Student Management System

BIT 2ND Semester

15th September 2023

Submitted By:

Pawanraj Ratala Joshi

LC00017001984

Submitted To:

Bishranta Bhattarai

Declaration

I hereby declare that this mini project work entitled “Student Management System” submitted to the Faculty of Science, Lincoln University, Texas College of Management and IT, Kathmandu is an original piece of work under the supervision of Mr.Bishranta and is submitted in partial fulfillment of the requirements for the degree of Bachelor of Information Technology (BIT). This project has not been submitted to any other university or institution to date. Thank you!

Signature:

Name of Student: Pawanraj Ratala Joshi

Date: 15th September 2023

Abstract

The following report provides a comprehensive overview of the 2^{nd} Sem mini project of Student Management System, an application using Java programming language. This project covers the specific details of the project's research. This document explores the development of a comprehensive Student Management System using the Java Programming language. This application addresses the complex challenges inherent in academic administration, providing administrators, instructors, and students.

Acknowledgement

I appreciate the opportunity that is provided by Lincoln University. I also want to show my sincere gratitude to the Texas College of Management and IT for giving me the opportunity to complete my mini project in this topic and develop my skills further.

1. Project Supervisor

I owe a deep sense of gratitude to our project supervisor Mr. Bishranta Bhattarai for his guidance and constraints. His expertise in this topic and his guidance has been very helpful towards properly shaping my management system.

2. Special Thanks

I wish to express my gratitude to Dr.Suman Thapaliya Sir, the Head of our Department, for offering us this opportunity and allocating resources for this project.

Table of Contents

Aim and Objectives

The main objectives of the Mini Console-Based Student Management System project are as follows:

1. **Efficient Student Information Management**

   To centralize the student data, including personal details, their enrollments records and the subjects they are enrolled in.

2. **Financial Balance Tracking**

   It provides a platform where you can track student's financial balances, including tuition fees, scholarships they have received and payments wherever they have done.

3. **User Roles**

   It can be used for various roles such as administrators, admission officers, financial officers, and students too. Each can be provided with specific permission and access levels.

Tools to be Used.

To develop the Mini Console-Based Student Management System, I utilized the following tools and Technology:

1. **Programming Language**

   I have used Java programming language to create the student management system because of its higher cross functionality.

2. **Development Environment**

   I have used IntelliJ IDEA for coding. IntelliJ is primarily used for Java programming, but it also supports other wide range of programming languages.

3. **For Documentation**

   Microsoft Word is used to prepare my documentation and put together all the components to complete my project and submit it.

Introduction

My Student Management System can be an addition to the university digital infrastructure which was made using ‘Java Programming Language’ in the development environment of named IntelliJ IDEA.

This system manages the enrollment of new students, the year they want to study in, the courses they want to take and the payment they have made for their courses. With this tool, we can evaluate educational experiences, ensuring that the university can keep up their academic excellence. Look out to other sections to know how this application will transform the way we manage student records and finances.

Features and Functionality

1. Student Registration

This application allows administrators to add new students to the system by entering their details such as name and the year they want to enrolled in. Each student is assigned a unique student ID.

2. Course Management

Admins can create and manage courses offered by the university. They can specify details like course name and semester.

3. Enrollments and Tuition Management

This feature allows admins and administrators to enroll students in the specific year they are enrolled, which courses they have selected and how much payment they have made for that course. This system can manage all these details.

