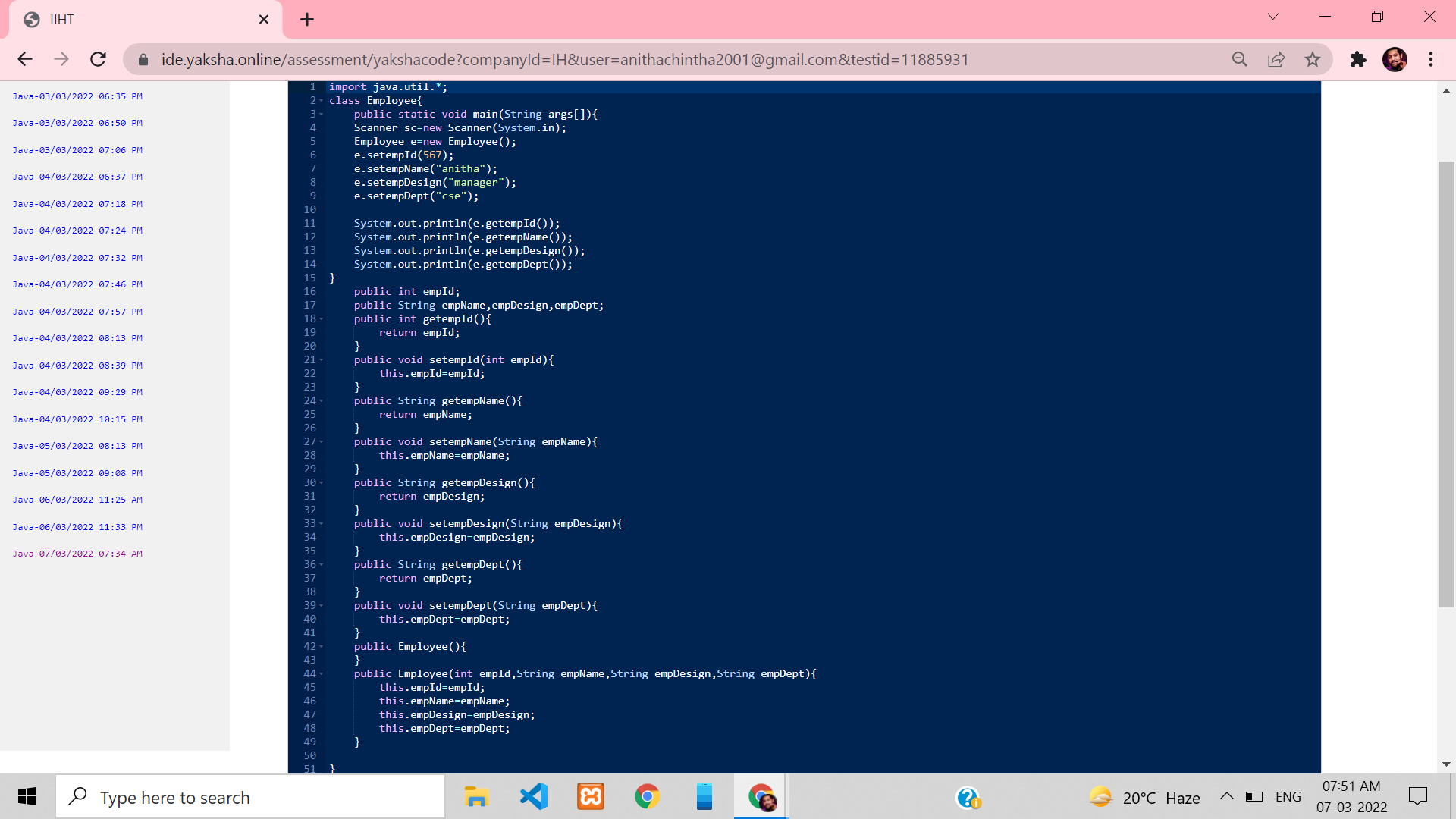
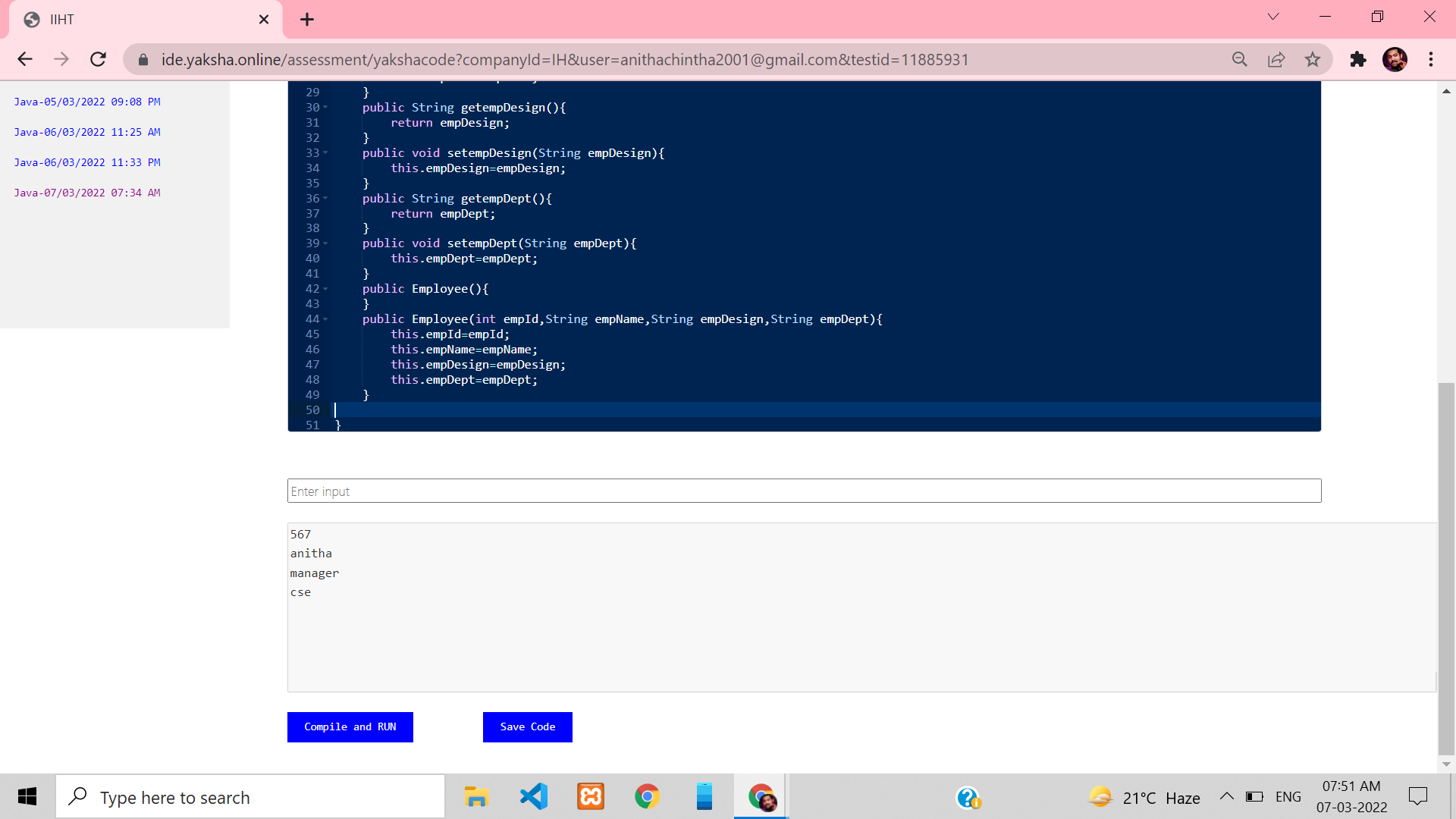
**~Assignment 5**

