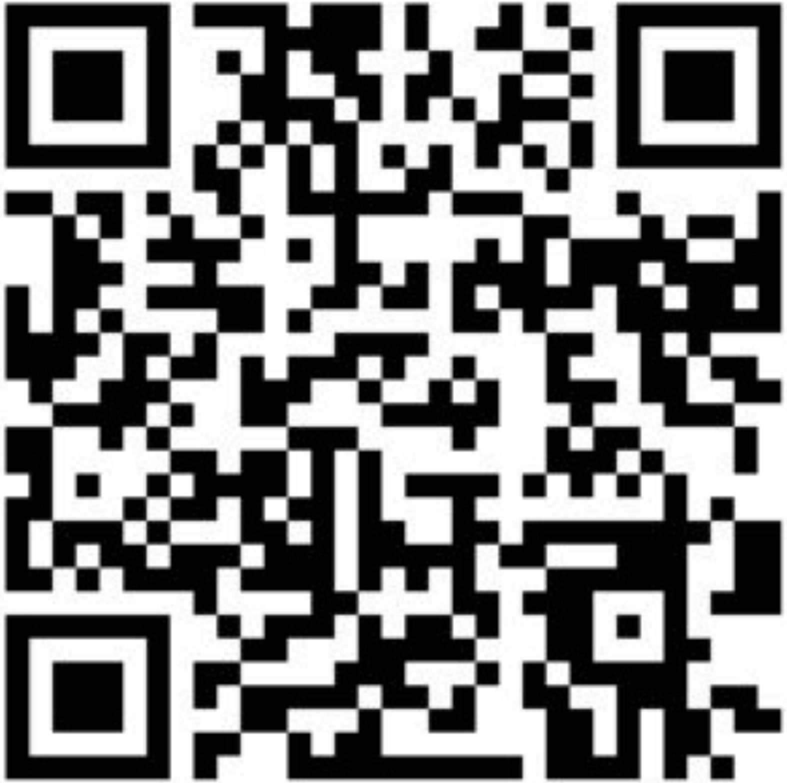
PSLG Week 11

Ben & Amy

ICTLC Online QR Code:



Github

Agenda

- Mappings
- Sets

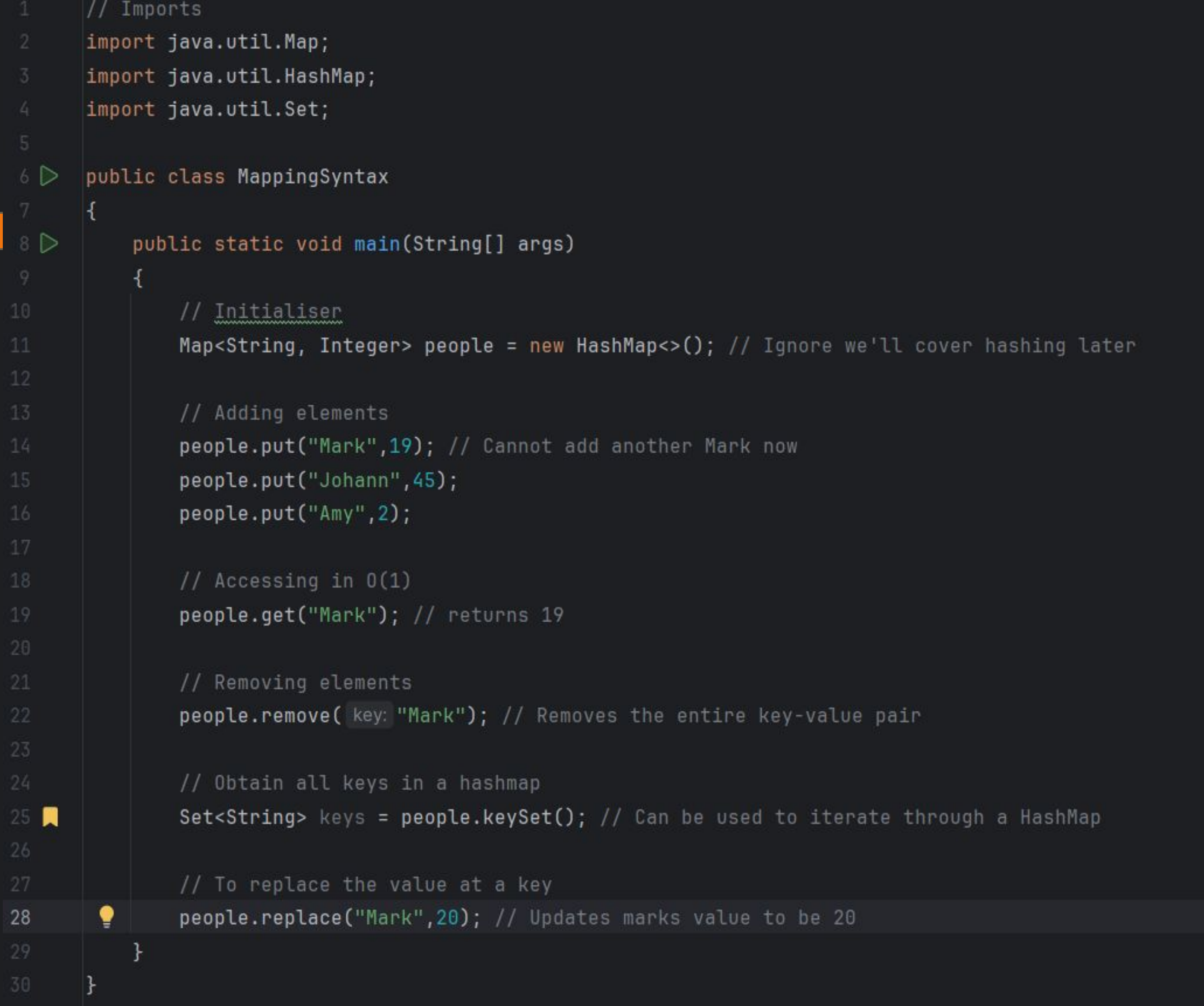
Mappings

A Mapping is a collection of *“Key value pairs”.*
In more simple terms, mappings are relations between two data types.

For example if you wanted to relate some sort of unique String (Like a name) to some distinct integer (Like an age), you can create a mapping between these two values that would look something like this:

“Mark” → 19
“Amy” → 2
“Sean” → 2

Mappings Syntax



```java
// Imports
import java.util.Map;
import java.util.HashMap;
import java.util.Set;

public class MappingSyntax
{
    public static void main(String[] args)
    {
        // Initialiser
        Map<String, Integer> people = new HashMap<>(); // Ignore we'll cover hashing later

        // Adding elements
        people.put("Mark",19); // Cannot add another Mark now
        people.put("Johann",45);
        people.put("Amy",2);

        // Accessing in O(1)
        people.get("Mark"); // returns 19

        // Removing elements
        people.remove( key: "Mark"); // Removes the entire key-value pair

        // Obtain all keys in a hashmap
        Set<String> keys = people.keySet(); // Can be used to iterate through a HashMap

        // To replace the value at a key
        people.replace("Mark",20); // Updates marks value to be 20
    }
}
```

Mapping Problem

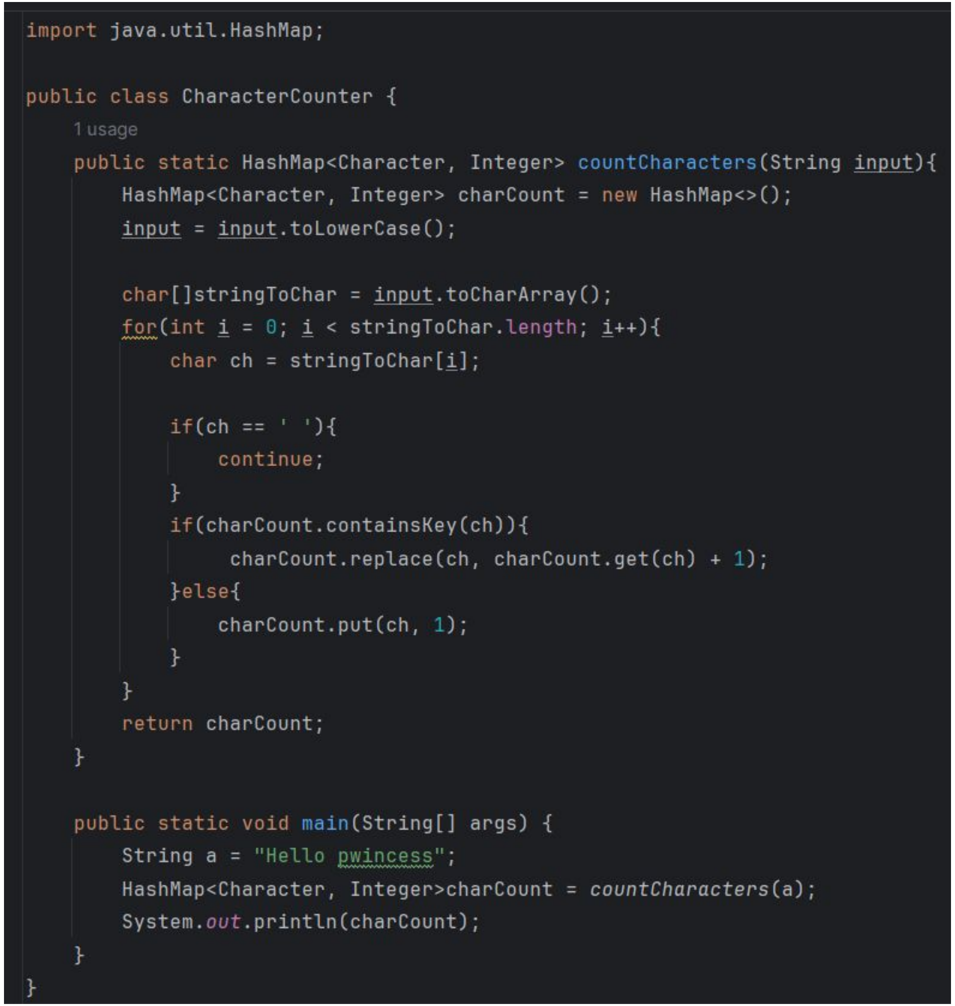
Write a java program that takes a string, and returns the number of occurrences of each character in that string