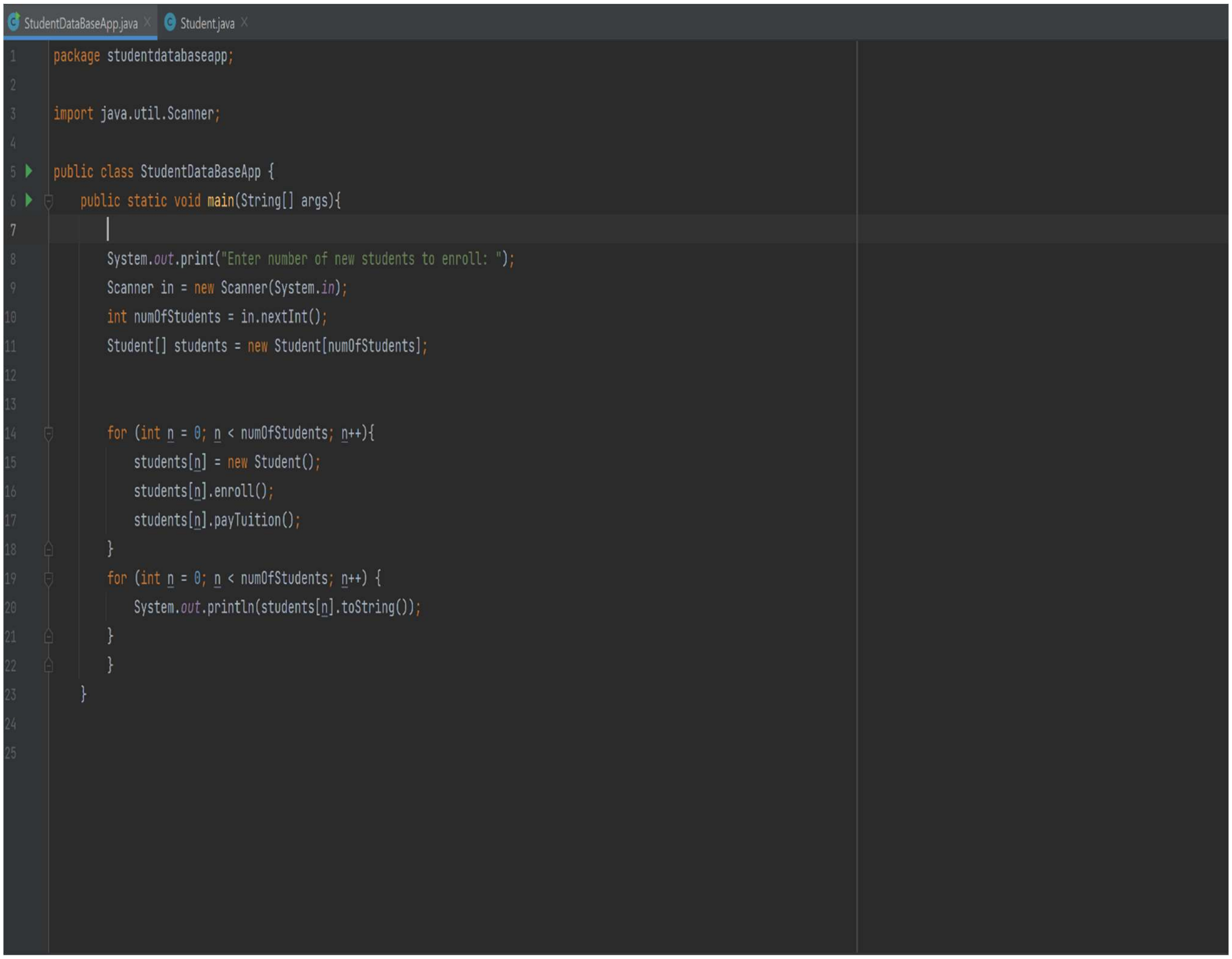
Hierarchy Structure of Project

- **StudentDataBaseApp.java**
