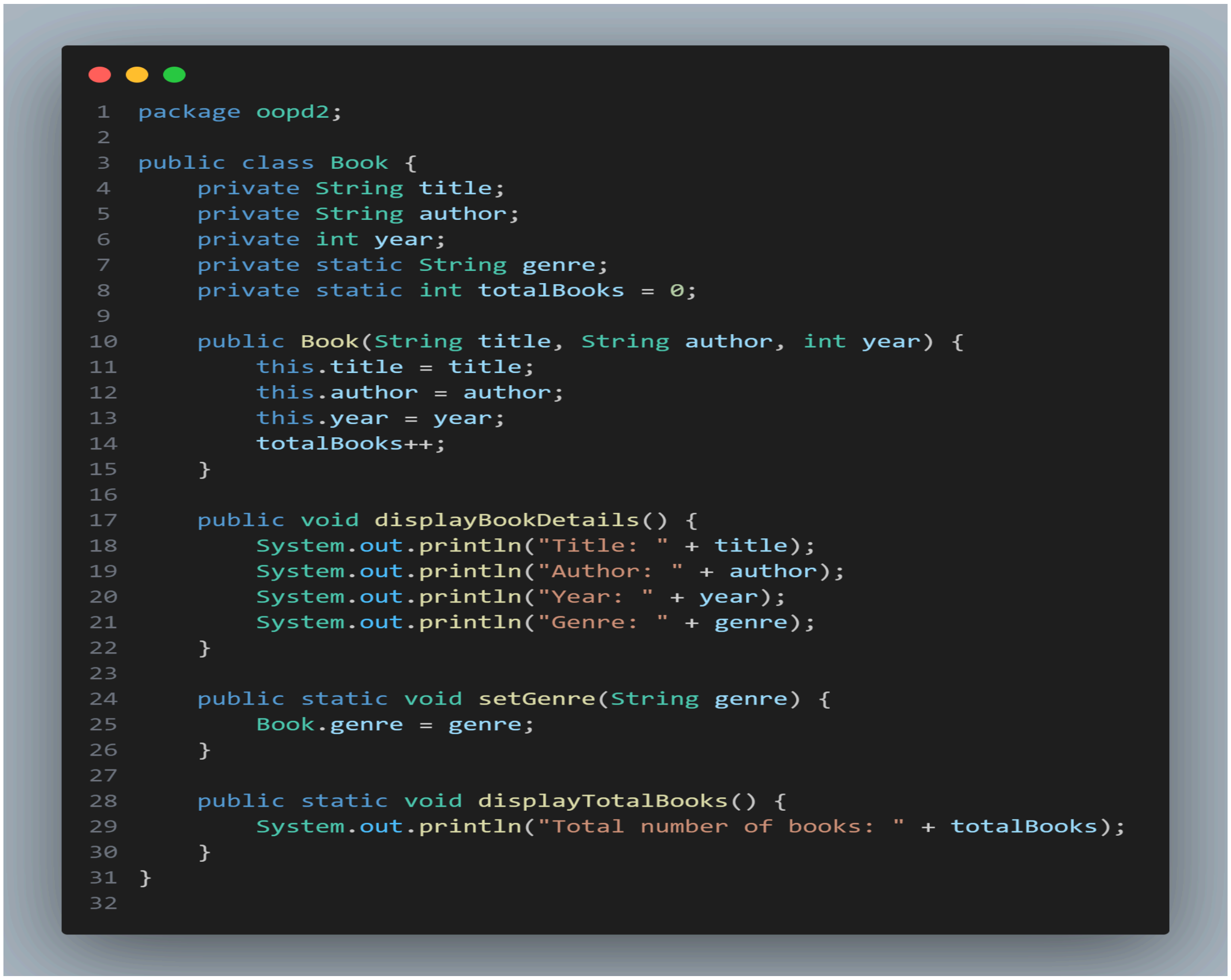
Explanation:

A worker is represented by the Employee class, which has instance variables for name, age, title, and pay. In addition, it contains two class variables that are shared by all instances of the class: companyName and companyAddress. The class maintains track of the total number of employees produced and has a parameterized constructor for initializing these attributes. While the static displayTotalEmployees method displays the overall number of employees, the displayEmployeeDetails method outputs an employee's details. This class serves as an example of how to maintain data pertaining to specific employees and their common employer by using both instance and class variables.

