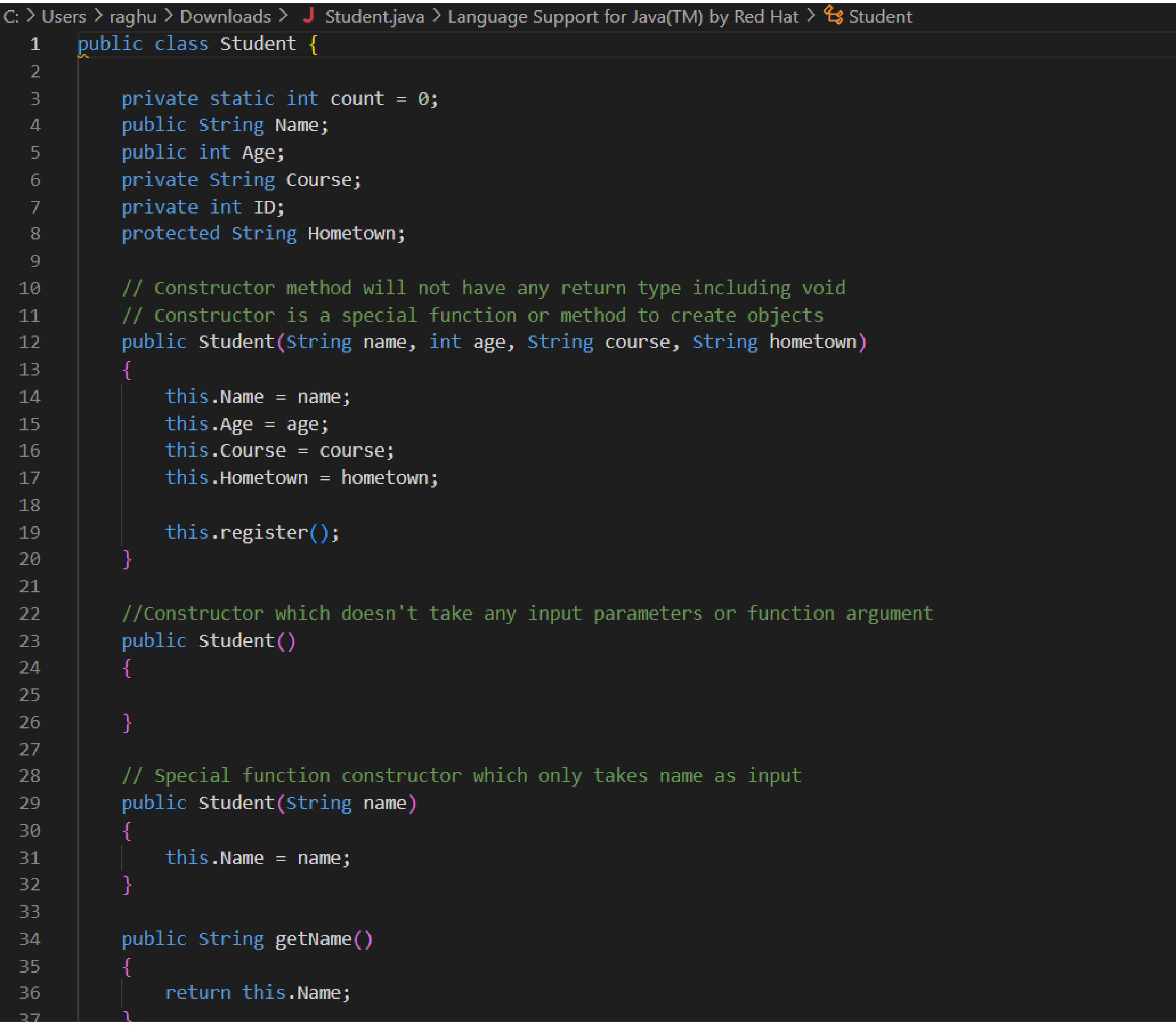
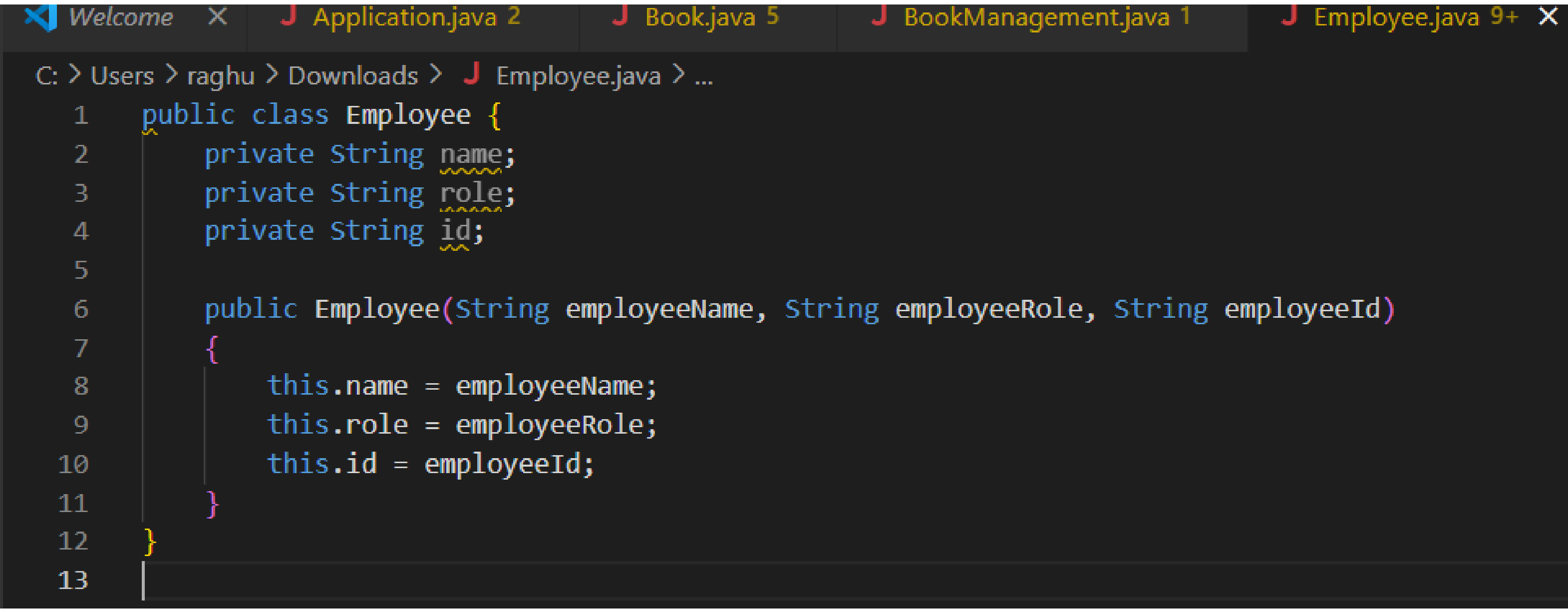
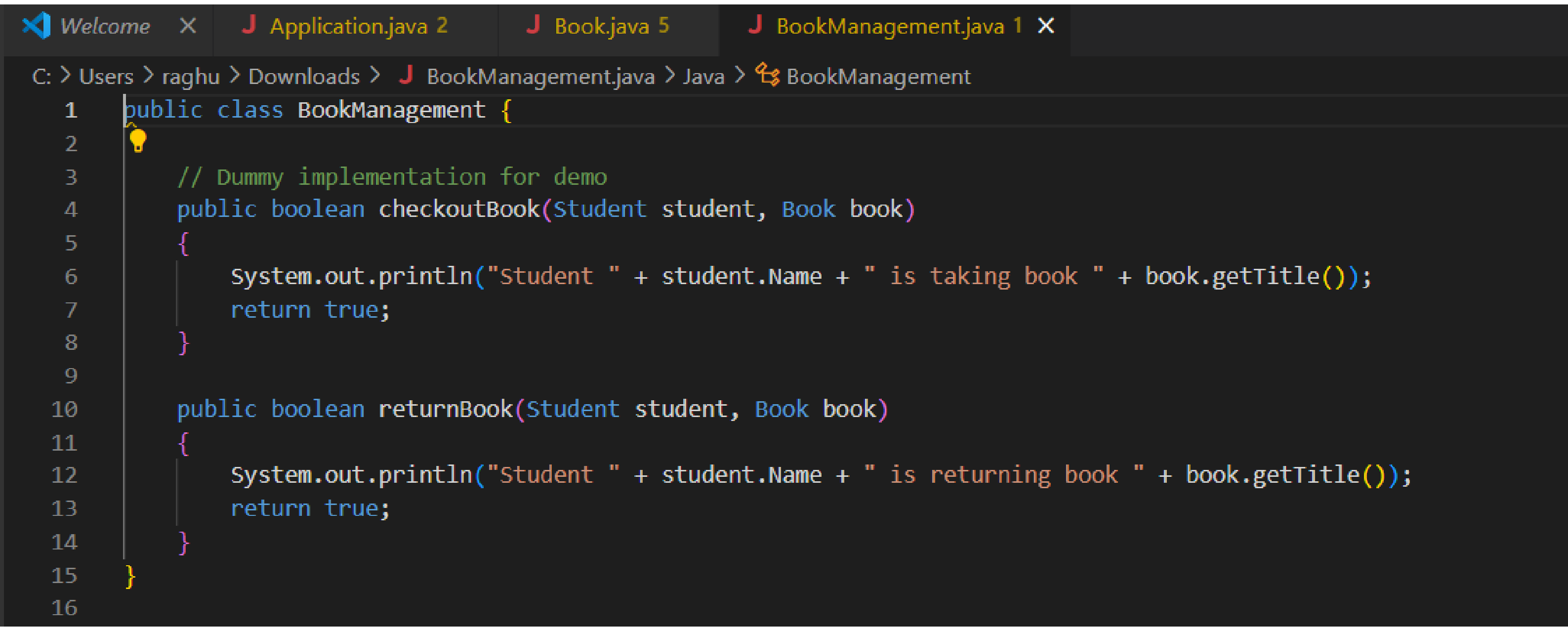

C: > Users > raghu > Downloads > J Student.java > Language Support for Java(TM) by Red Hat > Student

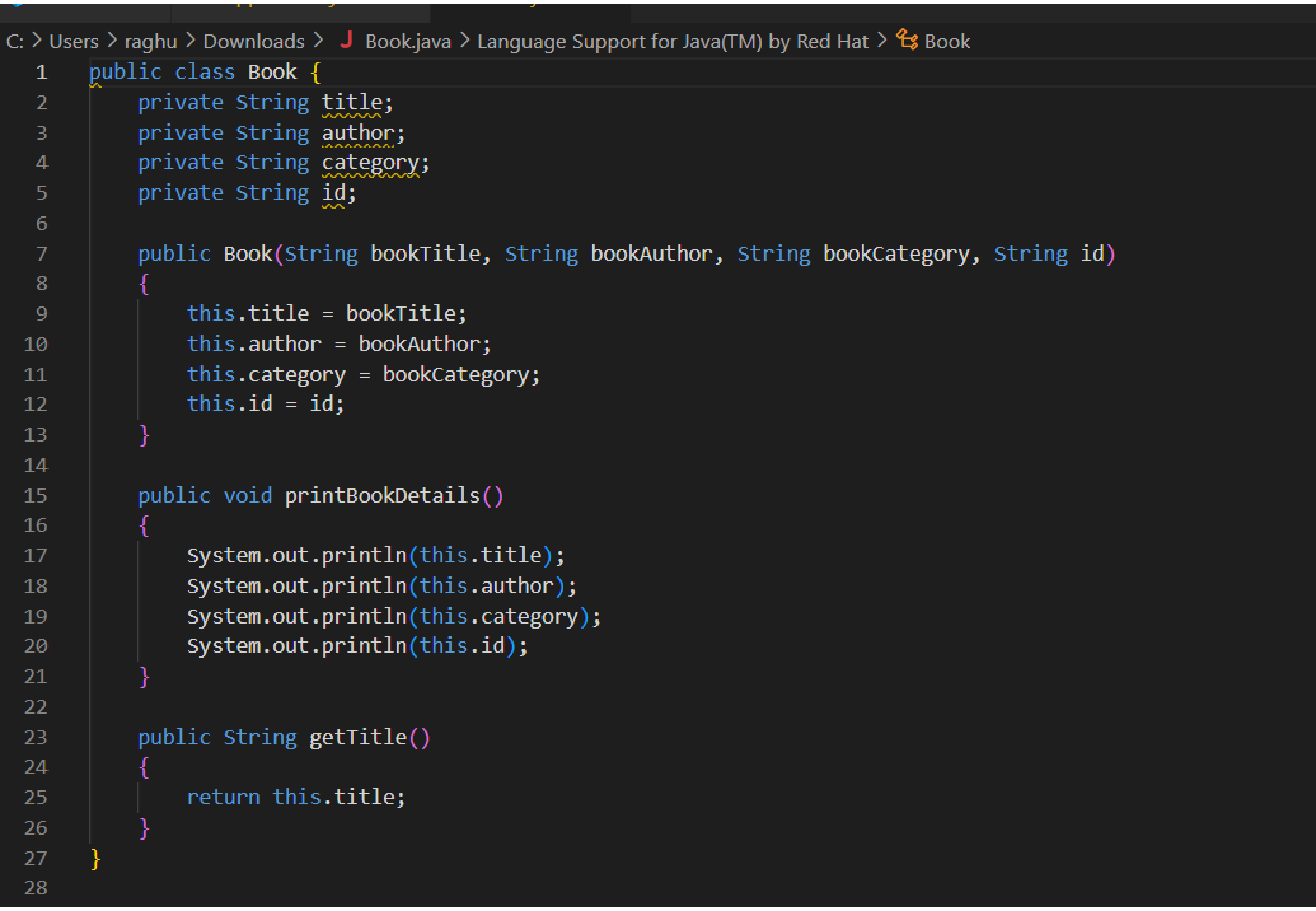
```java
public class Student {

    private static int count = 0;
    public String Name;
    public int Age;
    private String Course;
    private int ID;
    protected String Hometown;

    // Constructor method will not have any return type including void
    // Constructor is a special function or method to create objects
    public Student(String name, int age, String course, String hometown)
    {
        this.Name = name;
        this.Age = age;
        this.Course = course;
        this.Hometown = hometown;

        this.register();
    }

    //Constructor which doesn't take any input parameters or function argument
