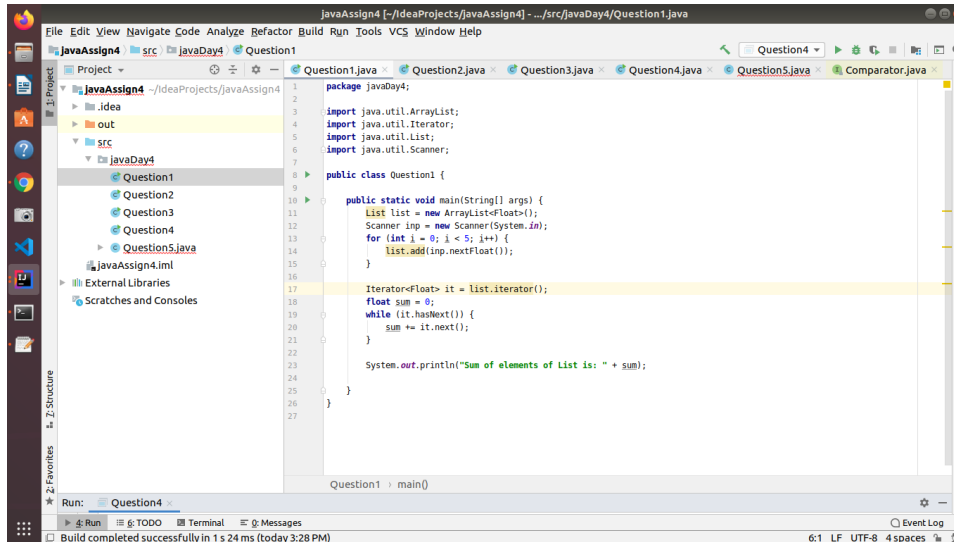
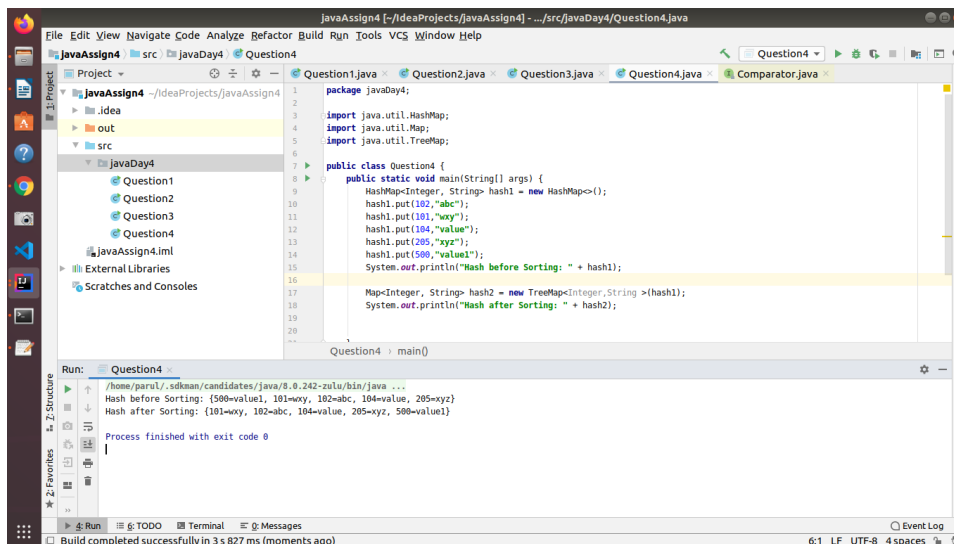


Question: 1:

Write Java code to define List . Insert 5 floating point numbers in List, and using an iterator, find the sum of the numbers in List.