Question2:

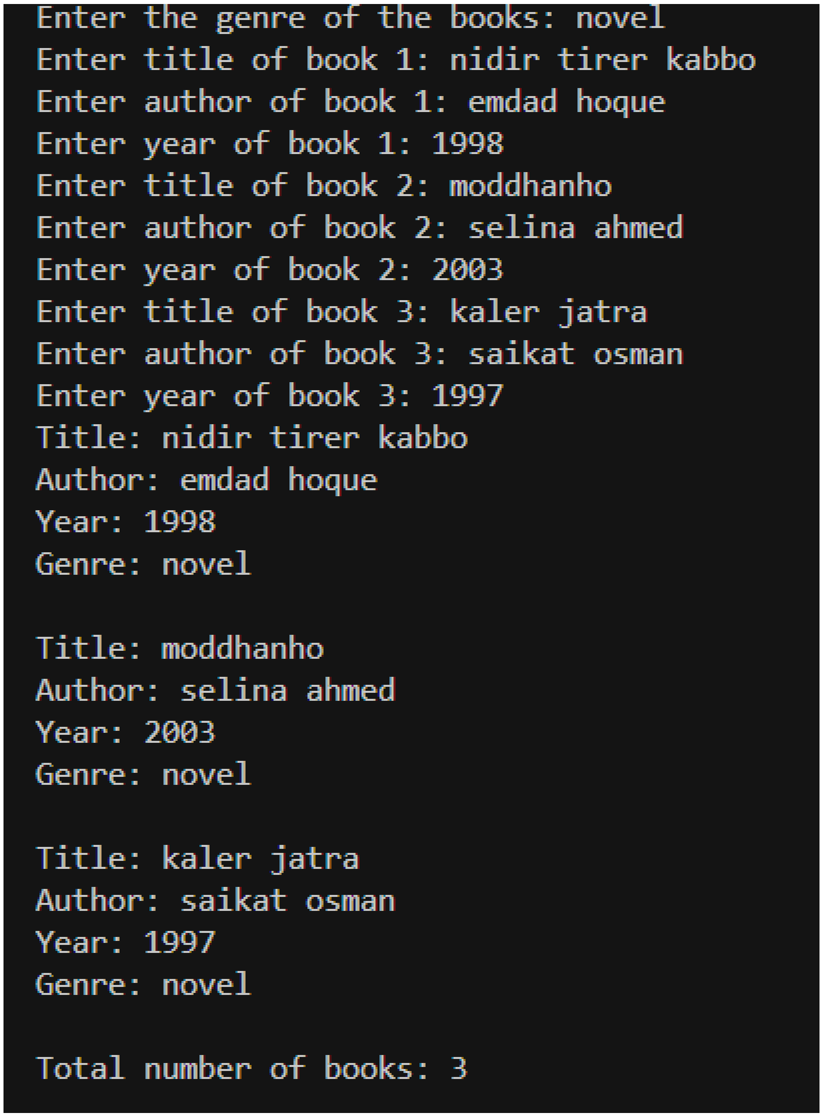
Create a Book class with properties such as title, author, year (object variables) and genre (class variable). Implement a parameterized constructor to initialize 3 objects. Include an object method to display the book details and a class method to display the total number of books.

Solution Code:



```java
package oopd2;

public class Book {
    private String title;
    private String author;
    private int year;
    private static String genre;
    private static int totalBooks = 0;

    public Book(String title, String author, int year) {
        this.title = title;
        this.author = author;
        this.year = year;
        totalBooks++;
    }

    public void displayBookDetails() {
        System.out.println("Title: " + title);
        System.out.println("Author: " + author);
        System.out.println("Year: " + year);
        System.out.println("Genre: " + genre);
    }

    public static void setGenre(String genre) {
        Book.genre = genre;
    }

    public static void displayTotalBooks() {
        System.out.println("Total number of books: " + totalBooks);
    }
}

```

Output:



```
Enter the genre of the books: novel
Enter title of book 1: nidir tirer kabbo
Enter author of book 1: emdad hoque
Enter year of book 1: 1998
Enter title of book 2: moddhanho
Enter author of book 2: selina ahmed
Enter year of book 2: 2003
Enter title of book 3: kaler jatra
Enter author of book 3: saikat osman
Enter year of book 3: 1997
Title: nidir tirer kabbo
Author: emdad hoque
Year: 1998
Genre: novel

Title: moddhanho
Author: selina ahmed
Year: 2003
Genre: novel

Title: kaler jatra
Author: saikat osman
Year: 1997
Genre: novel

Total number of books: 3
```