**Design and implement applications using basic OOP paradigms.**

**Write a program as below guideline:**

****

****

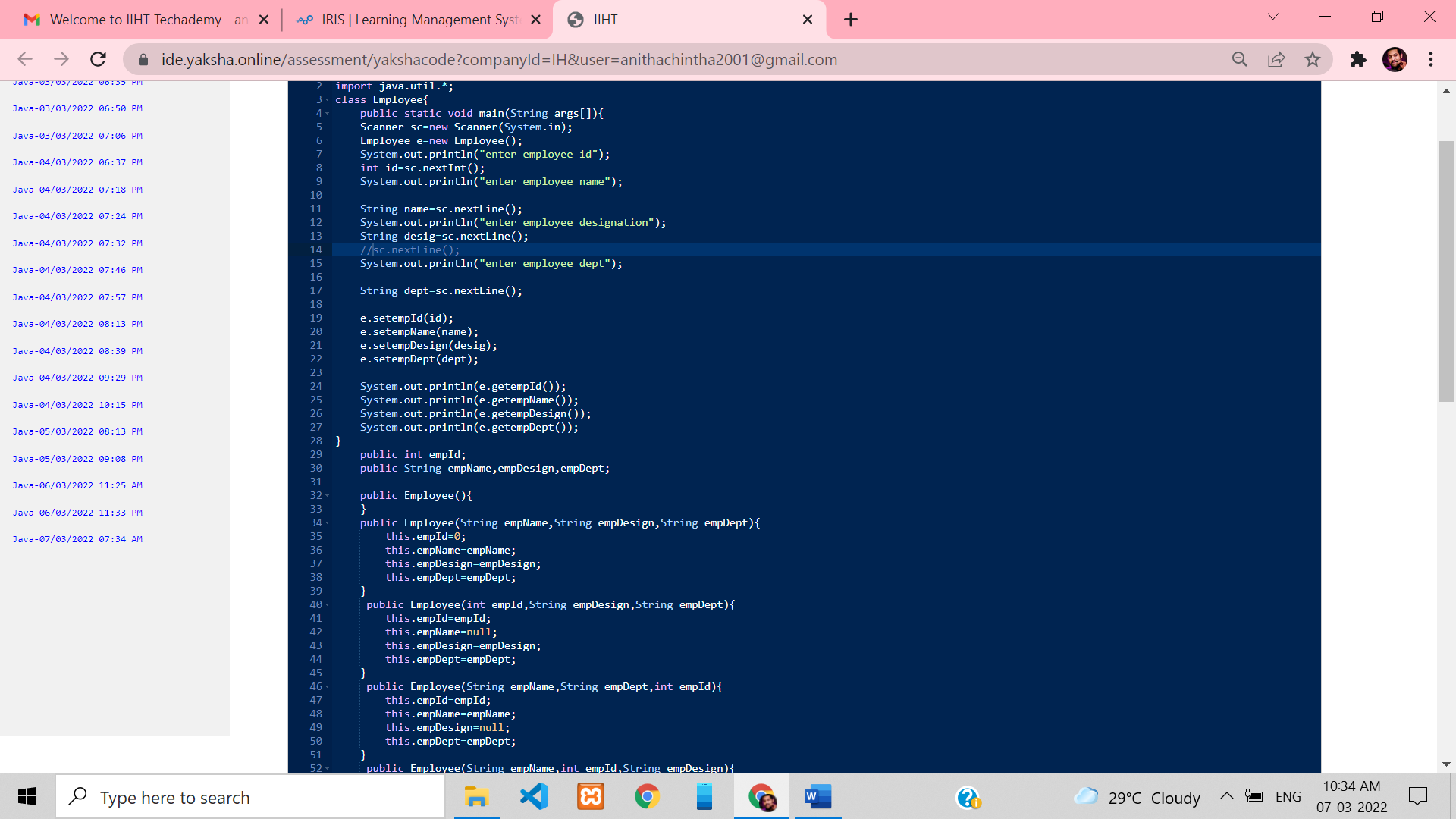
2. **Write a program, which creates an instance of employee class and sets the values for all the attributes.**

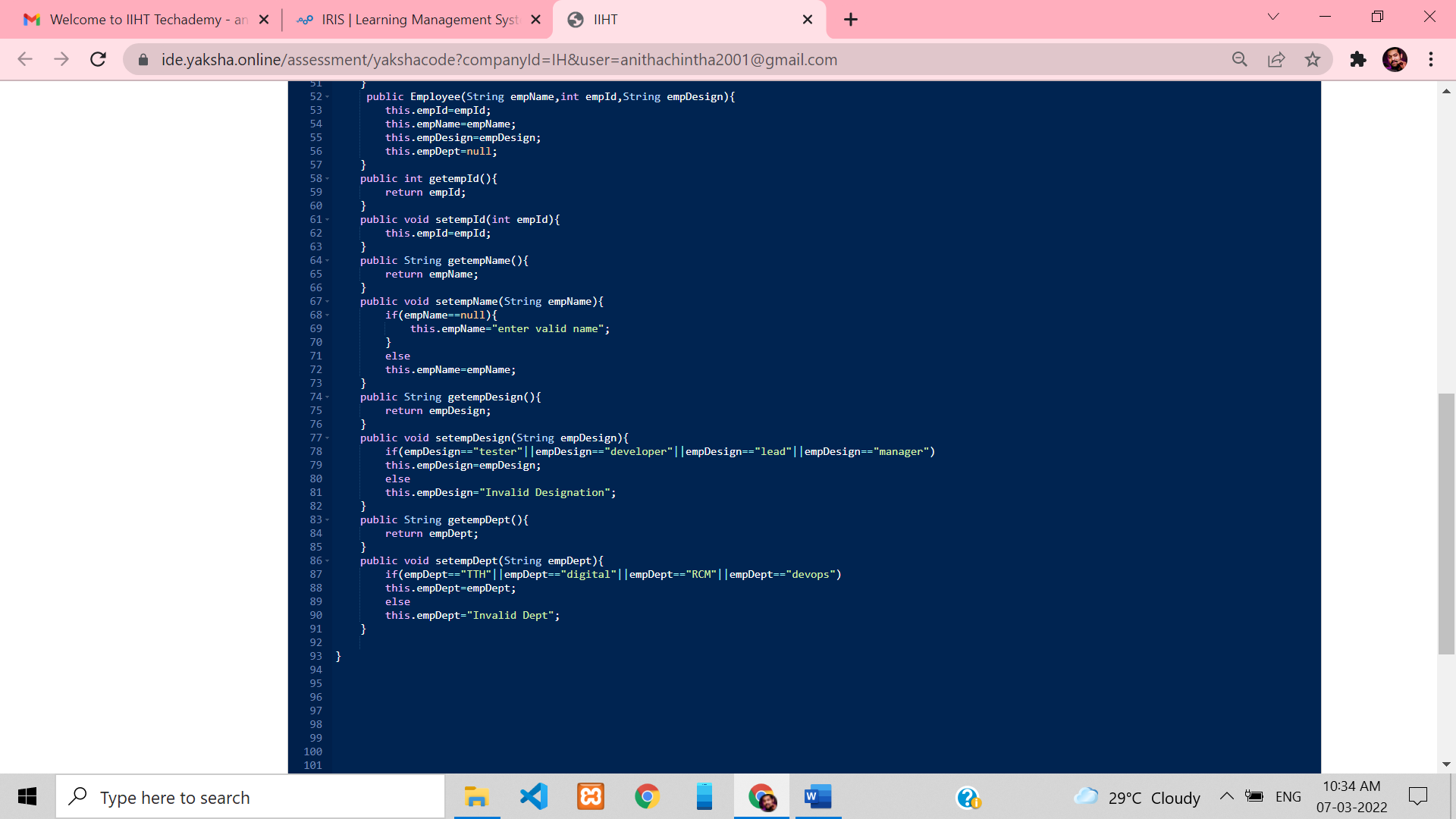
**• While setting value for empName, setEmpName() method should check for NullPointer and display appropriate error message.**