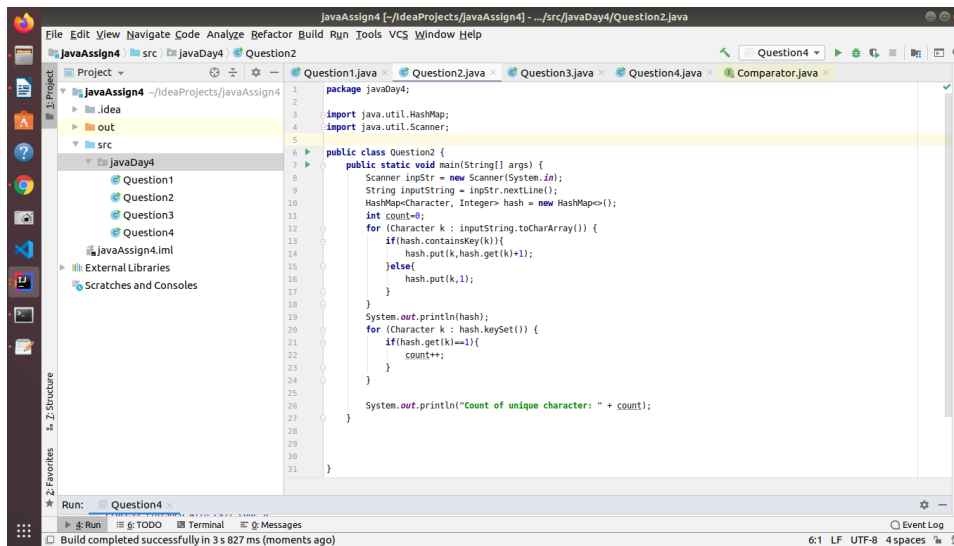
```
1 package javaDay4;
2
3 import java.util.ArrayList;
4 import java.util.Iterator;
5 import java.util.List;
6 import java.util.Scanner;
7
8 public class Question1 {
9
10     public static void main(String[] args) {
11         List list = new ArrayList<Float>();
12         Scanner inp = new Scanner(System.in);
13         for (int i = 0; i < 5; i++) {
14             list.add(inp.nextFloat());
15         }
16
17         Iterator<Float> it = list.iterator();
18         float sum = 0;
19         while (it.hasNext()) {
20             sum += it.next();
21         }
22
23         System.out.println("Sum of elements of List is: " + sum);
24     }
25 }
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```



```
1 package javaDay4;
2
3 import java.util.HashMap;
4 import java.util.Map;
5 import java.util.TreeMap;
6
7 public class Question4 {
8
9     public static void main(String[] args) {
10         HashMap<Integer, String> hash1 = new HashMap<>();
11         hash1.put(102, "abc");
12         hash1.put(101, "xyz");
13         hash1.put(104, "value");
14         hash1.put(205, "xyz");
15         hash1.put(500, "value1");
16         System.out.println("Hash before Sorting: " + hash1);
17
18         Map<Integer, String> hash2 = new TreeMap<Integer, String>(hash1);
19         System.out.println("Hash after Sorting: " + hash2);
20     }
21 }
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Question: 2:

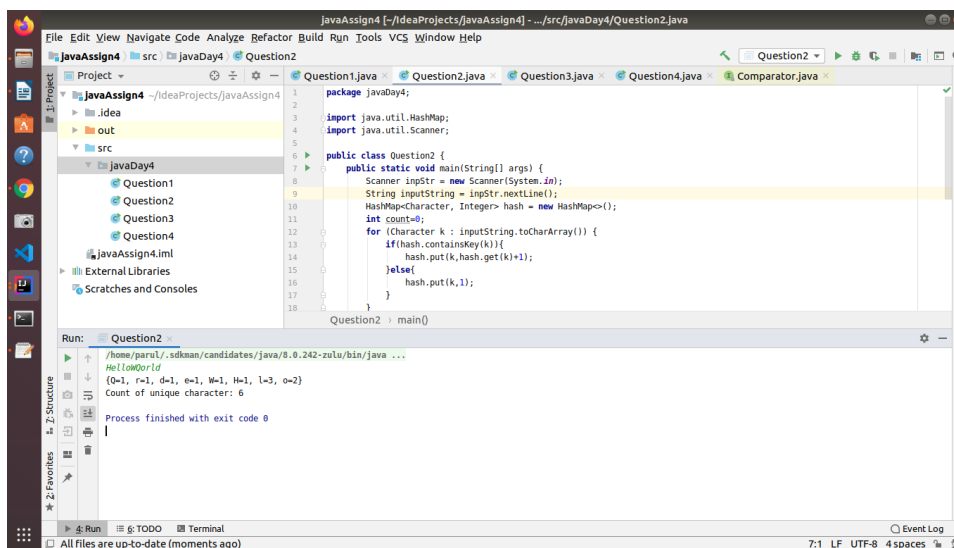
Write a method that takes a string and returns the number of unique characters in the string.



```
1 package javaDay4;
2
3 import java.util.HashMap;
4 import java.util.Scanner;
5
6 public class Question2 {
7     public static void main(String[] args) {
8         Scanner inputStr = new Scanner(System.in);
9         String inputString = inputStr.nextLine();
10        HashMap<Character, Integer> hash = new HashMap<>();
11        int count=0;
12        for (Character k : inputString.toCharArray()) {
13            if(hash.containsKey(k)){
14                hash.put(k,hash.get(k)+1);
15            }else{
16                hash.put(k,1);
17            }
18        }
19        System.out.println(hash);
20        for (Character k : hash.keySet()) {
21            if(hash.get(k)!=1){
22                count++;
23            }
24        }
25        System.out.println("Count of unique character: " + count);
26    }
27 }
28
29
30
31 }
```

Run: Question4

Build completed successfully in 3 s 827 ms (moments ago)



```
1 package javaDay4;
2
3 import java.util.HashMap;
4 import java.util.Scanner;
5
6 public class Question2 {
7     public static void main(String[] args) {
8         Scanner inputStr = new Scanner(System.in);
9         String inputString = inputStr.nextLine();
10        HashMap<Character, Integer> hash = new HashMap<>();
11        int count=0;
12        for (Character k : inputString.toCharArray()) {
13            if(hash.containsKey(k)){
14                hash.put(k,hash.get(k)+1);
15            }else{
16                hash.put(k,1);
17            }
18        }
19    }
20 }
```

Question2 -> main()

Run: Question2

/home/parul/.sdkman/candidates/java/8.0.242-zulu/bin/java ...

HelloWorld

{0=1, r=1, d=1, e=1, W=1, l=3, o=2}

Count of unique character: 6

Process finished with exit code 0

Question: 3:

Write a method that takes a string and print the number of occurrence of each character characters in the string.

```
package javaDay4;
import java.util.HashMap;
import java.util.Scanner;

public class Question3 {
    public static void main(String[] args) {
        Scanner inputStr = new Scanner(System.in);
        String inputString = inputStr.nextLine();
        HashMap<Character, Integer> hash = new HashMap<>();
        int count=0;
        for (Character k : inputString.toCharArray()) {
            if(hash.containsKey(k)){
                hash.put(k,hash.get(k)+1);
            }else{
                hash.put(k,1);
            }
        }
        System.out.println(hash);
    }
}
```

```
package javaDay4;
import java.util.HashMap;
import java.util.Scanner;

public class Question3 {
    public static void main(String[] args) {
        Scanner inputStr = new Scanner(System.in);
        String inputString = inputStr.nextLine();
        HashMap<Character, Integer> hash = new HashMap<>();
        int count=0;
        for (Character k : inputString.toCharArray()) {
            if(hash.containsKey(k)){
                hash.put(k,hash.get(k)+1);
            }else{
                hash.put(k,1);
            }
        }
        System.out.println(hash);
    }
}
```

Run: Question3
/home/parul/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
HelloJava
{a=2, e=1, v=1, h=1, j=1, l=2, o=1}
Process finished with exit code 0