e.g.

Hello = {h = 1, e = 1, l = 2, o = 1}

Hint : Use hashmap.replace(key, value);

Mappings Solution

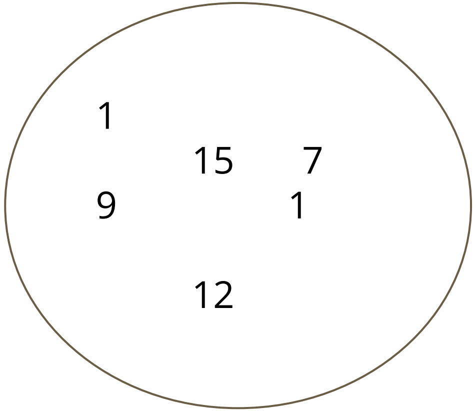


```
import java.util.HashMap;

public class CharacterCounter {
    1 usage
    public static HashMap<Character, Integer> countCharacters(String input){
        HashMap<Character, Integer> charCount = new HashMap<>();
        input = input.toLowerCase();

        char[]stringToChar = input.toCharArray();
        for(int i = 0; i < stringToChar.length; i++){
            char ch = stringToChar[i];

            if(ch == ' '){
                continue;
            }
            if(charCount.containsKey(ch)){
                 charCount.replace(ch, charCount.get(ch) + 1);
            }else{
                charCount.put(ch, 1);
            }
        }
        return charCount;
    }

    public static void main(String[] args) {
        String a = "Hello pwincess";
        HashMap<Character, Integer>charCount = countCharacters(a);
        System.out.println(charCount);
    }
}
```