**• While setting value for empDesig, the designation must have any of the following values:**

**developer, tester, Lead or manager. If none of these values is matching then setter method should display 'Invalid designation' error message.**

**• While setting value for empDept, the department must have any of the following values: TTH, RCM, Digital, DevOps. If none of these values is matching then setter method should display 'Invalid Dept' error message.**

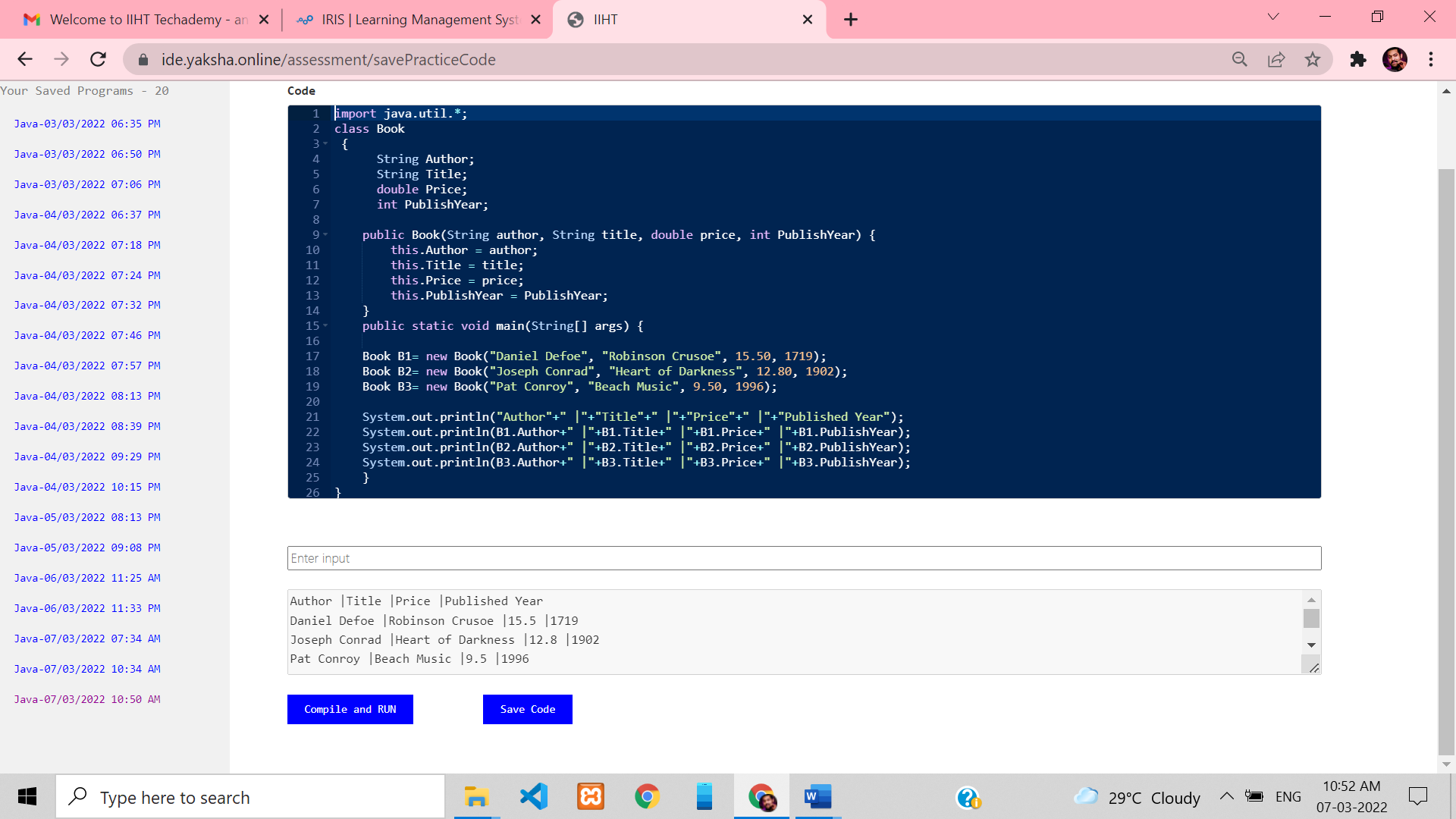