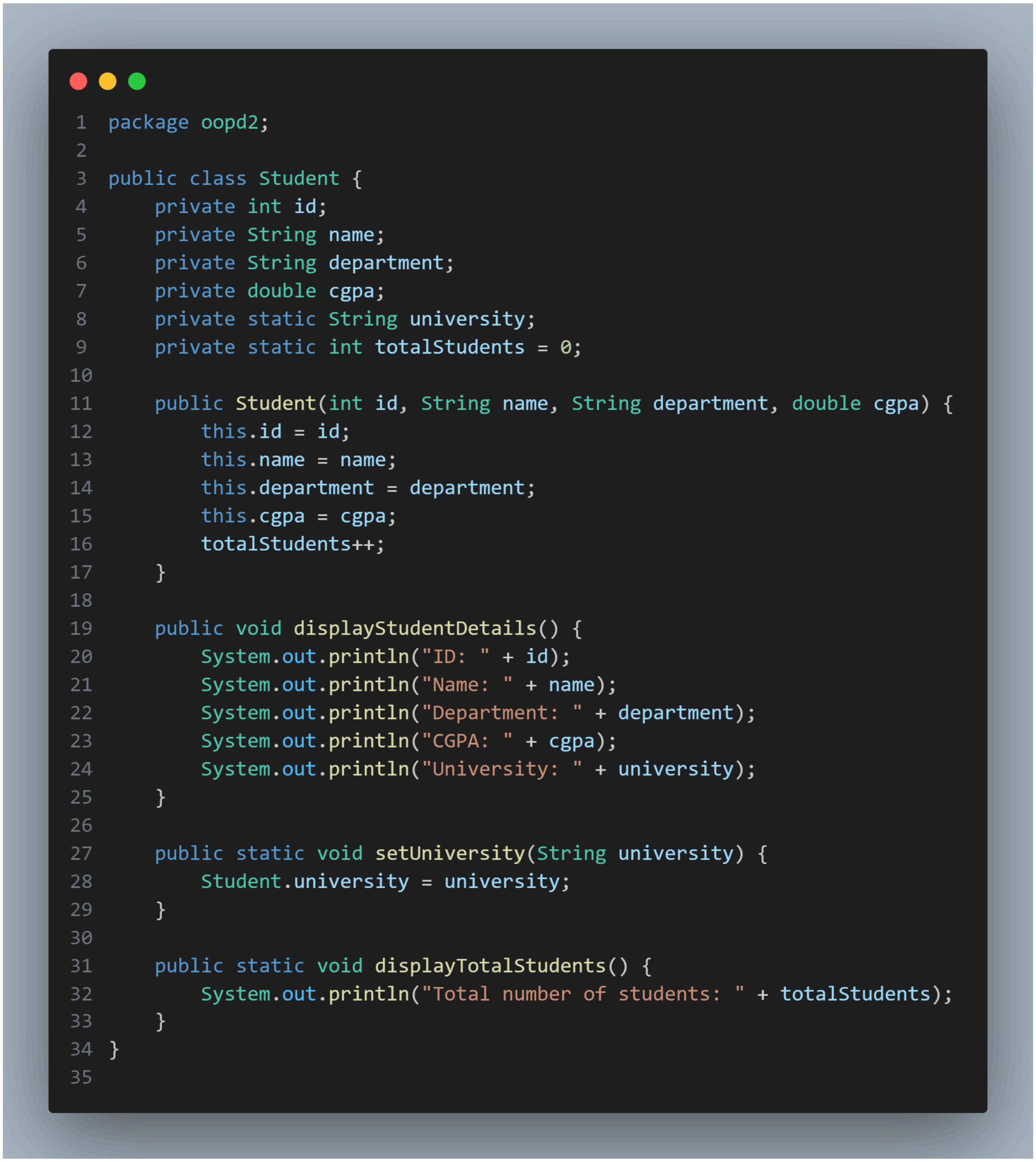
Explanation:

With characteristics unique to each book instance, such as author, year, and title, the Book class simulates a book. It also has a class variable genre, which is present in every book. The class has a static variable to record the total number of books created, and a parameterized constructor to initialize these attributes. While the static displayTotalBooks function publishes the total number of book instances, the displayBookDetails method prints a book's information. This course emphasizes how to represent both shared and unique qualities of books using static and instance components.