Question: 4:
Write a program to sort HashMap by value.

```

1 package javaDay4;
2
3 import java.util.*;
4
5 public class Question4 {
6     public static Map<String, Integer> sortByValue(Map<String, Integer> hash)
7     {
8         List<Map.Entry<String, Integer>> list = new LinkedList<>(hash.entrySet());
9
10        Collections.sort(list, new Comparator<Map.Entry<String, Integer>>() {
11            public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2)
12            {
13                return (o1.getValue()).compareTo(o2.getValue());
14            }
15        });
16
17        HashMap<String, Integer> tempMap = new LinkedHashMap<>();
18        for (Map.Entry<String, Integer> l : list) {
19            tempMap.put(l.getKey(), l.getValue());
20        }
21        return tempMap;
22    }
23
24    public static void main(String[] args) {
25        Map<String, Integer> hash = new HashMap<>();
26        hash.put("1", 200);
27        hash.put("2", 300);
28        hash.put("3", 400);
29        hash.put("3", 250);
30        System.out.println("Original Values");
31    }
32 }

```

Run: Question4

Build completed successfully in 1 s 211 ms (moments ago)

```

21        return tempMap;
22    }
23
24    public static void main(String[] args) {
25        Map<String, Integer> hash = new HashMap<>();
26        hash.put("1", 200);
27        hash.put("2", 300);
28        hash.put("3", 400);
29        hash.put("3", 250);
30        System.out.println("Original Values");
31        System.out.println(hash);
32        System.out.println("After Sorting");
33        hash = sortByValue(hash);
34        System.out.println(hash);
35    }
36 }

```

Run: Question4

```

/home/parul/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Original Values
{1=200, 2=300, 3=250, 3=400}
After Sorting
{1=200, 3=250, 2=300, 3=400}
Process finished with exit code 0

```

Build completed successfully in 1 s 211 ms (a minute ago)

Question: 5:

Write a program to sort Employee objects based on highest salary using Comparator. Employee class{ Double Age; Double Salary; String Name

```

package javaDay4;

import java.util.*;

class Employee {
    Double age;
    Double salary;
    String empName;
    public Employee(Double age, Double salary, String empName) {
        this.age = age;
        this.salary = salary;
        this.empName = empName;
    }

    @Override
    public String toString() {
        return "Employee{" +
            "age=" + age +
            ", salary=" + salary +
            ", empName=" + empName + '\'' +
            '}';
    }
}

public class Question5 {
    public static void main(String[] args) {
        List<Employee> list = new ArrayList<>(initialCapacity: 5);
        Employee E1 = new Employee((double)20, (double)20000, empName: "emp1");
        list.add(E1);
        Employee E2 = new Employee((double)22, (double)80000, empName: "emp2");
    }
}

```

Run: Question4

Build completed successfully in 1 s 211 ms (a minute ago)

```

package javaDay4;

import java.util.*;

class Employee {
    Double age;
    Double salary;
    String empName;
    public Employee(Double age, Double salary, String empName) {
        this.age = age;
        this.salary = salary;
        this.empName = empName;
    }

    @Override
    public String toString() {
        return "Employee{" +
            "age=" + age +
            ", salary=" + salary +
            ", empName=" + empName + '\'' +
            '}';
    }
}

public class Question5 {
    public static void main(String[] args) {
        List<Employee> list = new ArrayList<>(initialCapacity: 5);
        Employee E1 = new Employee((double)20, (double)20000, empName: "emp1");
        list.add(E1);
        Employee E2 = new Employee((double)22, (double)80000, empName: "emp2");
        list.add(E2);
        Employee E3 = new Employee((double)23, (double)70000, empName: "emp3");
        list.add(E3);
        Employee E4 = new Employee((double)21, (double)40000, empName: "emp4");
        list.add(E4);
        Employee E5 = new Employee((double)22, (double)25000, empName: "emp5");
        list.add(E5);

        for (Employee E : list) {
            System.out.println(E);
        }
        Collections.sort(list, new Comparator<Employee>() {
            @Override
            public int compare(Employee o1, Employee o2) {
                if (o1.salary > o2.salary) {
                    return 1;
                } else {
                    return -1;
                }
            }
        });
        System.out.println("After sorting on the basis of salary");
        for (Employee E : list) {
            System.out.println(E);
        }
    }
}