****



**Exercise: Design and implement applications using basic OOP paradigms.**

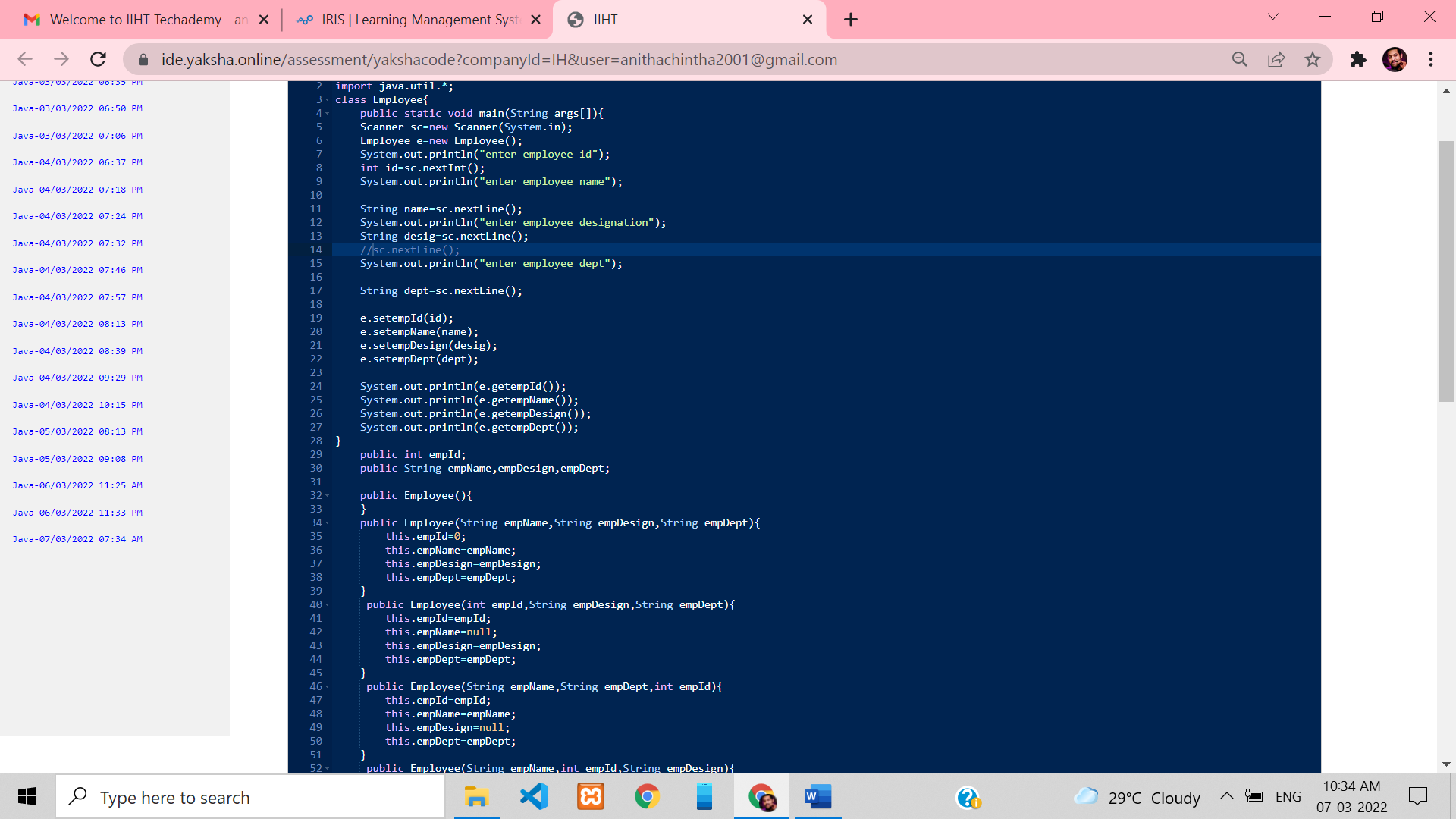
**Develop a program that assists bookstore employees. For each book, the program should track the book’s title, its price, its year of publication, and the author’s name. . . . Develop an appropriate Java Class. Create instances of the class to represent these three books:**

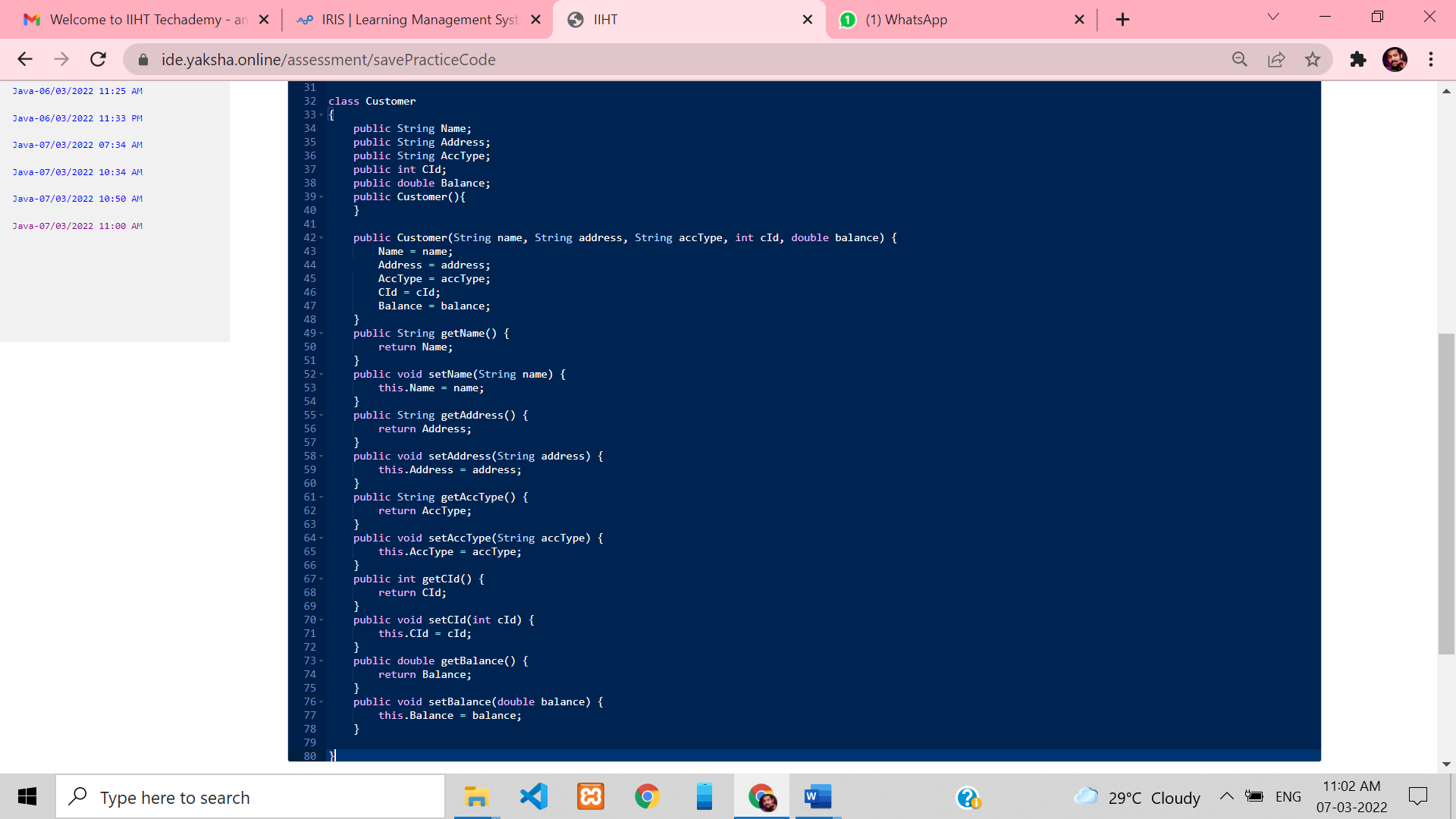
**• Daniel Defoe, Robinson Crusoe, $15.50, 1719; · Joseph Conrad, Heart of Darkness, $12.80, 1902; · Pat Conroy, Beach Music, $9.50, 1996.**