  - **Student.java**

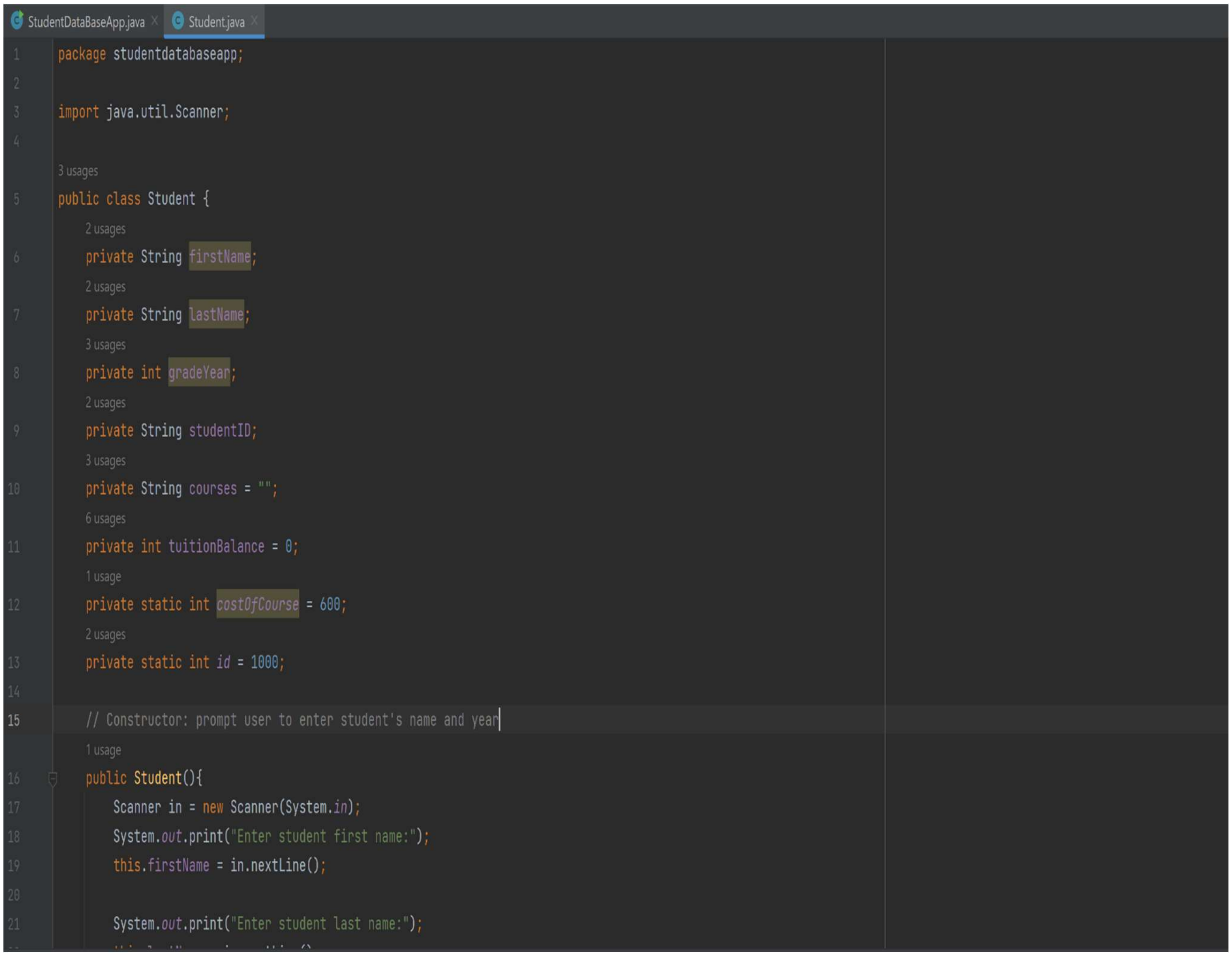
Implementation Details

a. Source Code