```

Run: Question4

Build completed successfully in 1 s 211 ms (a minute ago)

```

package javaDay4;

import java.util.*;

class Employee {
    Double age;
    Double salary;
    String empName;
    public Employee(Double age, Double salary, String empName) {
        this.age = age;
        this.salary = salary;
        this.empName = empName;
    }

    @Override
    public String toString() {
        return "Employee{" +
            "age=" + age +
            ", salary=" + salary +
            ", empName=" + empName + '\'' +
            '}';
    }
}

public class Question5 {
    public static void main(String[] args) {
        List<Employee> list = new ArrayList<>(initialCapacity: 5);
        Employee E1 = new Employee((double)20, (double)20000, empName: "emp1");
        list.add(E1);
        Employee E2 = new Employee((double)22, (double)80000, empName: "emp2");
        list.add(E2);
        Employee E3 = new Employee((double)23, (double)70000, empName: "emp3");
        list.add(E3);
        Employee E4 = new Employee((double)21, (double)40000, empName: "emp4");
        list.add(E4);
        Employee E5 = new Employee((double)22, (double)25000, empName: "emp5");
        list.add(E5);

        Collections.sort(list, new Comparator<Employee>() {
            @Override
            public int compare(Employee o1, Employee o2) {
                if (o1.salary > o2.salary) {
                    return 1;
                } else {
                    return -1;
                }
            }
        });
        System.out.println("After sorting on the basis of salary");
        for (Employee E : list) {
            System.out.println(E);
        }
    }
}

```

Run: Question5

```

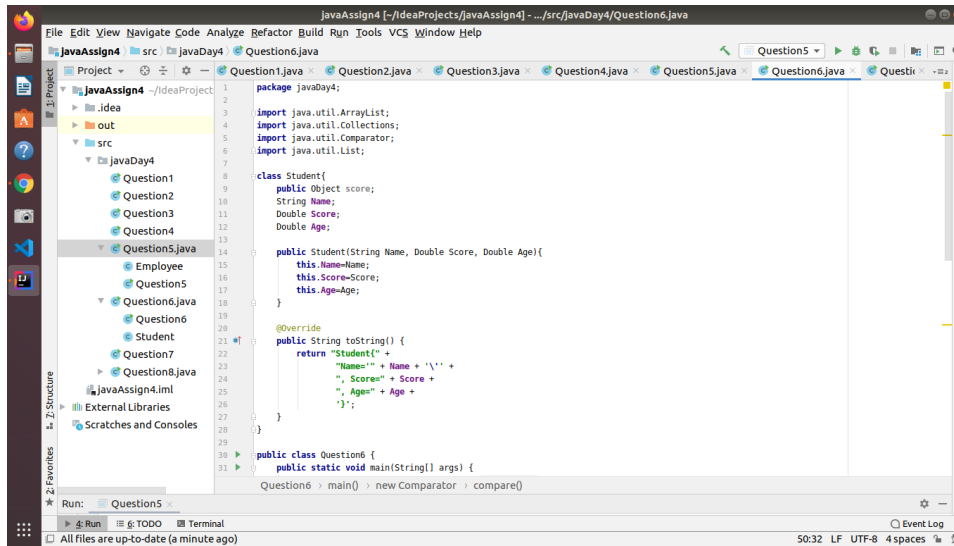
/home/parul/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Employee{age=20.0, salary=20000.0, empName='emp1'}
Employee{age=22.0, salary=80000.0, empName='emp2'}
Employee{age=23.0, salary=70000.0, empName='emp3'}
Employee{age=21.0, salary=40000.0, empName='emp4'}
Employee{age=22.0, salary=25000.0, empName='emp5'}
After sorting on the basis of salary
Employee{age=20.0, salary=20000.0, empName='emp1'}
Employee{age=22.0, salary=25000.0, empName='emp5'}
Employee{age=21.0, salary=40000.0, empName='emp4'}
Employee{age=23.0, salary=70000.0, empName='emp3'}
Employee{age=22.0, salary=80000.0, empName='emp2'}
Process finished with exit code 0