Sets

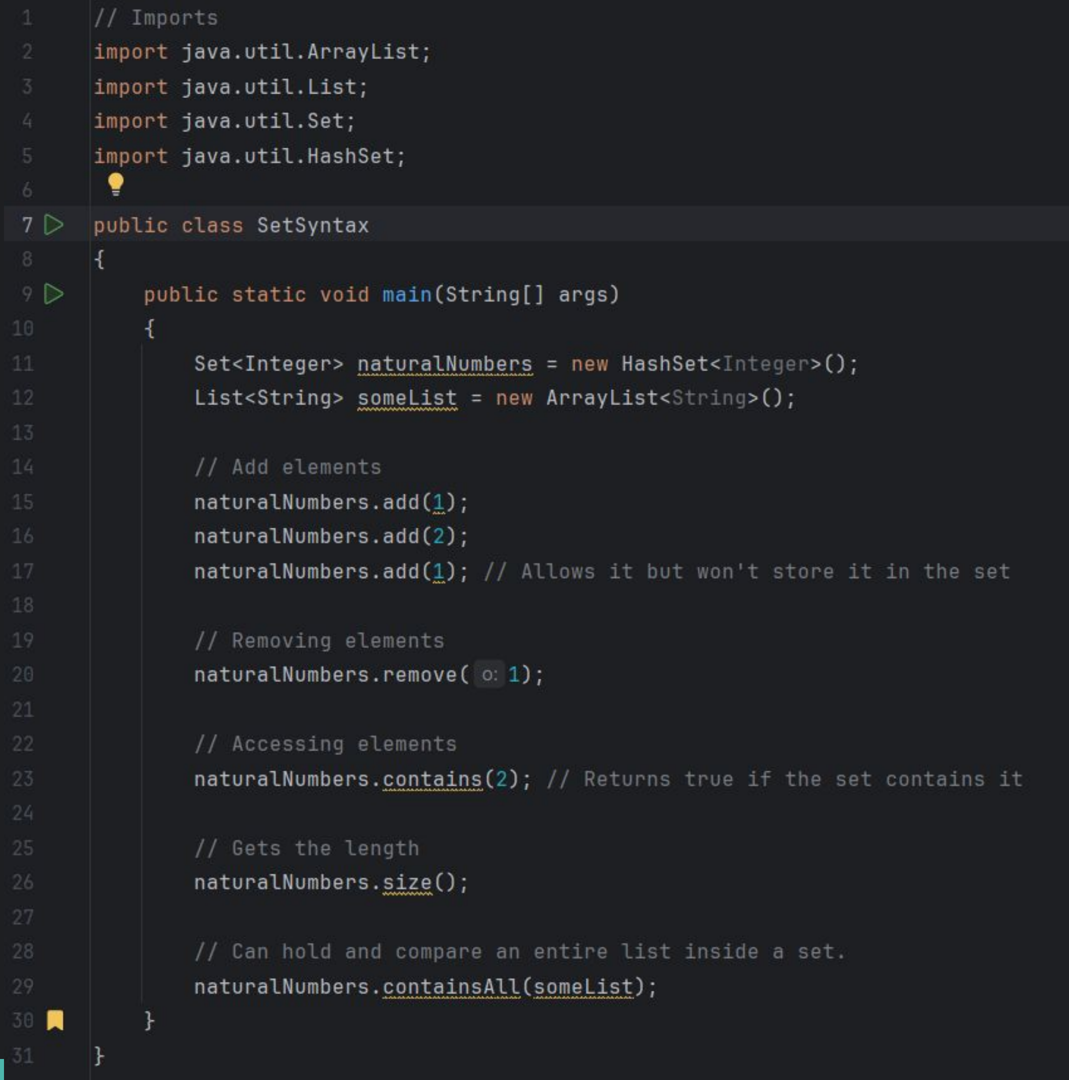
In a generic sense it’s a *collection of distinct elements* .
The difference between distinct and unique in this sense is that you’re allowed to have duplicates but will only be shown the first found instance of the duplicate

It helps to visualise a set in a pure mathematics sense like this:



Note: Sets cannot access the elements that are contained within them but can tell you if they contain them

Set Syntax



```java
// Imports
import java.util.ArrayList;
import java.util.List;
import java.util.Set;
import java.util.HashSet;

public class SetSyntax
{
    public static void main(String[] args)
    {
        Set<Integer> naturalNumbers = new HashSet<Integer>();
        List<String> someList = new ArrayList<String>();

        // Add elements
        naturalNumbers.add(1);
        naturalNumbers.add(2);
        naturalNumbers.add(1); // Allows it but won't store it in the set

        // Removing elements
        naturalNumbers.remove( o: 1);

        // Accessing elements
        naturalNumbers.contains(2); // Returns true if the set contains it

        // Gets the length
        naturalNumbers.size();

        // Can hold and compare an entire list inside a set.
        naturalNumbers.containsAll(someList);
    }
}
```

