Southeast University

Department of Computer Science and Engineering (CSE)

School of Sciences and Engineering

Semester: (Summer, Year: 2025)

LAB REPORT NO: **03**

Course Title: Introduction to Programming Language II (Java) Lab

Course Code: CSE282.2

Batch: 65

Lab Experiment Name: Introducing Input/ Output operations in JAVA.

Student Details

	Name	ID
1.	Md. Mahmud Hossain	2023200000799

Submission Date : **06-08-25**

Course Teacher's Name : **Dr. Mohammed Ashikur Rahman**

Lab Report Status

Marks: …… **Signature:.....**

Comments:..... **Date:.....**

Lab Task 3: Introducing Input/ Output operations in JAVA.

OBJECTIVES:

- To be familiar with Scanner class
- To learn how to take input from user for different data types
- To learn how to create objects after taking input from user

PROBLEM:

1. 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 (Get the object variables as input from the user). Include an object method to display the book details and a class method to display the total number of books.

2. 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 (Get the object variables as input from the user). Include an object method to display the student details and a class method to display the total number of students.

Solution:

1.

Problem Analysis:

The objective of this program is to define a Book class where the book details—title, author, and year—are taken from the user using the Scanner class. These are stored as object variables. The genre is shared by all books, so it is declared as a static class variable. A parameterized constructor initializes the book objects. An object method is used to display individual book details, and a class method shows the total number of books created.

Background Theory:

- **Object Variables:** These are unique to each object (e.g., title, author, year) and store individual Book information.
- **Class Variables (static):** genre is common to all book objects.
- **Scanner Class:** Used for taking input from the user (nextLine(), nextInt()).
- **Parameterized Constructor:** A special method used to initialize object variables with specific values when an object is created.
- **Object Methods:** These are regular methods that work with object-specific data, such as displaying a book's details.
- **Class Methods:** Returns the total number of books created.
- main() **Method:** Acts as the entry point of the program, where objects are created, methods are called, and execution begins.

Algorithm Design:

1. Define a Book class.
2. Declare a static class variable genre and totalBook.
3. Declare object variables: title, author, and year.
4. Create a parameterized constructor to initialize object variables and increment totalBook.
5. Create an object method displayDetails() to show book info.
6. Create a static method getTotalBook() to return the total book count.
7. In the main() method:
    - Assign a value to the class variable genre.
    - Use the Scanner class to take input for 3 books.
    - Create book objects with the input
    - Display each book’s details.
    - Display the total number of books.

Code:

```
import java.util.*;

public class Book {
   private static int count = 0;
   private static String genre;

   private String title;
   private String author;
```

```
private int year;


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


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


public static int getCount() {
   return count;
}

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

   System.out.print("Enter Genre: ");
   genre = sc.nextLine();

   Book[] books = new Book[3];

   for (int i = 0; i < 3; i++) {
      System.out.println("\nEnter details for Book " + (i + 1) + ":");

      System.out.print("Enter Title: ");
      String title = sc.nextLine();

      System.out.print("Enter Author: ");
      String author = sc.nextLine();

      System.out.print("Enter Year: ");
      int year = sc.nextInt();
      sc.nextLine();

      books[i] = new Book(title, author, year);
```

```
        }

        for (int i = 0; i < 3; i++) {
            System.out.println("\nBook " + (i + 1) + " Details:");
            books[i].displayDetails();
        }

        System.out.println("\nTotal number of Books: " + Book.getCount());
        sc.close();
    }
}
```

Output:

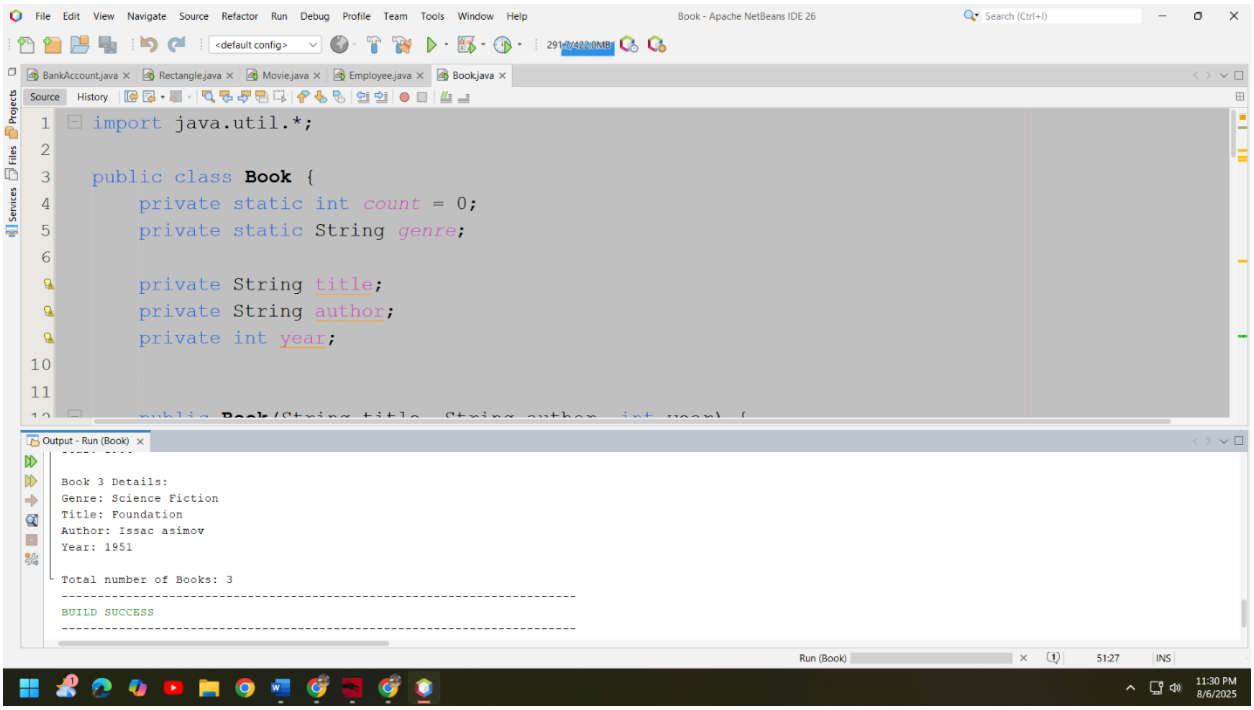

Figure 1: Output

2.

Problem Analysis:

This program creates a Student class to demonstrate input handling using the Scanner class. Each student has an id, name, department, and cgpa (object variables), while the university name is the same for all and is declared as a static variable. A parameterized constructor is used for initialization, an object method to show individual details, and a class method to return the total number of students.

Background Theory:

- **Object Variables:** Used to store individual student information.
- **Class Variable (static):** Shared variable like university.
- **Scanner Class:** Allows input from the user at runtime.
- **Constructor:** Initializes object variables.
- **Object Method:** Used to print each student's data.
- **Class Method:** Used to print total number of students created.

Algorithm Design:

1. Create a Student class.
2. Declare static variables university and count.
3. Declare object variables: id, name, department, cgpa.
4. Implement a parameterized constructor to initialize object data and increment the count.
5. Create an object method displayDetails() to show student info.
6. Create a static method getTotalStudents() to return the total number of students.
7. In the main() method:
    a. Set the value of the class variable university.
    b. Take input for 3 students using the Scanner.
    c. Create 3 student objects.
    d. Display details of each student.
    e. Print the total number of students.