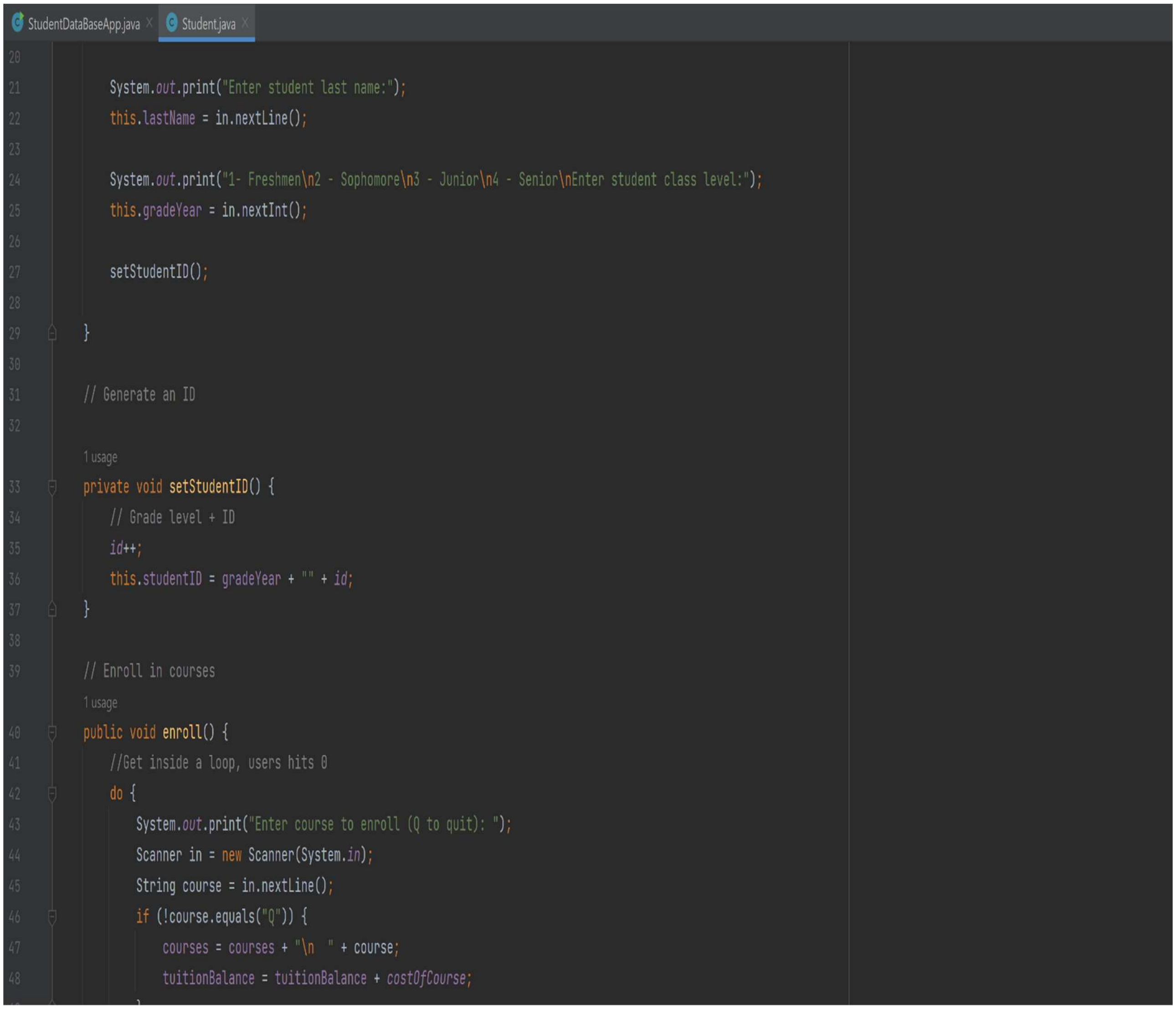
In the part, we will look at the implementation of source code for our project.