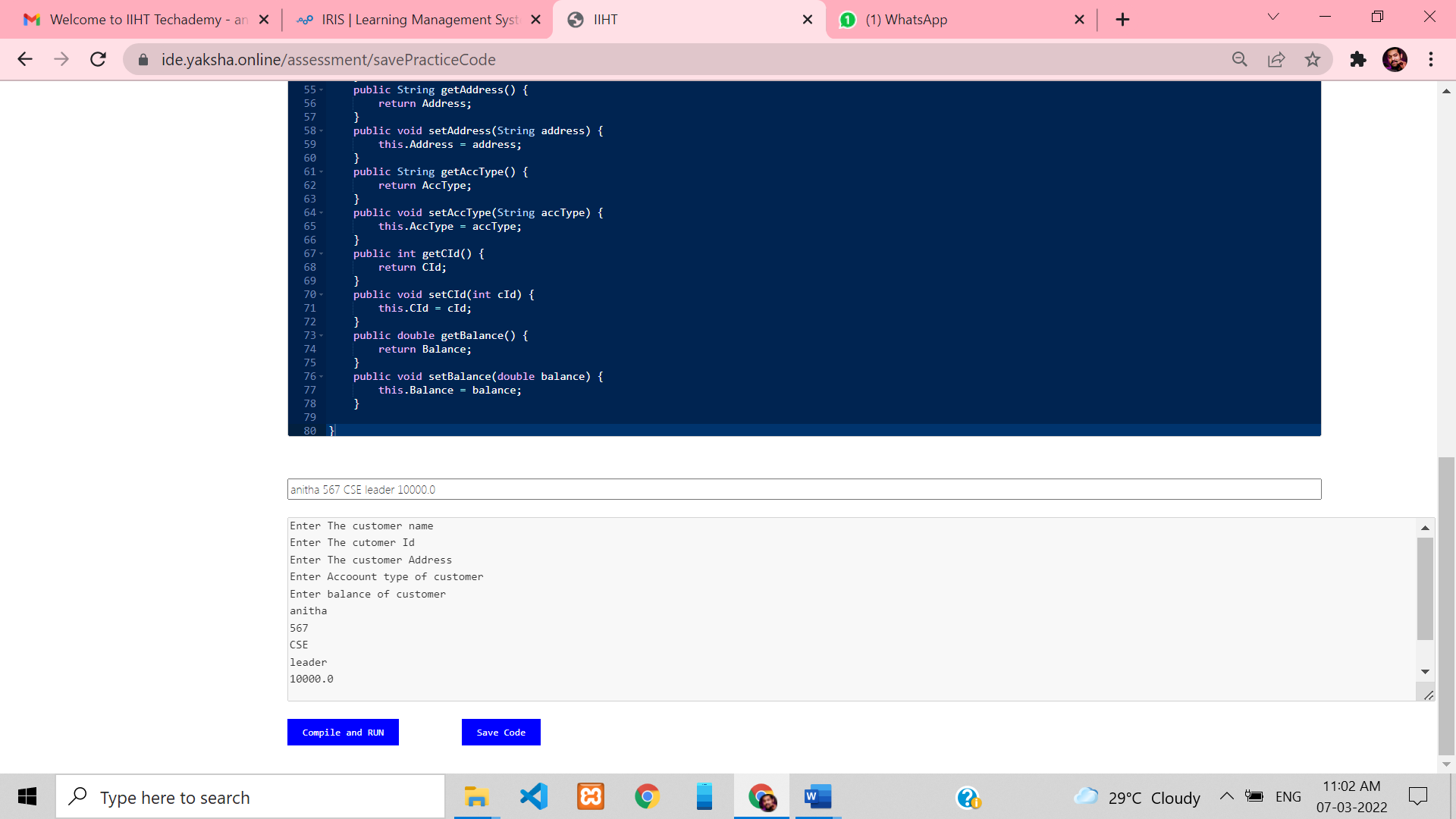


**Exercise: Design and implement applications using basic OOP paradigms.**

**XYZ bank wants to maintain customer details. It will register the customer details whenever a person opens an account with the bank. Below is the customer class diagram:**

****

****

****

**Exercise: Design and implement applications using basic OOP paradigms.**

**Objective:**

**Given a class diagram for a problem, use the method and constructor overloading concepts to solve the problem and test using a set of values in an IDE.**

**Problem Description: The admin of a pre-university college wants to calculate the marks of students in a particular course based on some criteria. Write a Java program to implement the below given class diagram.**

**Student**

**-studentId: int**

**-studentName: String**

**-marks: float**

**-secondChance: boolean**

**+Student(int,String,String)**

**+getStudentId(): int +getStudentName(): String +getMarks(): float +getSecondChance(): Boolean +identifyMarks(float): void +identifyMarks(float, float): void**

**Student Class:**

**Constructor: Initializes studentId, studentName and secondChance.**

**identifyMarks(float) method:**

**This method is used to set the marks of the student if the student has cleared in the first chance itself, i.e. second chance is false. This method accepts the marks scored by the student which must be set in the marks instance variable.**

**identifyMarks (float, float) method:**

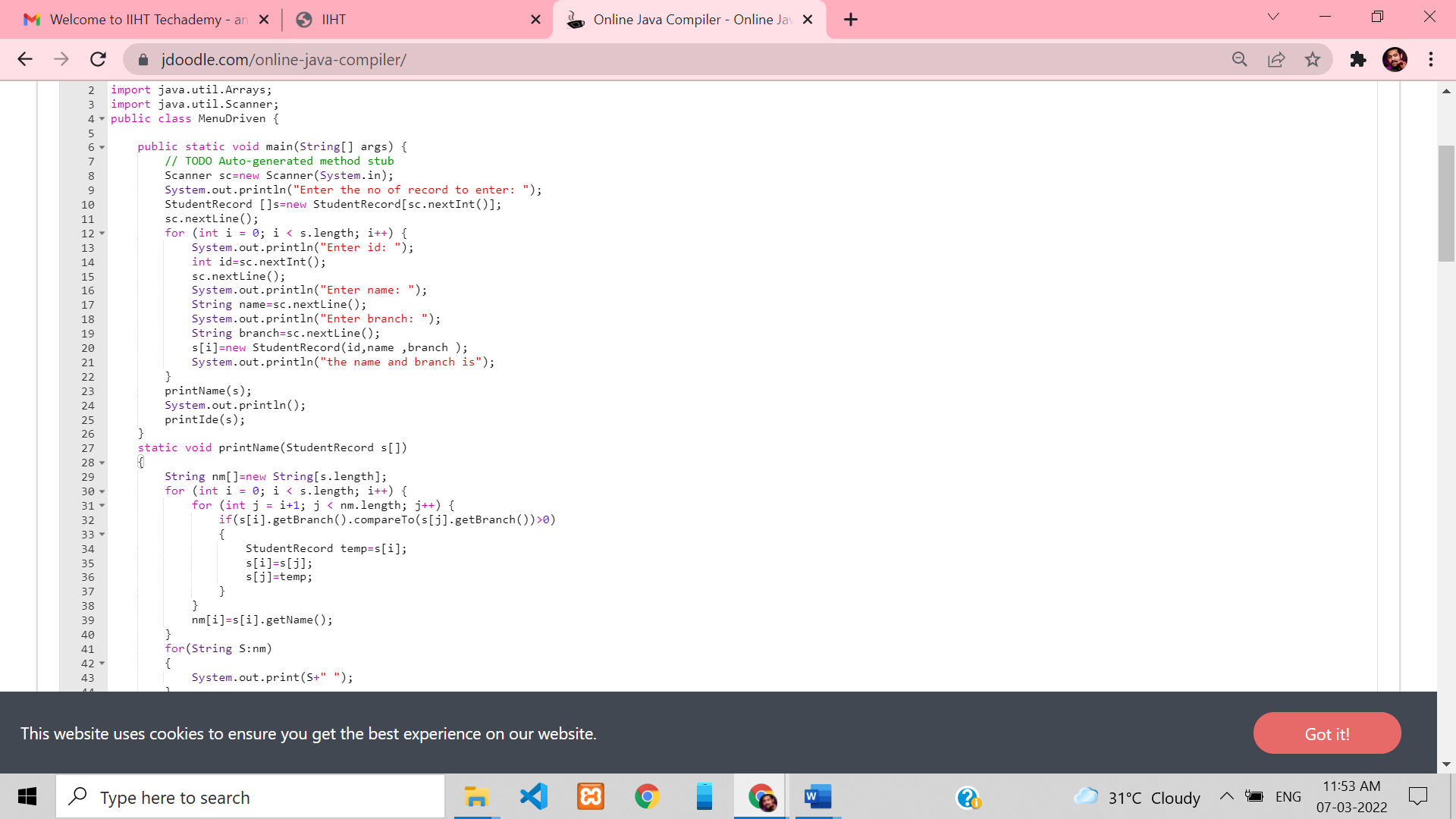
**This method is used to set the marks of the student if the student has taken the second chance i.e. second chance is true. This method accepts the marks scored by the student in the first chance and second chance. The maximum of both these marks must be identified and set in the marks instance variable.**

