

CSE220 – Sessional Task

Week 3

You'll have to implement a `StringSet` class, which is basically a set of Strings. That means it will contain strings, but no two strings will be the same.

1. Use the following skeleton.

```
class StringSet {  
  
    //declare variables here  
    StringSet(int size) {} //Constructor  
  
    int insert(String s) {} //return 1 for success, -1 if Set is full, -2 if already exists  
  
    String[] intersect(StringSet ss) {} //return the strings that are present in both this and  
    ss StringSet  
  
    boolean contains(String s) {} //check if s exists in this  
  
    public String toString()  
    {  
        //return "Set elements: " + <elements of this set> + "\n"  
    }  
}
```

2. Use the following code to test your solution.

<pre>public class LabWork { public static void main(String [] args) { StringSet s1 = new StringSet(3); StringSet s2 = new StringSet(3); String temp1[] = {"abc", "def", "abc", "sunday", "monday"}; String temp2[] = {"wednesday", "sunday", "wednesday", "abc", "monday"}; for (String t: temp1) { int x = s1.insert(t); if (x == -1) System.out.println("Set is full, can't insert " + t); if (x == -2) System.out.println(t + " already exists"); } System.out.println(s1); for (String t: temp2) { int x = s2.insert(t); if (x == -1) System.out.println("Set is full, can't insert " + t); if (x == -2) System.out.println(t + " already exists"); } System.out.println(s2); System.out.print("Common elements: "); for (String t: s1.intersect(s2)) System.out.print(t + " "); System.out.print("\n"); } }</pre>	<p><u>Output:</u> abc already exists Set is full, can't insert monday Set elements: abc def sunday wednesday already exists Set is full, can't insert monday Set elements: wednesday sunday abc Common elements: abc sunday</p>
--	---