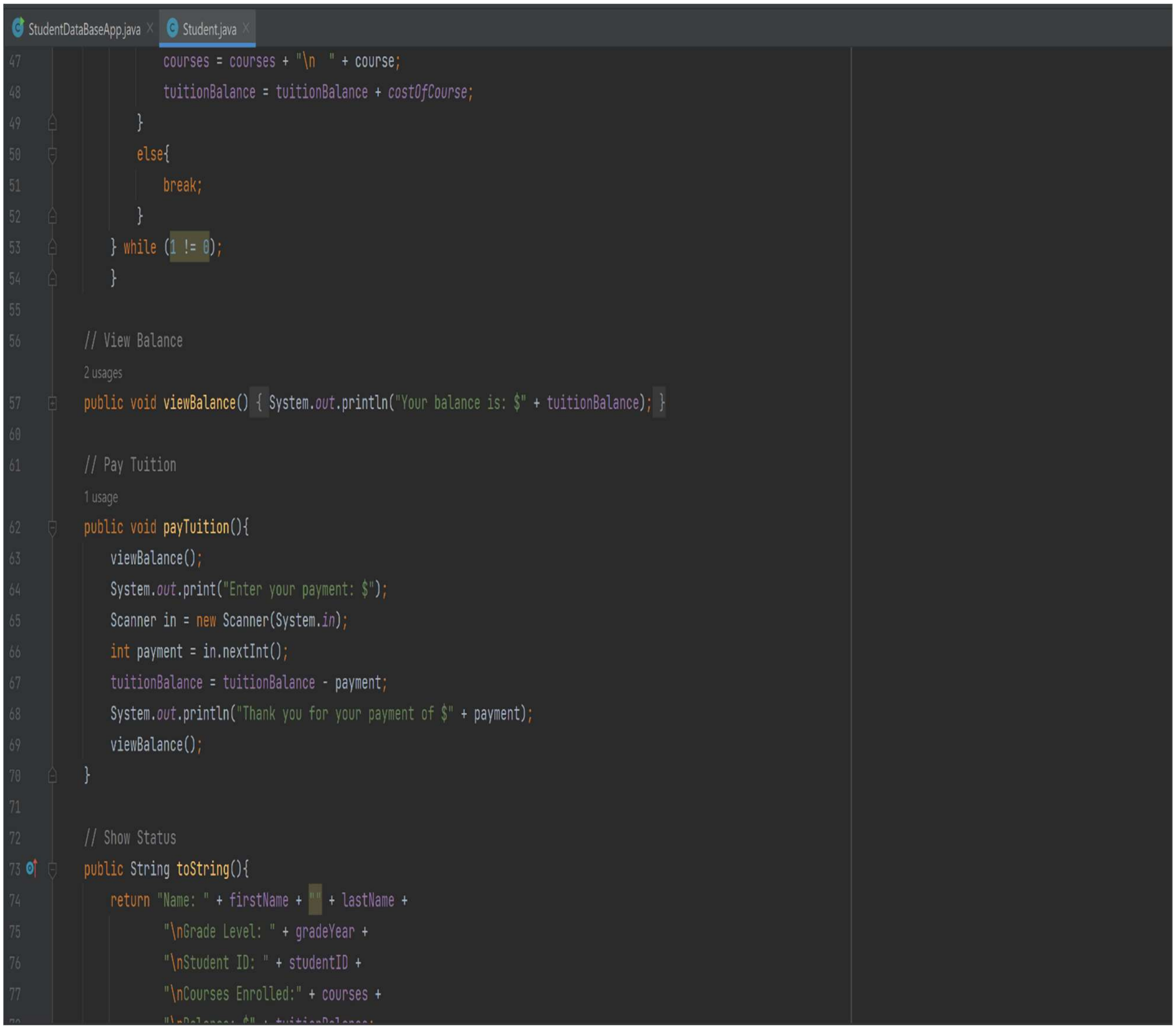
```
package studentdatabaseapp;

import java.util.Scanner;

public class StudentDataBaseApp {
    public static void main(String[] args){

        System.out.print("Enter number of new students to enroll: ");
        Scanner in = new Scanner(System.in);
        int numOfStudents = in.nextInt();
        Student[] students = new Student[numOfStudents];


        for (int n = 0; n < numOfStudents; n++){
            students[n] = new Student();
            students[n].enroll();
            students[n].payTuition();
        }
        for (int n = 0; n < numOfStudents; n++) {
            System.out.println(students[n].toString());
        }
        }
    }
```