```

All Files are up-to-date (moments ago)

Question: 6:

Write a program to sort the Student objects based on Score , if the score are same then sort on First Name . Class Student{ String Name; Double Score; Double Age



```
package javaDay4;

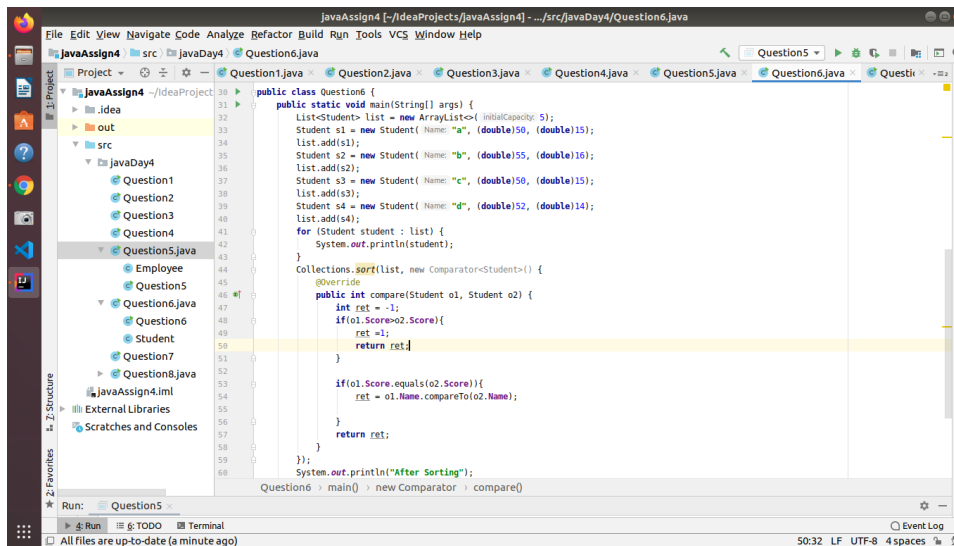
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;

class Student {
    public Object score;
    String Name;
    Double Score;
    Double Age;

    public Student(String Name, Double Score, Double Age) {
        this.Name = Name;
        this.Score = Score;
        this.Age = Age;
    }

    @Override
    public String toString() {
        return "Student{" +
            "Name=" + Name + ", " +
            "Score=" + Score + ", " +
            "Age=" + Age +
            '}';
    }
}

public class Question6 {
    public static void main(String[] args) {
        Question6 main() { new Comparator > compare()
    }
}
```



```
public class Question6 {
    public static void main(String[] args) {
        List<Student> list = new ArrayList<> (initialCapacity: 5);
        Student s1 = new Student( Name: "a", (double)50, (double)15);
        list.add(s1);
        Student s2 = new Student( Name: "b", (double)55, (double)16);
        list.add(s2);
        Student s3 = new Student( Name: "c", (double)50, (double)15);
        list.add(s3);
        Student s4 = new Student( Name: "d", (double)52, (double)14);
        list.add(s4);
        for (Student student : list) {
            System.out.println(student);
        }
        Collections.sort(list, new Comparator<Student>() {
            @Override
            public int compare(Student o1, Student o2) {
                int ret = -1;
                if (o1.Score > o2.Score) {
                    ret = 1;
                }
                return ret;
            }
        });
        if (o1.Score.equals(o2.Score)) {
            ret = o1.Name.compareTo(o2.Name);
        }
        return ret;
    }
}

System.out.println("After Sorting");
Question6 main() { new Comparator > compare()
}
```

```

1  if(o1.Score.equals(o2.Score)){
2      ret = o1.Name.compareTo(o2.Name);
3  }
4  }
5  return ret;
6  });
7  System.out.println("After Sorting");
8  for (Student student : list) {
9      System.out.println(student);
10 }
11 }
12 }
13 }
14 }
15 }
16 }
17 }
18 }
19 }
20 }
21 }
22 }
23 }
24 }
25 }
26 }
27 }
28 }
29 }
30 }
31 }
32 }
33 }
34 }
35 }
36 }
37 }
38 }
39 }
40 }
41 }
42 }
43 }
44 }
45 }
46 }
47 }
48 }
49 }
50 }
51 }
52 }
53 }
54 }
55 }
56 }
57 }
58 }
59 }
60 }
61 }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }

```

Run: Question6 - main()

```

/home/parul/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Student(Name="a", Score=50.0, Age=15.0)
Student(Name="b", Score=55.0, Age=16.0)
Student(Name="c", Score=50.0, Age=15.0)
Student(Name="d", Score=52.0, Age=14.0)
After Sorting
Student(Name="a", Score=50.0, Age=15.0)
Student(Name="c", Score=50.0, Age=15.0)
Student(Name="d", Score=52.0, Age=14.0)
Student(Name="b", Score=55.0, Age=16.0)
Process finished with exit code 0

```

Question: 7:

Print the elements of an array in the decreasing frequency if 2 numbers have same frequency then print the one which came first.

```

1  package javaDay4;
2
3  import java.util.*;
4
5  public class Question7 {
6      public static void main(String[] args) {
7          Scanner inp = new Scanner(System.in);
8          String[] str = inp.nextLine().split(" ");
9          Map<String, Integer> noOfOccurrences = occurrences(str);
10         for (String k : noOfOccurrences.keySet()) {
11             System.out.println(k + " : " + noOfOccurrences.get(k));
12         }
13     }
14
15     static Map occurrences(String[] arr) {
16         Map<String, Integer> hash = new LinkedHashMap<>();
17         for (String c : arr) {
18             if (hash.containsKey(c)) {
19                 hash.put(c, hash.get(c) + 1);
20             } else {
21                 hash.put(c, 1);
22             }
23         }
24         hash = sortByValue(hash);
25         return hash;
26     }
27
28     public static Map<String, Integer> sortByValue(Map<String, Integer> hash) {
29         List<Map.Entry<String, Integer>> list = new LinkedList<>(hash.entrySet());
30         list.sort(new Comparator<>() {
31             public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2) {
32                 return o2.getValue().compareTo(o1.getValue());
33             }
34         });
35         Map<String, Integer> result = new LinkedHashMap<>();
36         for (Map.Entry<String, Integer> entry : list) {
37             result.put(entry.getKey(), entry.getValue());
38         }
39         return result;
40     }
41 }

```

Run: Question7 - main()

```

Process finished with exit code 0

```

```

27 }
28
29 public static Map<String, Integer> sortByValue(Map<String, Integer> hash) {
30     List<Map.Entry<String, Integer>> list = new LinkedList<>(hash.entrySet());
31
32     Collections.sort(list, new Comparator<Map.Entry<String, Integer>>() {
33         public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2) {
34             int ret = -1;
35             if (o2.getValue() > o1.getValue()) {
36                 ret = 1;
37             }
38             else if (o2.getValue() == o1.getValue()) {
39                 return 0;
40             }
41             return ret;
42         }
43     });
44
45     Map<String, Integer> tempMap = new LinkedHashMap<>();
46     for (Map.Entry<String, Integer> l : list) {
47         tempMap.put(l.getKey(), l.getValue());
48     }
49     return tempMap;
50 }
51 }

```

Run: Question6
Process finished with exit code 0

```

40 }
41
42 }
43
44 Map<String, Integer> tempMap = new LinkedHashMap<>();
45 for (Map.Entry<String, Integer> l : list) {
46     tempMap.put(l.getKey(), l.getValue());
47 }
48 return tempMap;
49 }
50 }
51 }

```

Run: Question7
/home/parul/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
2 2 1 1 4 4 4 5 6 6 7
4 : 3
2 : 2
1 : 2
6 : 2
5 : 1
7 : 1
Process finished with exit code 0

Question: 8:

Design a Data Structure SpecialStack that supports all the stack operations like push(), pop(), isEmpty(), isFull() and an additional operation getMin() which should return minimum element from the SpecialStack. (Expected complexity $O(1)$)