    public Student()
    {

    }

    // Special function constructor which only takes name as input
    public Student(String name)
    {
        this.Name = name;
    }

    public String getName()
    {
        return this.Name;
    }
```



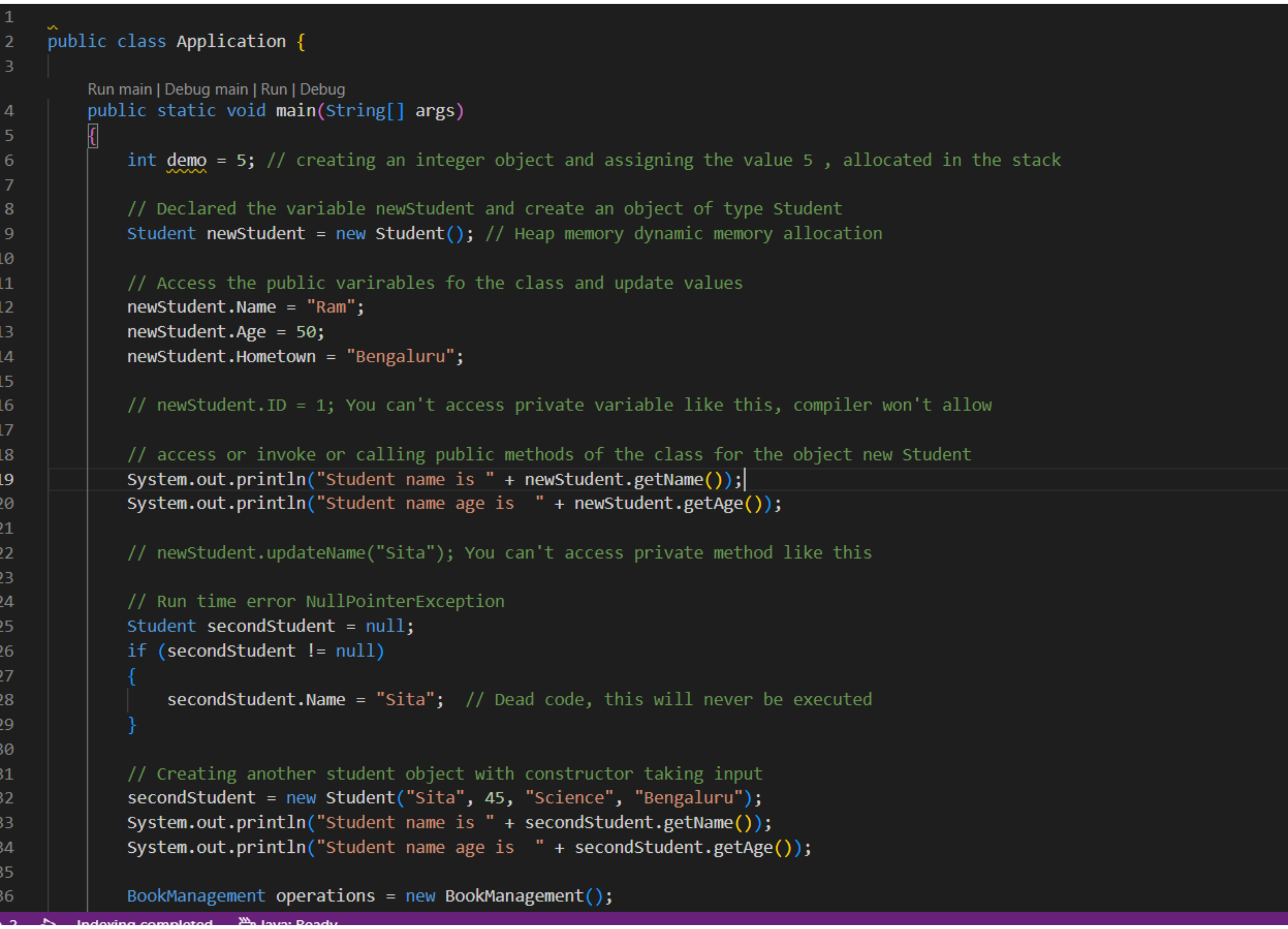
```java
public class Employee {
    private String name;
    private String role;
    private String id;

    public Employee(String employeeName, String employeeRole, String employeeId)
    {
        this.name = employeeName;
        this.role = employeeRole;
        this.id = employeeId;
    }
}

```


Welcome
Application.java 2
Book.java 5
BookManagement.java 1
C: > Users > raghu > Downloads > BookManagement.java > Java > BookManagement

```java
public class BookManagement {

    // Dummy implementation for demo
    public boolean checkoutBook(Student student, Book book)
