Exam Question: Managing Student Records with Maps and File Operations in Java

Problem Statement:

You are tasked with creating a Java program to manage student records using a `HashMap`. Each student record consists of a unique student ID (String) and the student's name (String). You need to implement CRUD (Create, Read, Update, Delete) operations on this `HashMap`. Additionally, you should provide functionality to save the student records to a file and load them back from the file.

Requirements:

1. **Class Structure**:
   - Create a class named `StudentManager`.
   - This class should contain a private `HashMap<String, String>` to store student records.
2. **Methods to Implement**:

   a. **Add a Student Record**:

```java
public void addStudent(String studentID, String studentName)
```

   - This method should add a new student record to the `HashMap`.
   - If the student ID already exists, the method should print a message indicating that the student ID already exists.

   b. **Retrieve a Student Record**:

```java
public String getStudent(String studentID)
```

   - This method should return the name of the student corresponding to the given student ID.
   - If the student ID does not exist, the method should return `null`.

c. **Update a Student Record**:

```java
public void updateStudent(String studentID, String newStudentName)
```

- This method should update the name of the student with the given student ID.
- If the student ID does not exist, the method should print a message indicating that the student ID does not exist.

d. **Delete a Student Record**:

```java
public void deleteStudent(String studentID)
```

- This method should delete the student record with the given student ID.
- If the student ID does not exist, the method should print a message indicating that the student ID does not exist.

e. **Save Student Records to a File**:

```java
public void saveToFile(String filename) throws IOException
```

- This method should save all student records to a file specified by `filename`.
- Each line in the file should contain a student ID and the corresponding name, separated by a comma.

f. **Load Student Records from a File**:

```java
public void loadFromFile(String filename) throws IOException
```

- This method should load student records from a file specified by `filename`.
- The file format is the same as specified in the saveToFile method.
- If the file does not exist, the method should print a message indicating that the file was not found.

3. **Testing the Class**:

- Write a `main` method to demonstrate the functionality of the `StudentManager` class.
- Add, retrieve, update, and delete a few student records.
- Save the records to a file.
- Clear the `HashMap` and load the records back from the file.
- Print the records to verify that they were correctly saved and loaded.