Question3:

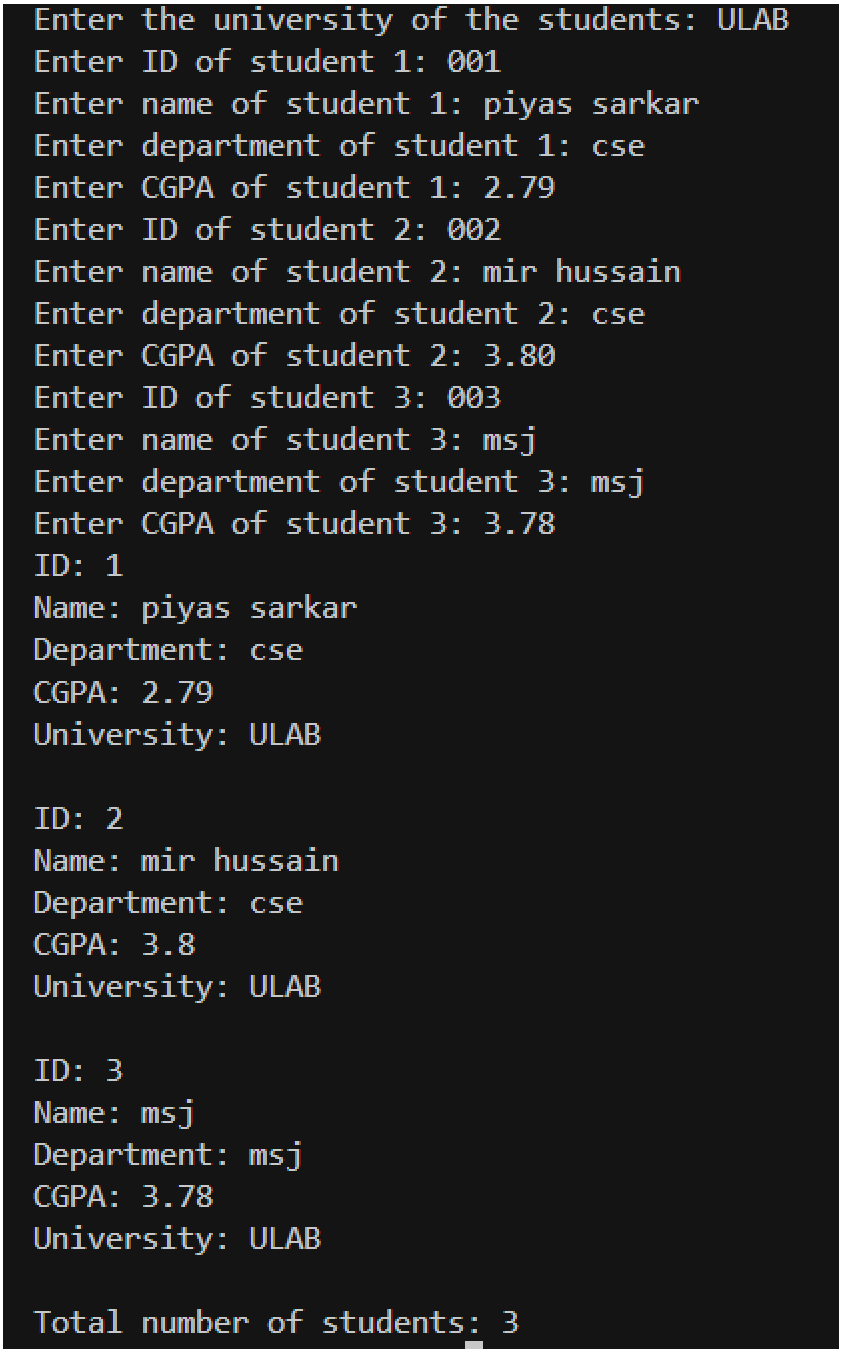
Create a Student class with properties such as id, name, department, cgpa (object variables) and university (class variable). Implement a parameterized constructor to initialize 3 objects. Include an object method to display the student details and a class method to display the total number of students.