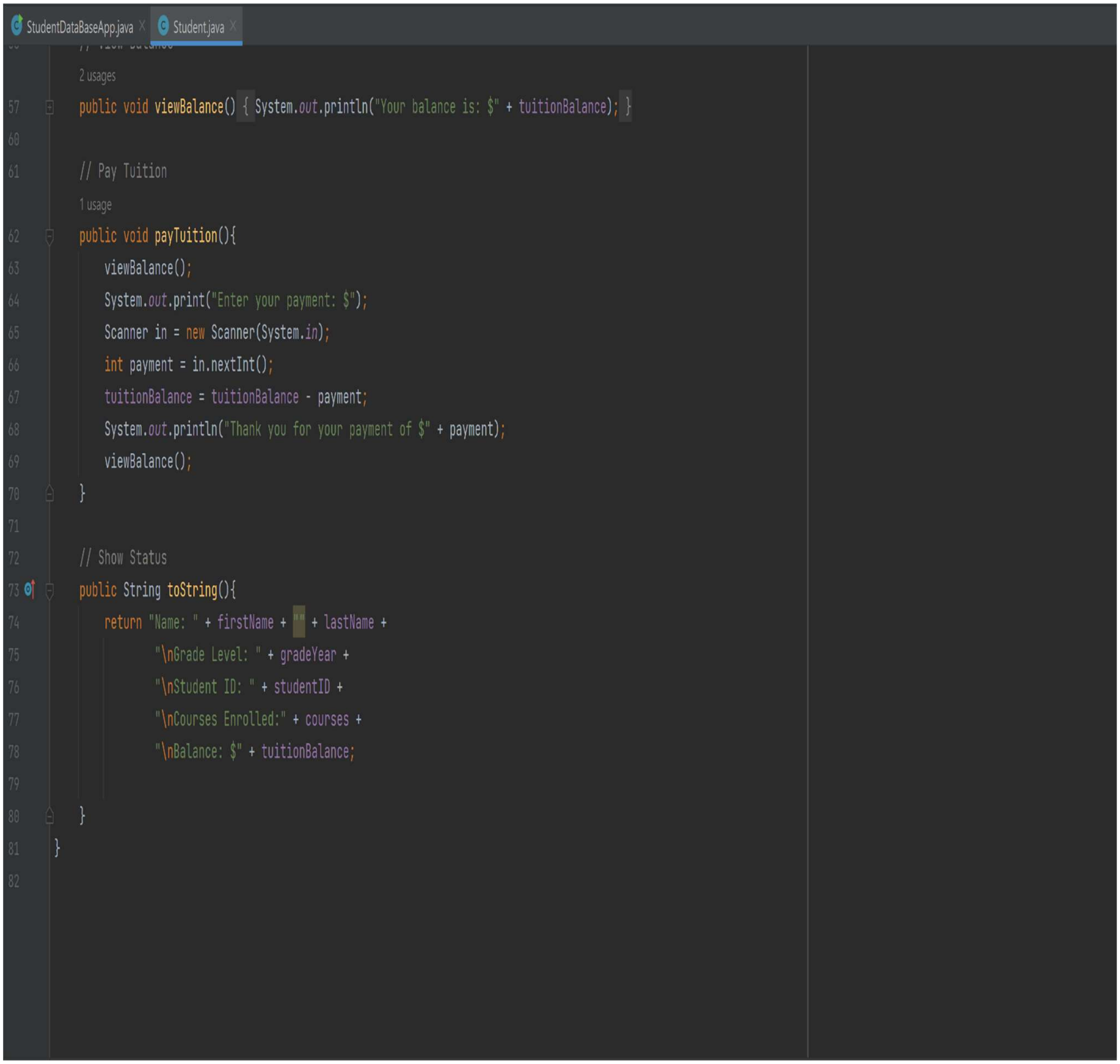
```
package studentdatabaseapp;

import java.util.Scanner;

3 usages
public class Student {
    2 usages
    private String firstName;
    2 usages
    private String lastName;
    3 usages
    private int gradeYear;
    2 usages
    private String studentID;
    3 usages
    private String courses = "";
    6 usages
    private int tuitionBalance = 0;
    1 usage
    private static int costOfCourse = 600;
    2 usages
    private static int id = 1000;

    // Constructor: prompt user to enter student's name and year
    1 usage
    public Student(){
        Scanner in = new Scanner(System.in);
        System.out.print("Enter student first name:");
        this.firstName = in.nextLine();

        System.out.print("Enter student last name:");
```



```java

        System.out.print("Enter student last name:");
        this.lastName = in.nextLine();

        System.out.print("1- Freshmen\n2 - Sophomore\n3 - Junior\n4 - Senior\nEnter student class level:");
        this.gradeYear = in.nextInt();

        setStudentID();

    }

    // Generate an ID

    1 usage
    private void setStudentID() {
        // Grade level + ID
        id++;
        this.studentID = gradeYear + "" + id;
    }

    // Enroll in courses
    1 usage
    public void enroll() {
        //Get inside a loop, users hits 0
        do {
            System.out.print("Enter course to enroll (Q to quit): ");
            Scanner in = new Scanner(System.in);
            String course = in.nextLine();
            if (!course.equals("Q")) {
                courses = courses + "\n  " + course;
                tuitionBalance = tuitionBalance + costOfCourse;
```