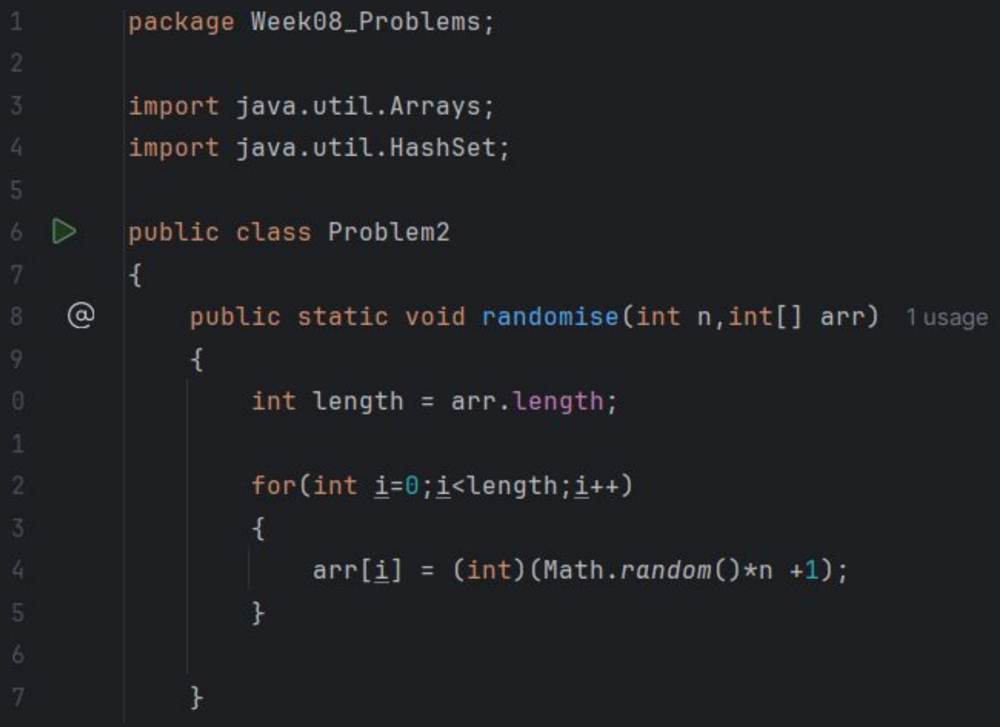
Set Problem

Write a function in Java that takes an array of integers and *returns the* ***first duplicate number*** it encounters. If there are no duplicates, *return -1*.

You must use a Set to track seen elements.

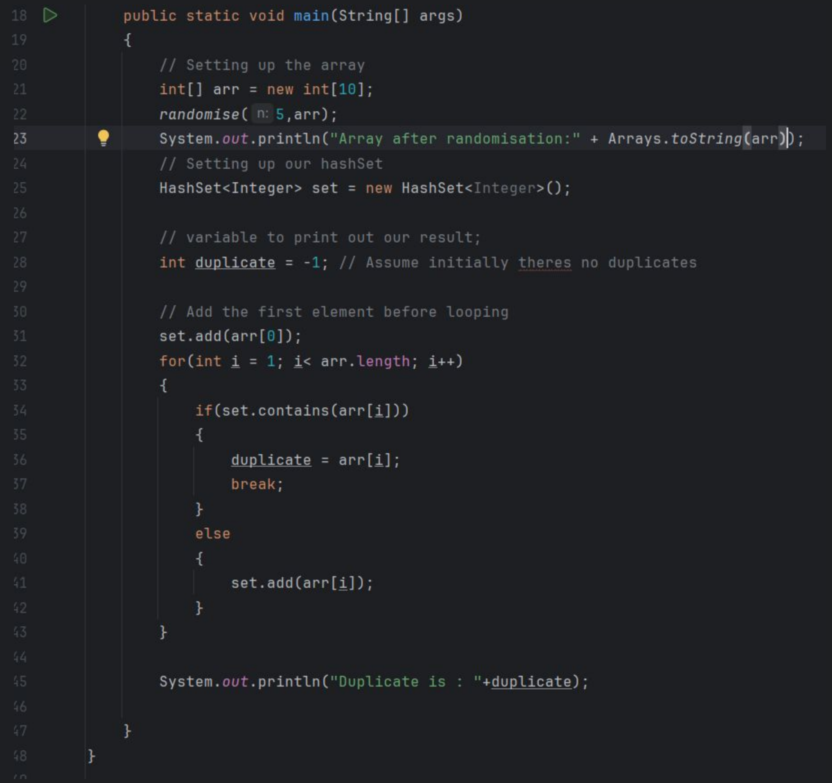
Hashing Solution

Randomise



```
package Week08_Problems;

import java.util.Arrays;
import java.util.HashSet;

public class Problem2
{
    public static void randomise(int n,int[] arr)  1 usage
    {
        int length = arr.length;

        for(int i=0;i<length;i++)
        {
            arr[i] = (int)(Math.random()*n +1);
        }

    }
```

Main Driver



```
    public static void main(String[] args)
    {
        // Setting up the array
        int[] arr = new int[10];
        randomise( n: 5,arr);
        System.out.println("Array after randomisation:" + Arrays.toString(arr));
        // Setting up our hashSet
        HashSet<Integer> set = new HashSet<Integer>();

        // variable to print out our result;
        int duplicate = -1; // Assume initially theres no duplicates

        // Add the first element before looping
        set.add(arr[0]);
        for(int i = 1; i< arr.length; i++)
        {
            if(set.contains(arr[i]))
            {
                duplicate = arr[i];
                break;
            }
            else
            {
                set.add(arr[i]);
            }
        }

        System.out.println("Duplicate is : "+duplicate);

    }
}
```

Final Secret Hidden Challenge (If we have time)

You are given a list of students and their corresponding subjects. Each student can be enrolled in multiple subjects, and each subject can have multiple students. Your task is to write a Java program that does the following:

1. **Store the subjects each student is enrolled in** using a HashMap<String, HashSet<String>>. The key will be the student's name, and the value will be a HashSet of subject names.