    {
        System.out.println("Student " + student.Name + " is taking book " + book.getTitle());
        return true;
    }

    public boolean returnBook(Student student, Book book)
    {
        System.out.println("Student " + student.Name + " is returning book " + book.getTitle());
        return true;
    }
}

```


C: > Users > raghu > Downloads > Book.java > Language Support for Java(TM) by Red Hat > Book

```java
public class Book {
    private String title;
    private String author;
    private String category;
    private String id;

    public Book(String bookTitle, String bookAuthor, String bookCategory, String id)
    {
        this.title = bookTitle;
        this.author = bookAuthor;
        this.category = bookCategory;
        this.id = id;
    }

    public void printBookDetails()
    {
        System.out.println(this.title);
        System.out.println(this.author);
        System.out.println(this.category);
        System.out.println(this.id);
    }

    public String getTitle()
    {
        return this.title;
    }
}

```



```java

public class Application {

    public static void main(String[] args)
    {
        int demo = 5; // creating an integer object and assigning the value 5 , allocated in the stack

        // Declared the variable newStudent and create an object of type Student
        Student newStudent = new Student(); // Heap memory dynamic memory allocation

        // Access the public varirables fo the class and update values
        newStudent.Name = "Ram";
        newStudent.Age = 50;
        newStudent.Hometown = "Bengaluru";

        // newStudent.ID = 1; You can't access private variable like this, compiler won't allow

        // access or invoke or calling public methods of the class for the object new Student
        System.out.println("Student name is " + newStudent.getName());
        System.out.println("Student name age is  " + newStudent.getAge());

        // newStudent.updateName("Sita"); You can't access private method like this

        // Run time error NullPointerException
        Student secondStudent = null;
        if (secondStudent != null)
        {
            secondStudent.Name = "Sita";  // Dead code, this will never be executed
        }

        // Creating another student object with constructor taking input
        secondStudent = new Student("Sita", 45, "Science", "Bengaluru");
        System.out.println("Student name is " + secondStudent.getName());
        System.out.println("Student name age is  " + secondStudent.getAge());

        BookManagement operations = new BookManagement();
```