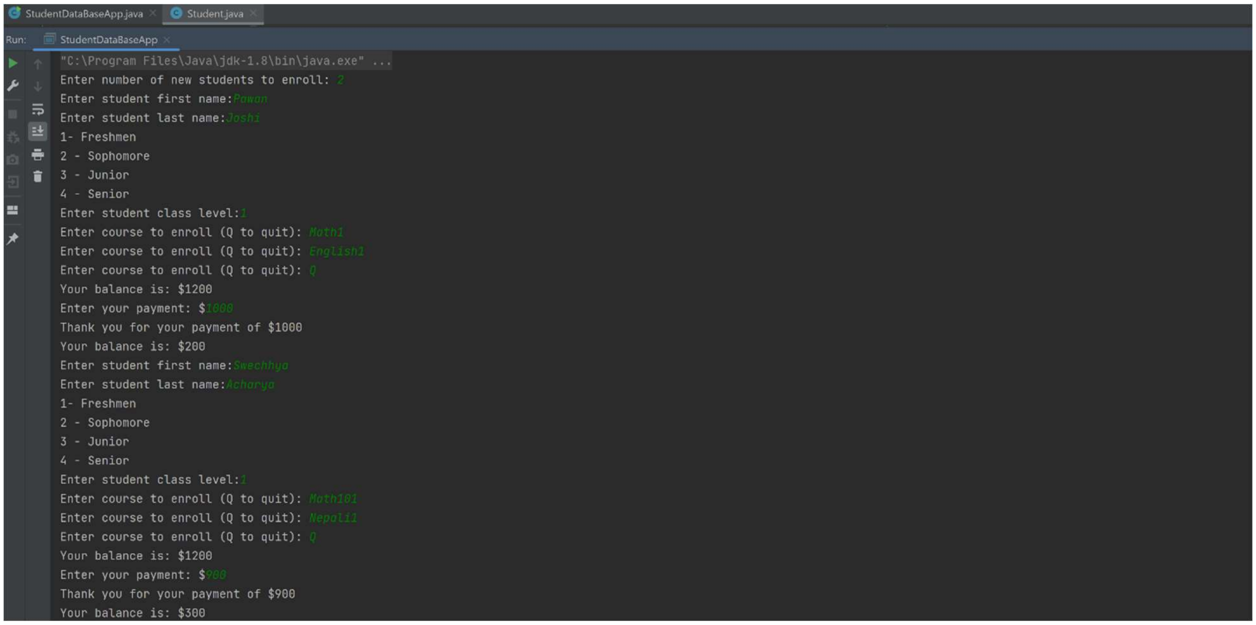


```
                courses = courses + "\n  " + course;
                tuitionBalance = tuitionBalance + costOfCourse;
            }
            else{
                break;
            }
        } while (1 != 0);
        }

    // View Balance
    public void viewBalance() { System.out.println("Your balance is: $" + tuitionBalance); }

    // Pay Tuition
    public void payTuition(){
        viewBalance();
        System.out.print("Enter your payment: $");
        Scanner in = new Scanner(System.in);
        int payment = in.nextInt();
        tuitionBalance = tuitionBalance - payment;
        System.out.println("Thank you for your payment of $" + payment);
        viewBalance();
    }

    // Show Status
    public String toString(){
        return "Name: " + firstName + " " + lastName +
                "\nGrade Level: " + gradeYear +
                "\nStudent ID: " + studentID +
                "\nCourses Enrolled:" + courses +
                "\nBalance: $" + tuitionBalance;
```