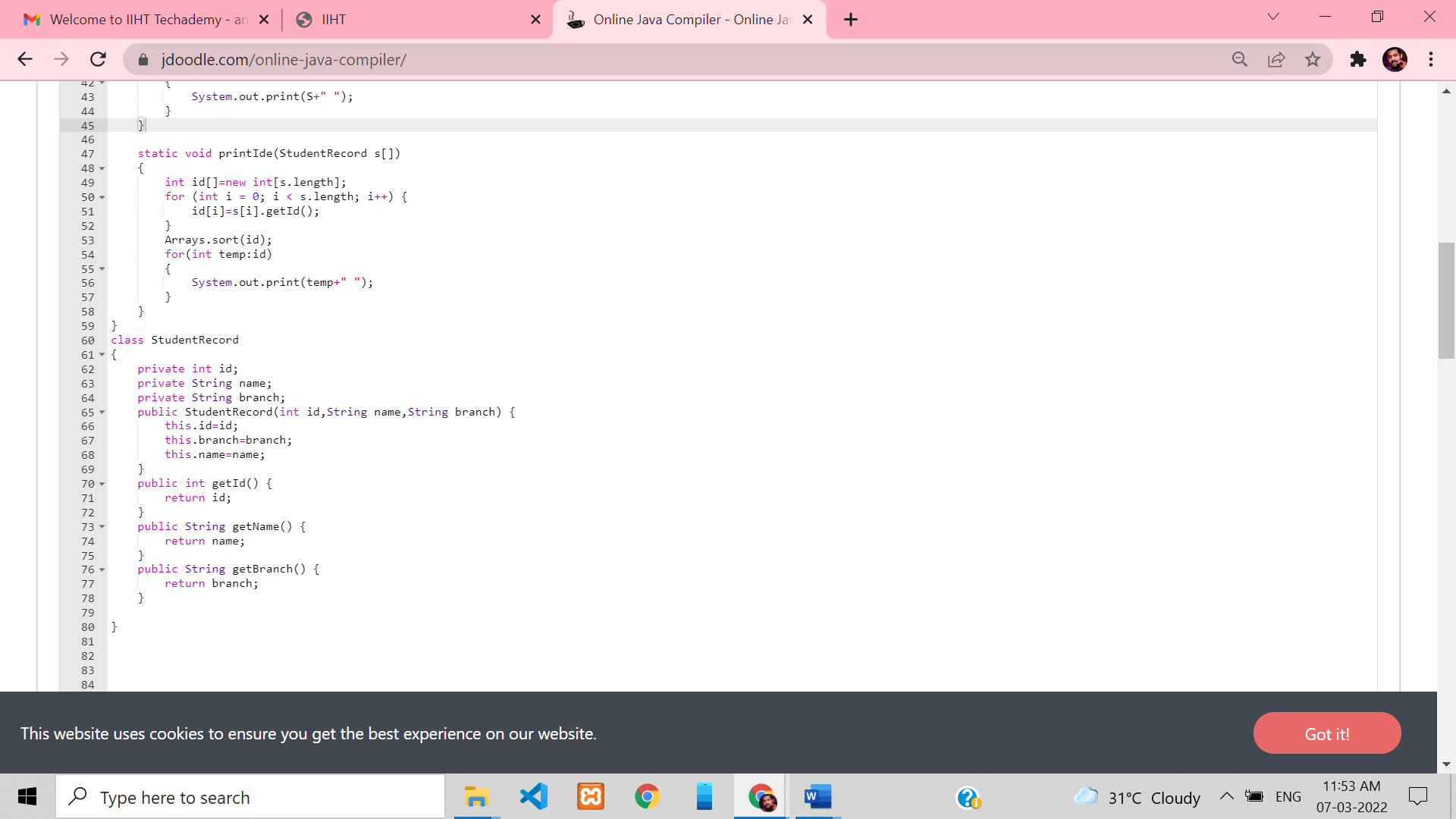
**Starter Class:**

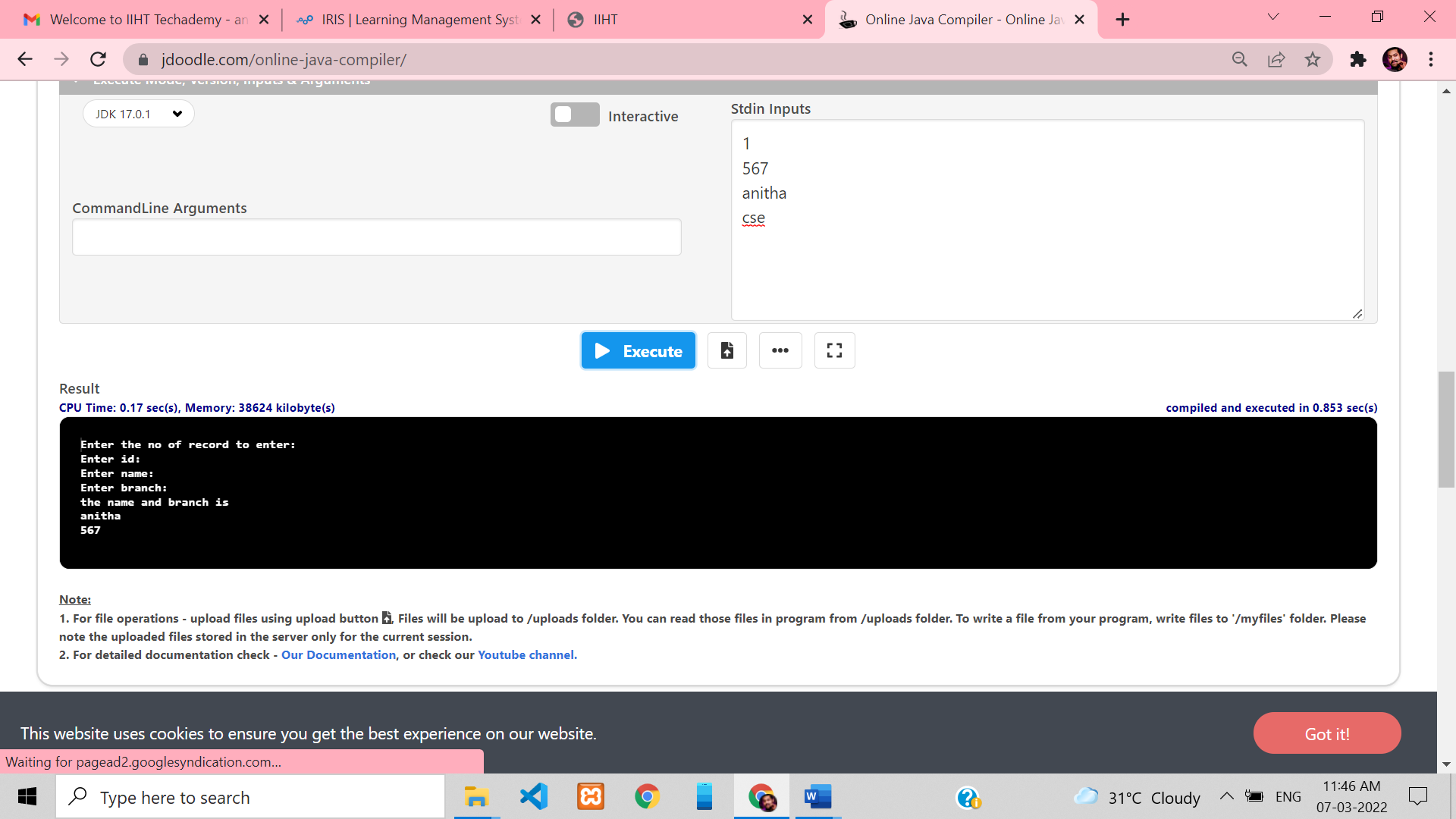
**Write a starter class named Demo,**

**Step1: Create an object of Student class by passing appropriate values to the constructor. Step2: Based on the value used for second chance instance variable, invoke the appropriate identifyMarks() method.**

**Step3: Invoke the getter methods and display all the instance variable values of the Student object created. Create one more object (use different value for second chance) by repeating steps 1 to 3 and test your program**.