Solution Code:



```java
package oopd2;

public class Student {
    private int id;
    private String name;
    private String department;
    private double cgpa;
    private static String university;
    private static int totalStudents = 0;

    public Student(int id, String name, String department, double cgpa) {
        this.id = id;
        this.name = name;
        this.department = department;
        this.cgpa = cgpa;
        totalStudents++;
    }

    public void displayStudentDetails() {
        System.out.println("ID: " + id);
        System.out.println("Name: " + name);
        System.out.println("Department: " + department);
        System.out.println("CGPA: " + cgpa);
        System.out.println("University: " + university);
    }

    public static void setUniversity(String university) {
        Student.university = university;
    }

    public static void displayTotalStudents() {
        System.out.println("Total number of students: " + totalStudents);
    }
}

```

Output:

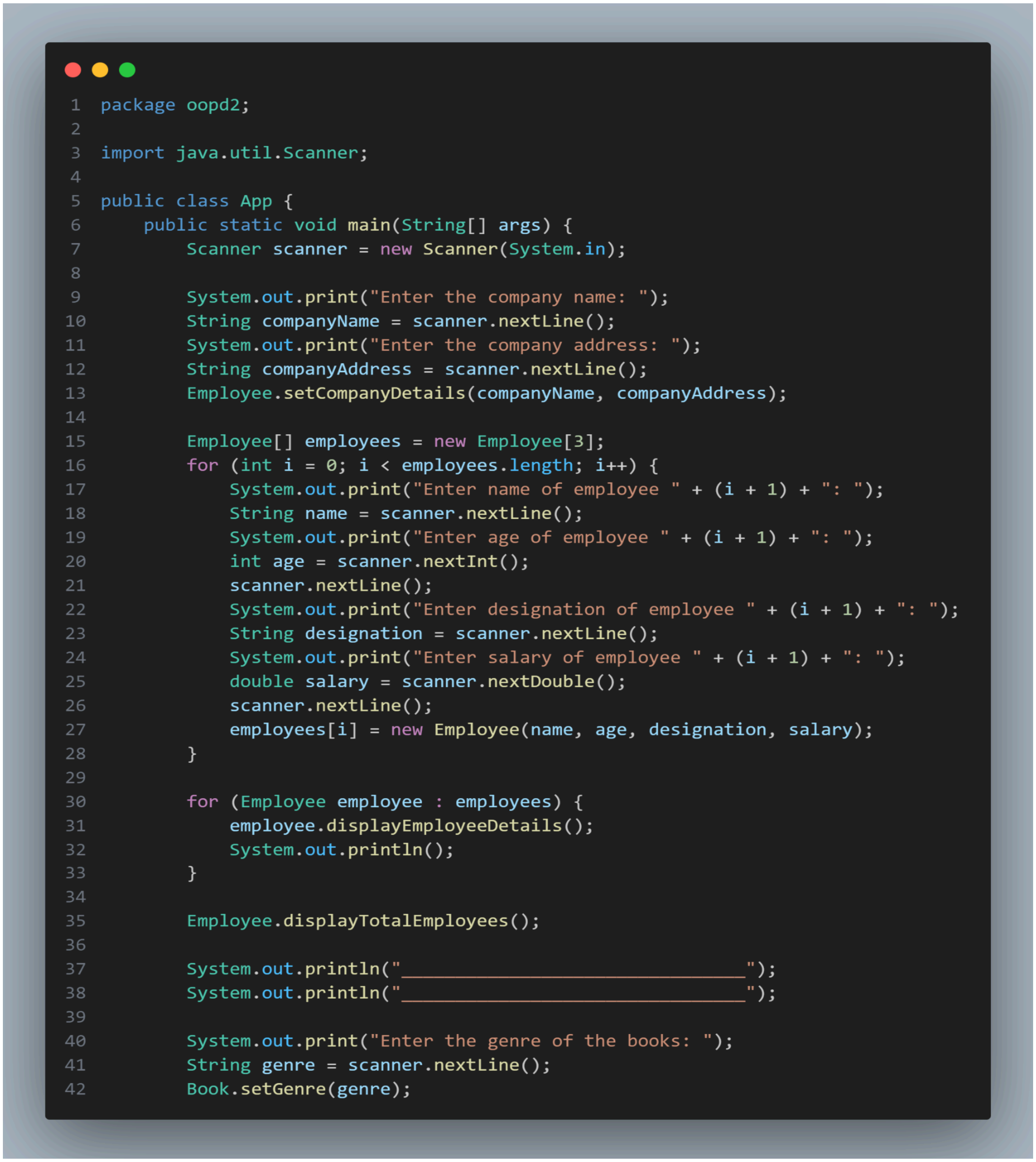


```
Enter the university of the students: ULAB
Enter ID of student 1: 001
Enter name of student 1: piyas sarkar
Enter department of student 1: cse
Enter CGPA of student 1: 2.79
Enter ID of student 2: 002
Enter name of student 2: mir hussain
Enter department of student 2: cse
Enter CGPA of student 2: 3.80
Enter ID of student 3: 003
Enter name of student 3: msj
Enter department of student 3: msj
Enter CGPA of student 3: 3.78
ID: 1
Name: piyas sarkar
Department: cse
CGPA: 2.79
University: ULAB

ID: 2
Name: mir hussain
Department: cse
CGPA: 3.8
University: ULAB

ID: 3
Name: msj
Department: msj
CGPA: 3.78
University: ULAB

Total number of students: 3
```