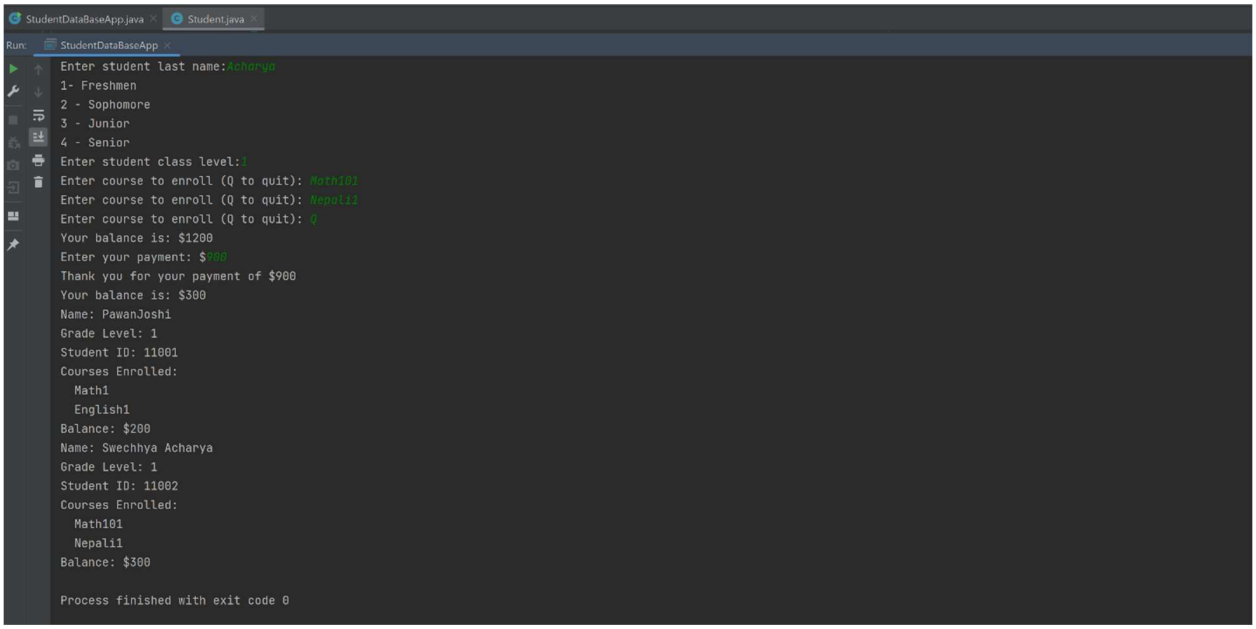


```
    2 usages
    public void viewBalance() { System.out.println("Your balance is: $" + tuitionBalance); }

    // Pay Tuition
    1 usage
    public void payTuition(){
        viewBalance();
        System.out.print("Enter your payment: $");
        Scanner in = new Scanner(System.in);
        int payment = in.nextInt();
        tuitionBalance = tuitionBalance - payment;
        System.out.println("Thank you for your payment of $" + payment);
        viewBalance();
    }

    // Show Status
    public String toString(){
        return "Name: " + firstName + "" + lastName +
                "\nGrade Level: " + gradeYear +
                "\nStudent ID: " + studentID +
                "\nCourses Enrolled:" + courses +
                "\nBalance: $" + tuitionBalance;

    }
}
```

Output



```
"C:\Program Files\Java\jdk-1.8\bin\java.exe" ...
Enter number of new students to enroll: 2
Enter student first name:Pawan
Enter student last name:Joshi
1- Freshmen
2 - Sophomore
3 - Junior
4 - Senior
Enter student class level:1
Enter course to enroll (Q to quit): Math1
Enter course to enroll (Q to quit): English1
Enter course to enroll (Q to quit): Q
Your balance is: $1200
Enter your payment: $1000
Thank you for your payment of $1000
Your balance is: $200
Enter student first name:Swechhya
Enter student last name:Acharya
1- Freshmen
2 - Sophomore
3 - Junior
4 - Senior
Enter student class level:1
Enter course to enroll (Q to quit): Math101
Enter course to enroll (Q to quit): Nepali1
Enter course to enroll (Q to quit): Q
Your balance is: $1200
Enter your payment: $900
Thank you for your payment of $900
Your balance is: $300
```



```
Enter student last name:Acharya
1- Freshmen
2 - Sophomore
3 - Junior
4 - Senior
Enter student class level:1
Enter course to enroll (Q to quit): Math101
Enter course to enroll (Q to quit): Nepali1
Enter course to enroll (Q to quit): Q
Your balance is: $1200
Enter your payment: $900
Thank you for your payment of $900
Your balance is: $300
Name: PawanJoshi
Grade Level: 1
Student ID: 11001
Courses Enrolled:
  Math1
  English1
Balance: $200
Name: Swechhya Acharya
Grade Level: 1
Student ID: 11002
Courses Enrolled:
  Math101
  Nepali1
Balance: $300

Process finished with exit code 0
```