2. **Find all students who are enrolled in a given subject** using a HashSet<String>. The input will be the name of a subject, and your program will return the names of all students enrolled in that subject.

Example Input :

Input:

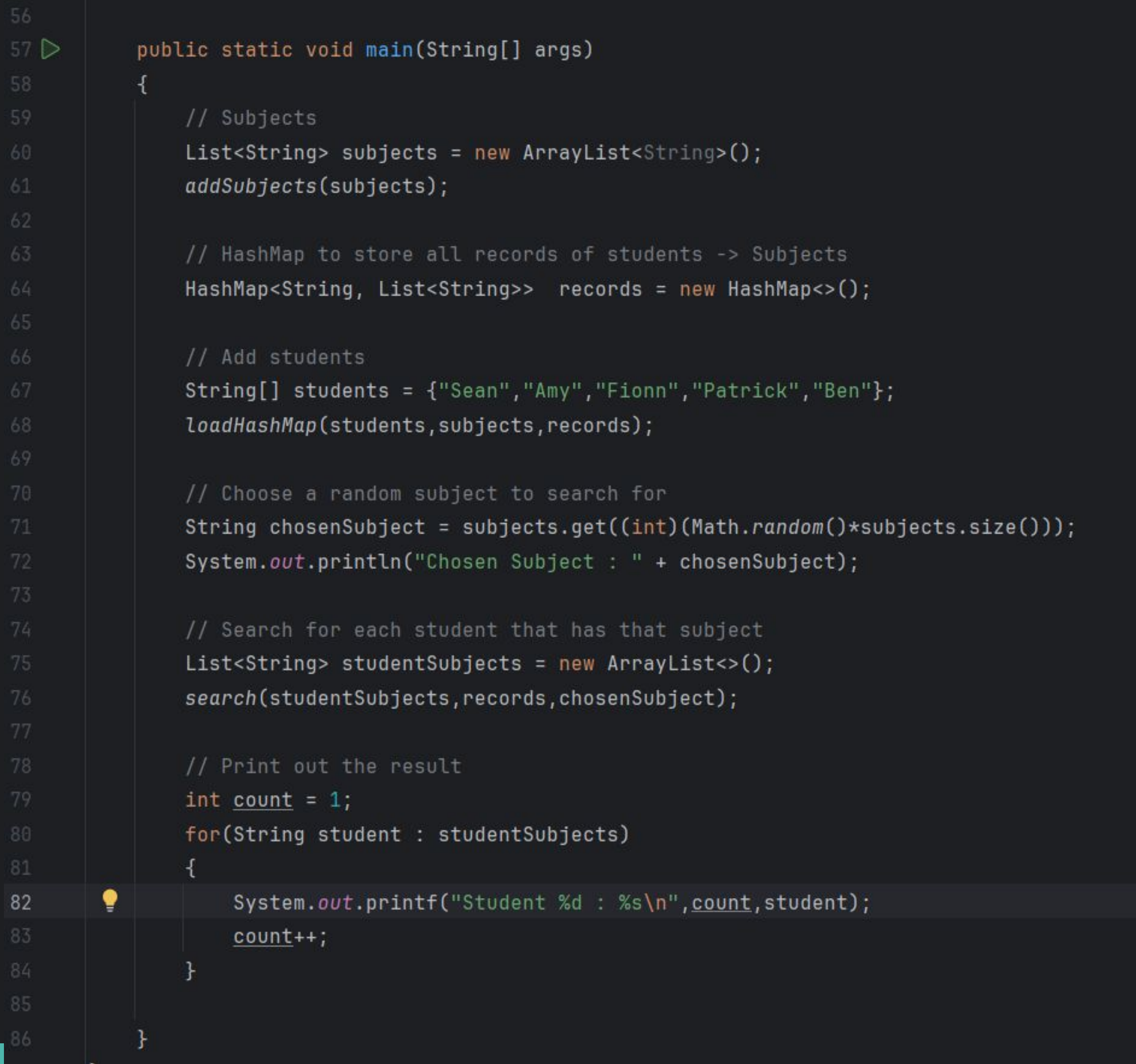
```
{

    "Alice": ["Math", "Science", "History"],

    "Bob": ["Math", "Art"],

    "Charlie": ["History", "Art"],

    "David": ["Science"]

}
```

Find all students enrolled in "Math":

Output:

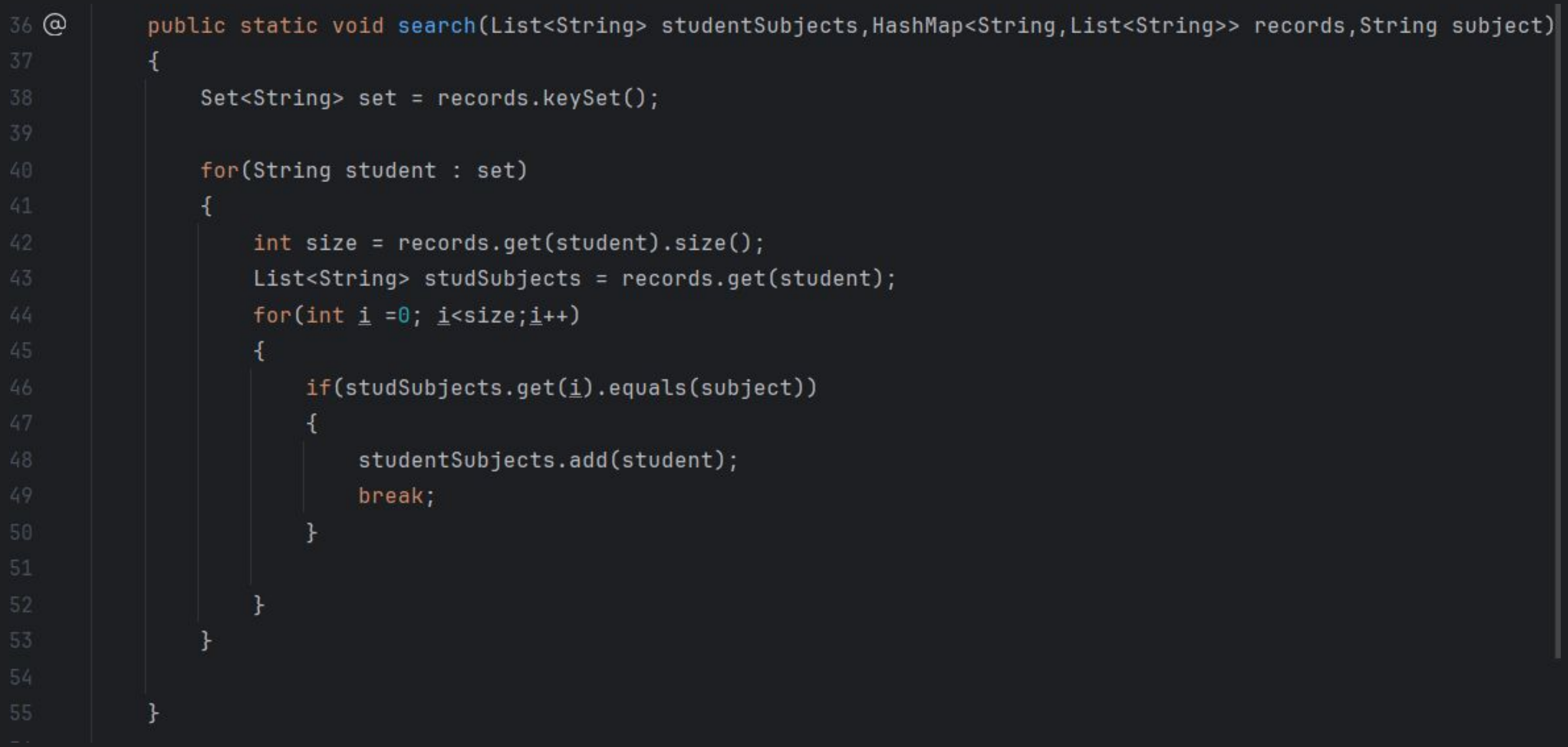
```
["Alice", "Bob"]
```

Final Solution



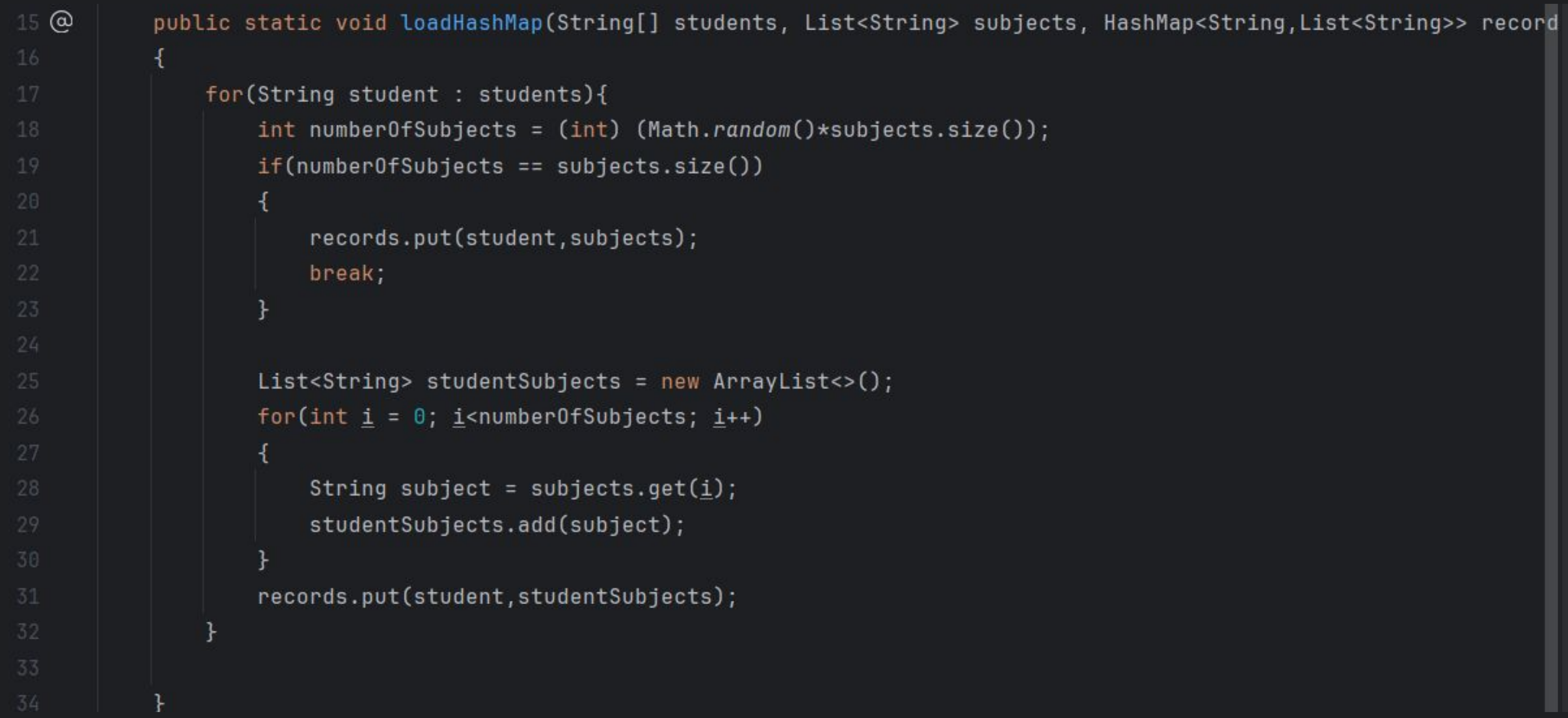
```java
    public static void main(String[] args)
    {
        // Subjects
        List<String> subjects = new ArrayList<String>();
        addSubjects(subjects);

        // HashMap to store all records of students -> Subjects
        HashMap<String, List<String>>  records = new HashMap<>();

        // Add students
        String[] students = {"Sean","Amy","Fionn","Patrick","Ben"};
        loadHashMap(students,subjects,records);

        // Choose a random subject to search for
        String chosenSubject = subjects.get((int)(Math.random()*subjects.size()));
        System.out.println("Chosen Subject : " + chosenSubject);

        // Search for each student that has that subject
        List<String> studentSubjects = new ArrayList<>();
        search(studentSubjects,records,chosenSubject);

        // Print out the result
        int count = 1;
        for(String student : studentSubjects)
        {
            System.out.printf("Student %d : %s\n",count,student);
            count++;
        }

    }
```

Final Solution



```
    public static void search(List<String> studentSubjects,HashMap<String,List<String>> records,String subject)
    {
        Set<String> set = records.keySet();

        for(String student : set)
        {
            int size = records.get(student).size();
            List<String> studSubjects = records.get(student);
            for(int i =0; i<size;i++)
            {
                if(studSubjects.get(i).equals(subject))
                {
                    studentSubjects.add(student);
                    break;
                }

            }
        }

    }
```

Final Solution



```
    public static void loadHashMap(String[] students, List<String> subjects, HashMap<String,List<String>> record
    {
        for(String student : students){
            int numberOfSubjects = (int) (Math.random()*subjects.size());
            if(numberOfSubjects == subjects.size())
            {
                records.put(student,subjects);
                break;
            }

            List<String> studentSubjects = new ArrayList<>();
            for(int i = 0; i<numberOfSubjects; i++)
            {
                String subject = subjects.get(i);
                studentSubjects.add(subject);
            }
            records.put(student,studentSubjects);
        }

    }
```