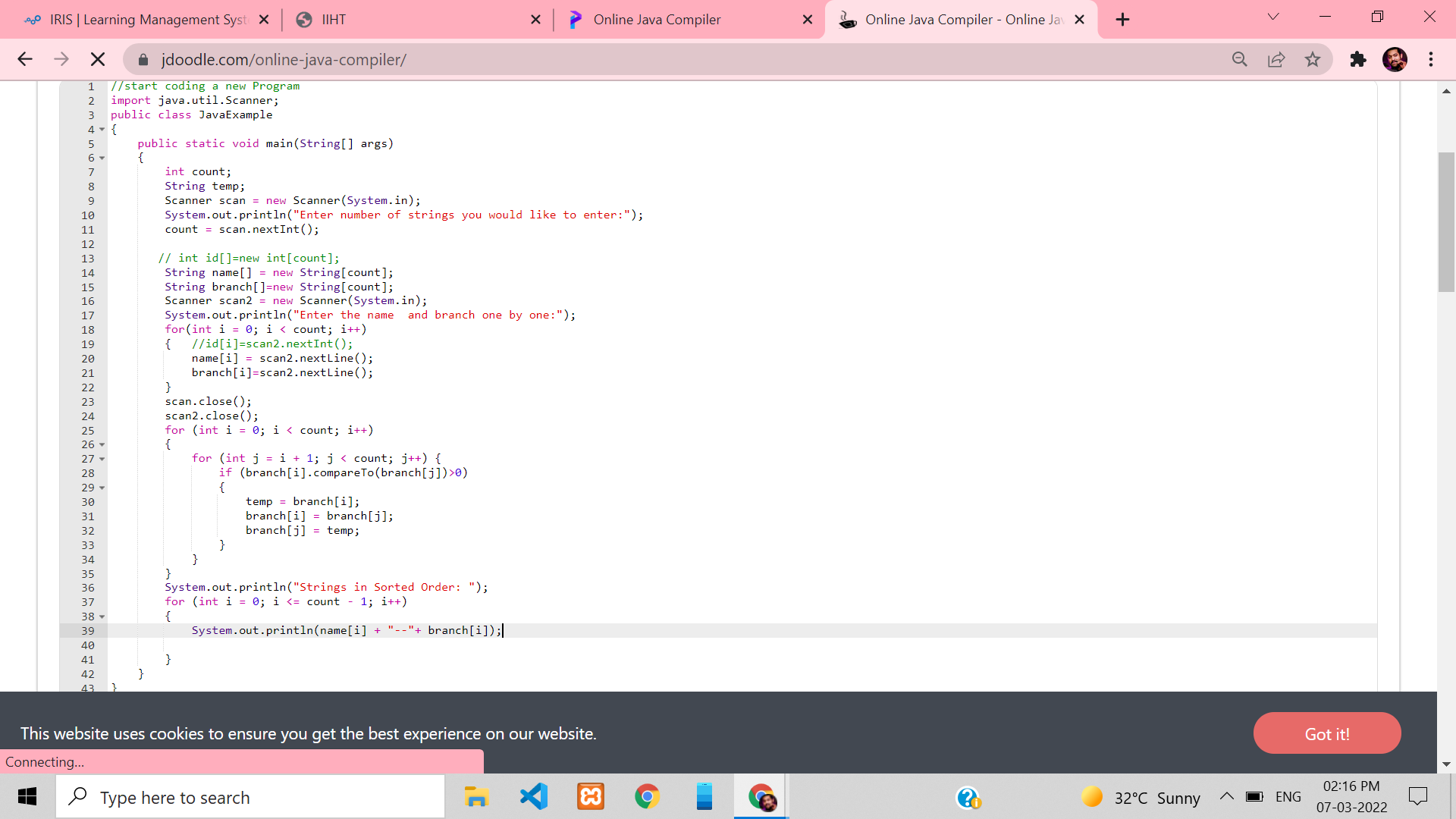


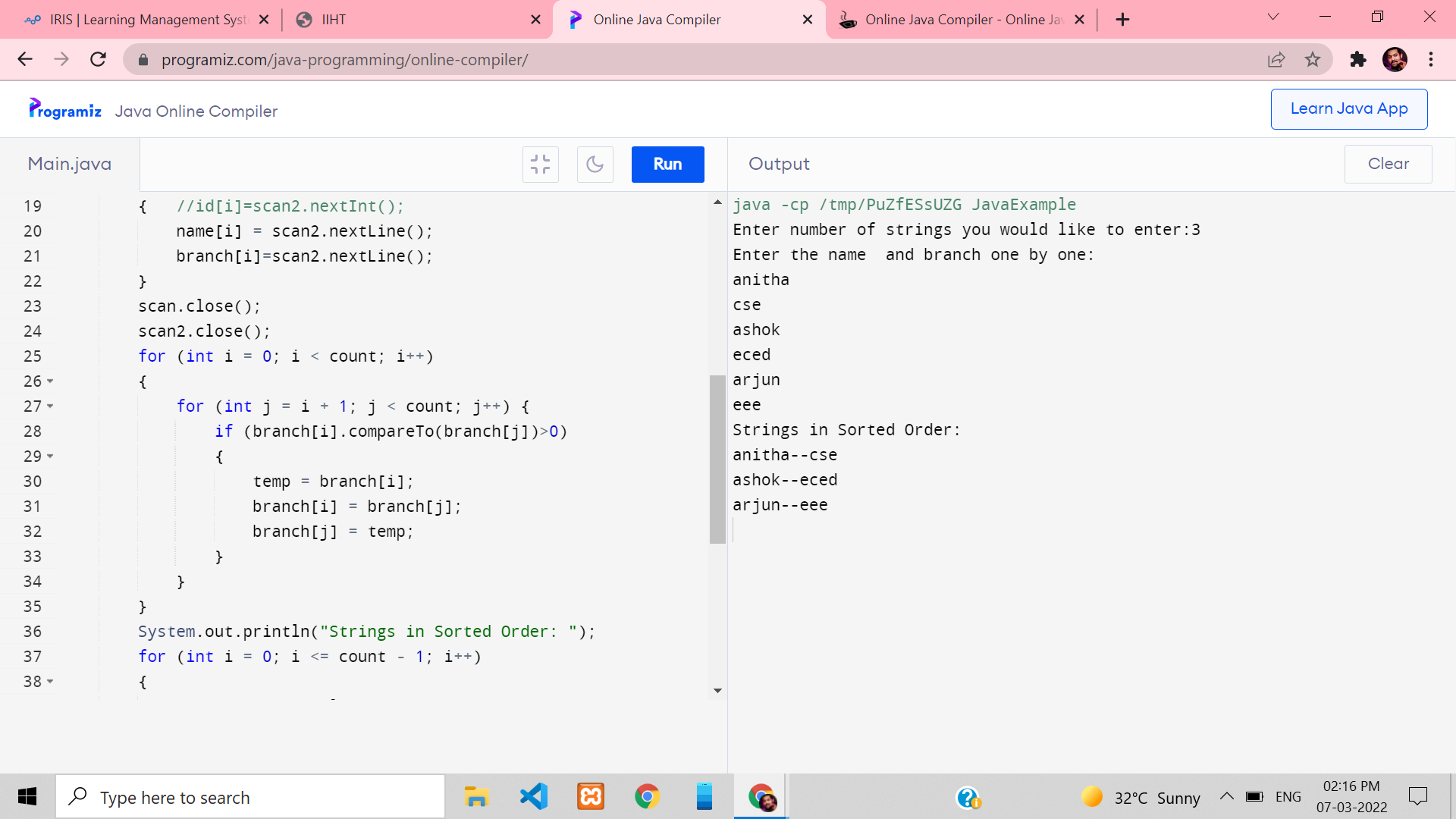
**Exercise 34: Design and implement applications using basic OOP paradigms.**

**Create menu driven program to implement following scenario:**

**1. Create Student Record**

**2. Display Student Names in sorted order based on branch (alphabetical order) 3. Display Student ID in ascending sorted order.