Code:

```
import java.util.*;

public class Student {
   private int id;
   private String name;
   private String department;
   private double cgpa;
```

```
static String university;
private static int count = 0;

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

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

static int getTotalStudents() {
    return count;
}

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

    System.out.print("Enter University Name: ");
    university = sc.nextLine();

    Student[] students = new Student[3];

    for (int i = 0; i < 3; i++) {
        System.out.println("\nEnter details for Student " + (i + 1) + ":");

        System.out.print("ID: ");
        int id = sc.nextInt();
        sc.nextLine();

        System.out.print("Name: ");
```

```
            String name = sc.nextLine();

            System.out.print("Department: ");
            String department = sc.nextLine();

            System.out.print("CGPA: ");
            double cgpa = sc.nextDouble();
            sc.nextLine();

            students[i] = new Student(id, name, department, cgpa);
        }

        for (int i = 0; i < 3; i++) {
            System.out.println("\nStudent " + (i + 1) + " Details:");
            students[i].displayDetails();
        }

        System.out.println("Total number of students: " + getTotalStudents());
        sc.close();
    }
}
```

Output:

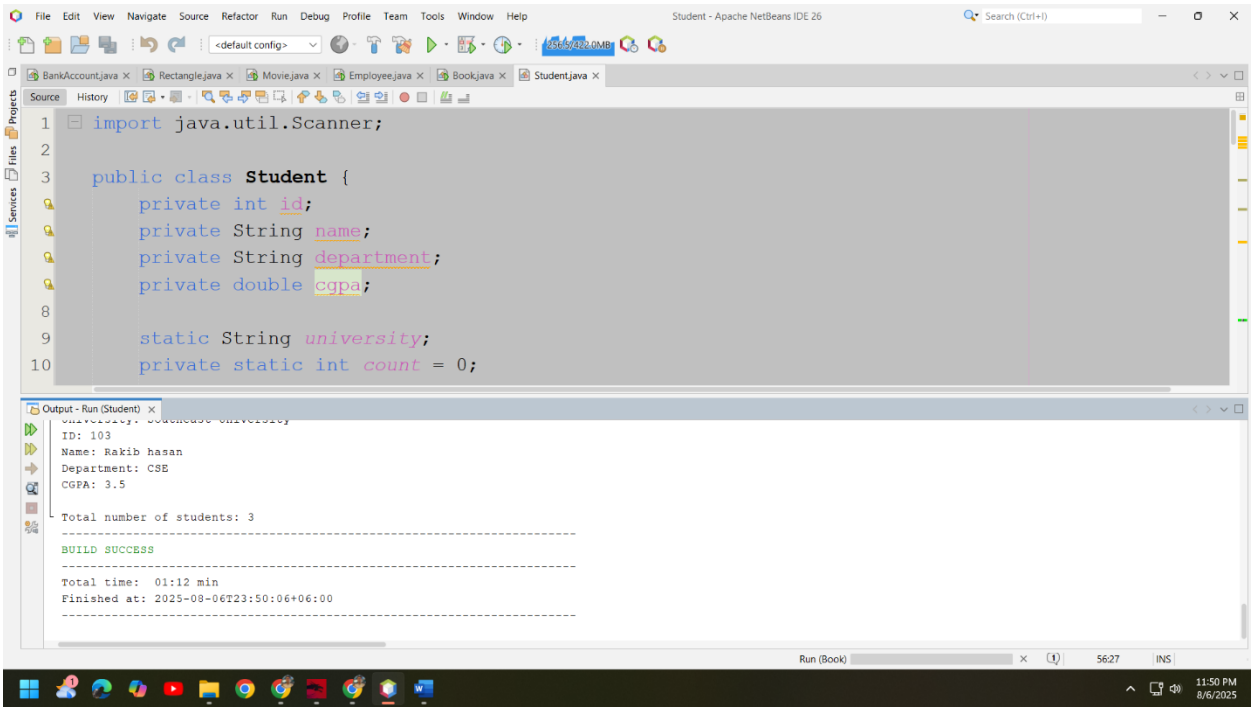

Figure 2: Output