3. **Key-Value Storage Comparison:**

In Java, dictionaries and maps are used to store collections of key-value pairs. While dictionaries were used historically, maps offer a more modern and flexible approach.

[1] What are the result of each snippet code?

Snippet 1 (Dictionary Class):

```
// (Assuming Hashtable is used internally)
import java.util.Dictionary;
import java.util.Hashtable;

public class StudentRecordsDictionary {

  public static void main(String[] args) {
    Dictionary<Integer, String> studentRecords = new Hashtable<>();
    studentRecords.put(123, "Alice");
    String name = studentRecords.get(123);
    System.out.println("Student with ID 123: " + name);
  }
}
```

Result: .....Student with ID 123: Alice.....

Snippet 2 (Map Interface):

```
import java.util.HashMap;
import java.util.Map;

public class StudentRecordsMap {

  public static void main(String[] args) {
    Map<Integer, String> studentRecords = new HashMap<>();
    studentRecords.put(123, "Alice");
    if (studentRecords.containsKey(789)) {
      System.out.println("Student with ID 789 exists");
    } else {
      System.out.println("Student with ID 789 does not exist");
    }
  }
}
```

Result: .....Student with ID 789 does not exist.....

[2] Based on your analysis, list three key differences between using the Dictionary class and the Map interface for storing student records.

1.Dictionary is an abstract class, while Map is an interface with multiple implementations.

2.Map is more flexible, supports multiple implementations

3.HashMap allows null keys/values, but Dictionary does not allow

[3] Which approach (Dictionary class or Map interface) do you think is more modern and flexible? Why?

Map is the better choice because it provides multiple implementations, better performance, and is the industry standard.