```java
import java.util.HashSet;
import java.util.Set;

public class Part1 {
   public static  Set<String> intersection(Set<String> s1, Set<String> s2)
   {
      Set<String> s3 = new HashSet<String>();
      if (s1.contains("bb") && s2.contains("bb"))
         s3.add("bb");
      if (s1.contains("cc")&& s2.contains("cc"))
         s3.add("cc");
      return s3;
   }
   public static void main(String[] args)
   {
      Set<String> s1 = new HashSet<String>();
      Set<String> s2 = new HashSet<String>();
      s1.add("aa");
      s1.add("bb");
      s1.add("cc");
      s2.add("dd");
      s2.add("cc");
      s2.add("bb");
      Set<String> s3 = new HashSet<String>();
      s3 = intersection(s1,s2);
      System.out.println(s3);
   }


}
```


```java
import java.util.*;
public class Employee implements Comparable<Employee>
{
   String name;
   double salary;
   public Employee(String name, double s)
   {
      this.name = name;
```

```
        salary = s;
    }
    public String toString()
    {
        return ( this.name + "," + this.salary);
    }

    @Override
    public int compareTo(Employee o) {
        if(this.salary == o.salary){
            return 0;
        }
        else if(this.salary > o.salary){
            return 1;
        }
        return -1;
    }


}//end clas
```

```
import java.util.*;


public class Part2 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Set<Employee> EmployeeSet = new TreeSet<Employee>();
        //Set<Employee> EmployeeSet = new TreeSet<Employee>(new SalaryComparator());
        Employee e1 = new Employee ("eee",78000.0);
        Employee e2 = new Employee ("bbb",45000.0);
        Employee e3 = new Employee ("ccc",100000.0);
        EmployeeSet.add(e1);
        EmployeeSet.add(e2);
        EmployeeSet.add(e3);
        Iterator get = EmployeeSet.iterator();
        while (get.hasNext())
        {
            System.out.println(get.next().toString());
        }



    }
```

```
}
```

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_231.jdk/Contents/Home/bin/java ...
bbb,45000.0
eee,78000.0
ccc,100000.0

Process finished with exit code 0
```

```
import java.util.*;
public class Part3
{
  public static void main(String [] args)
  {
    String text = "Good morning.Have a good class." + "Have a good visit.Have fun!";
    //Create a TreeMap to hold words as key and count as value
    //YOUR CODE
    TreeMap<String, Integer> StringMap = new TreeMap<String,Integer>();
    //Extracting a word from a text by using the split method in the String
    //class. The words will be a, class, fun, good, Good, Have, morning and visit.
    String [] words = text.split("[ \n\t\r.,;:!?(){}]");
    //For each word extracted in the array words, WRITE CODE to check whether it is already in
    //the stored as a key in the map (use the methods containsKey). If not, a new pair consisting of
    //the word and its initial count is stored in the map.
    //Otherwise, the count for the word is incremented by 1.
    //YOUR CODE
    for(int i =0; i< words.length;i++)
    {
      words[i] = words[i].toLowerCase();
      System.out.println(words[i]);
    }
    int count = 0;
    for (String s : words)
    {

      if (StringMap.containsKey(s))
      {
        StringMap.put(s, StringMap.get(s)+1);
      }
      else
      {
        StringMap.put(s, 1);
      }
    }
    //Write code to obtain entries of the map in a set, and traverse the set to display the key
    //and the count in each entry.entries into a set
    //YOUR CODE

    System.out.println();
```

```
        for (String word: StringMap.keySet())
        {
            String key = word.toString();
            String value = StringMap.get(word).toString();
            System.out.println(key + " " + value);
        }
    }//end main
}//end class
```

```
good
morning
have
a
good
class
have
a
good
visit
have
```

```
have
fun

a 2
class 1
fun 1
good 3
have 3
morning 1
visit 1

Process finished with exit code 0
```

```
import java.util.Comparator;

public class SalaryComparator implements Comparator<Employee> {

    // This compares employees based on salaries
    public int compare(Employee o1, Employee o2) {
        if (o1.salary >= o2.salary) {
            return 1;
        } else {
            return -1;
        }
    }



}
```