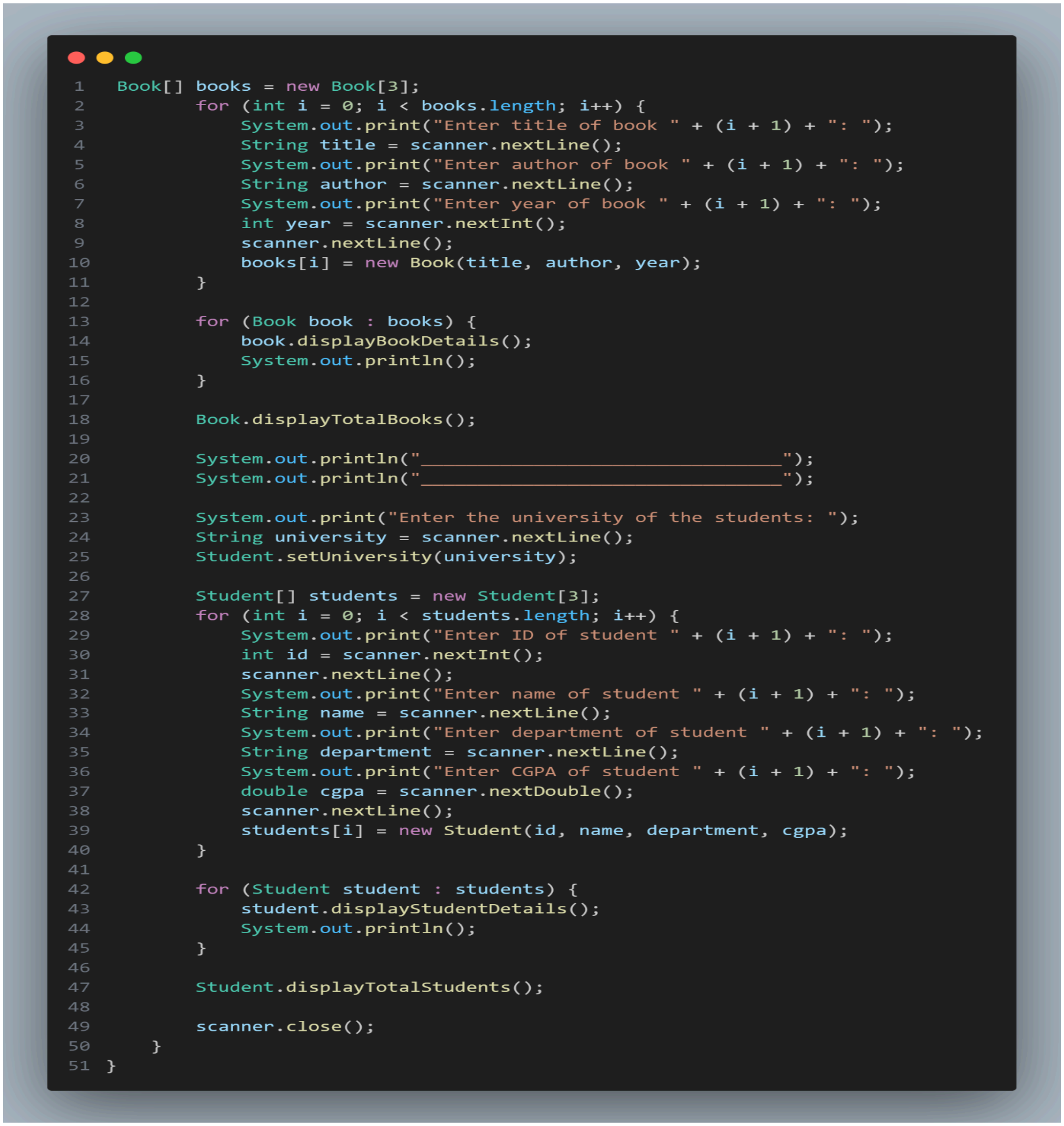
Explanation:

The Student class maintains a static variable for the university name that is shared by all student instances, in addition to encapsulating information on students, including ID, name, department, and CGPA. These fields are initialized by the parameterized constructor, which also advances a static counter that keeps track of the overall number of pupils. There are ways to show the overall number of students as well as specific student information. By encapsulating student-specific data and using static members for class-wide attributes, this class effectively manages and represents student data, demonstrating the fundamentals of object-oriented programming.

Main class:



```java
package oopd2;

import java.util.Scanner;

public class App {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the company name: ");
        String companyName = scanner.nextLine();
        System.out.print("Enter the company address: ");
        String companyAddress = scanner.nextLine();
        Employee.setCompanyDetails(companyName, companyAddress);

        Employee[] employees = new Employee[3];
        for (int i = 0; i < employees.length; i++) {
            System.out.print("Enter name of employee " + (i + 1) + ": ");
            String name = scanner.nextLine();
            System.out.print("Enter age of employee " + (i + 1) + ": ");
            int age = scanner.nextInt();
            scanner.nextLine();
            System.out.print("Enter designation of employee " + (i + 1) + ": ");
            String designation = scanner.nextLine();
            System.out.print("Enter salary of employee " + (i + 1) + ": ");
            double salary = scanner.nextDouble();
            scanner.nextLine();
            employees[i] = new Employee(name, age, designation, salary);
        }

        for (Employee employee : employees) {
            employee.displayEmployeeDetails();
            System.out.println();
        }

        Employee.displayTotalEmployees();

        System.out.println("________________________________");
        System.out.println("________________________________");

        System.out.print("Enter the genre of the books: ");
        String genre = scanner.nextLine();
        Book.setGenre(genre);
```



```java
 Book[] books = new Book[3];
        for (int i = 0; i < books.length; i++) {
            System.out.print("Enter title of book " + (i + 1) + ": ");
            String title = scanner.nextLine();
            System.out.print("Enter author of book " + (i + 1) + ": ");
            String author = scanner.nextLine();
            System.out.print("Enter year of book " + (i + 1) + ": ");
            int year = scanner.nextInt();
            scanner.nextLine();
            books[i] = new Book(title, author, year);
        }

        for (Book book : books) {
            book.displayBookDetails();
            System.out.println();
        }

        Book.displayTotalBooks();

        System.out.println("_______________________________");
        System.out.println("_______________________________");

        System.out.print("Enter the university of the students: ");
        String university = scanner.nextLine();
        Student.setUniversity(university);

        Student[] students = new Student[3];
        for (int i = 0; i < students.length; i++) {
            System.out.print("Enter ID of student " + (i + 1) + ": ");
            int id = scanner.nextInt();
            scanner.nextLine();
            System.out.print("Enter name of student " + (i + 1) + ": ");
            String name = scanner.nextLine();
            System.out.print("Enter department of student " + (i + 1) + ": ");
            String department = scanner.nextLine();
            System.out.print("Enter CGPA of student " + (i + 1) + ": ");
            double cgpa = scanner.nextDouble();
            scanner.nextLine();
            students[i] = new Student(id, name, department, cgpa);
        }

        for (Student student : students) {
            student.displayStudentDetails();
            System.out.println();
        }

        Student.displayTotalStudents();

        scanner.close();
    }
}
```

The program's entry point, the Main class, is made to coordinate the production and presentation of Employee, Book, and Student objects. This class reads user-provided input data, such as university name, book genre, and company details, using the Scanner class. After that, it builds arrays of each sort of object and initializes them with data entered by the user. In addition, the Main class calls methods that show the overall number of objects created as well as the specifics of each object, illustrating the efficient use of instance and class methods for data management and presentation.