**

****

****

**Exercise: Design and implement applications using basic OOP paradigms.**

**Create a method which accepts array of ‘Student’ objects and returns Student object who has scored highest marks.**

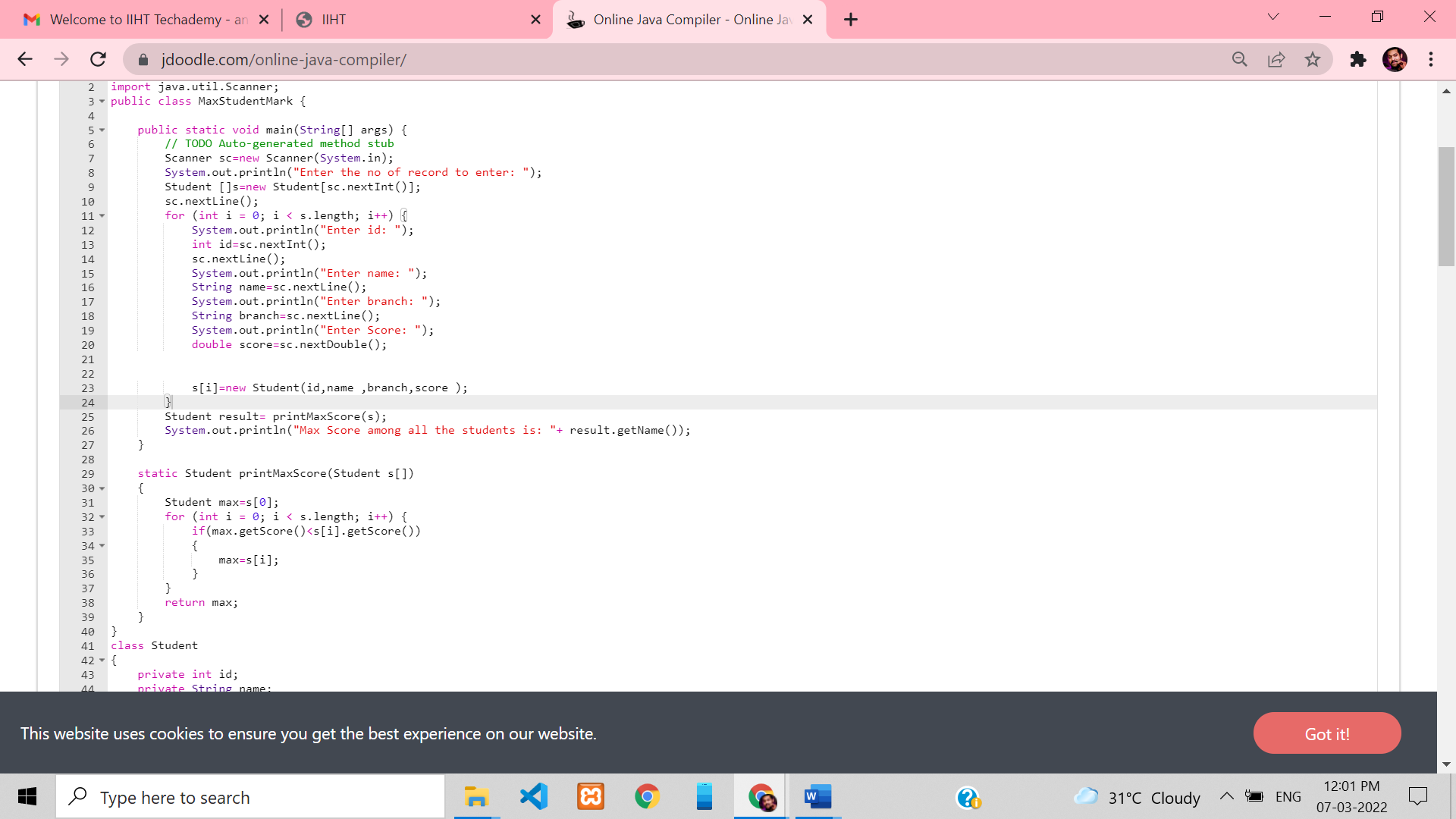
**Note: Each Student object should have following values:**

**• ID**

**• Name**

**• Branch**

**• Sco**

****

