# Session: Collections

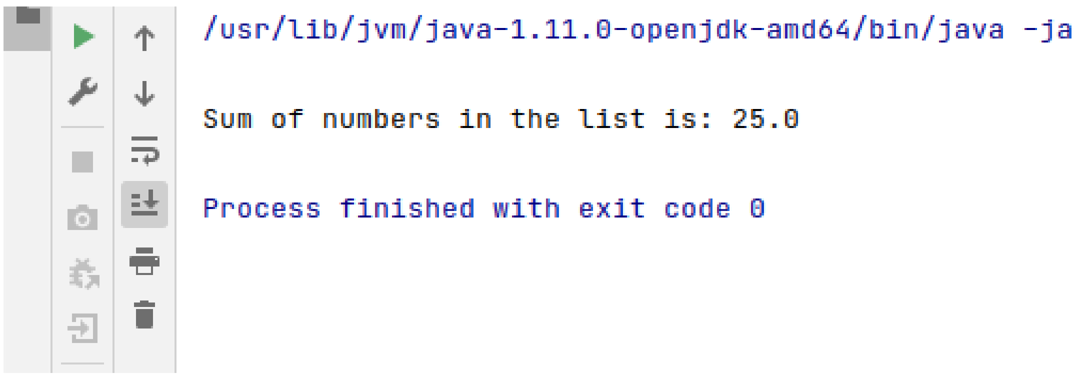
# Assignment

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.

**CODE**

```
package com.company;
import java.util.ArrayList;
import java.util.Iterator;
public class Ques1 {
    public static void main(String[] args) {

        ArrayList<Float> arrayList = new ArrayList<Float>();
        arrayList.add(3.2f);
        arrayList.add(1.8f);
        arrayList.add(6.4f);
        arrayList.add(3.6f);
        arrayList.add(10.0f);

        float sum = 0.0f;
        Iterator iterator = arrayList.iterator();
        while(iterator.hasNext()){
            sum = sum + (float) iterator.next();
        }
        System.out.println("\nSum of numbers in the list is: "+sum);
    }
}
```

**OUTPUT**



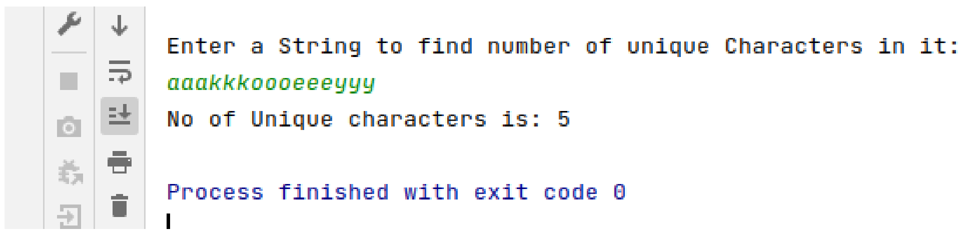
```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -ja

Sum of numbers in the list is: 25.0

Process finished with exit code 0
```

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

**CODE**

```
package com.company;
import java.util.*;
public class Ques2 {
   public static void main(String[] args) {

      Scanner scanner = new Scanner(System.in);
      System.out.println("\nEnter a String to find number of unique Characters in it: ");
      String str = scanner.nextLine();
      String unique[] = str.split("");
      int uval = findUnique(unique);
      System.out.println("No of Unique characters is: "+uval);
   }
   public static int findUnique(String[] str)
   {
      Set<String> set = new HashSet<String>(Arrays.asList(str));
      return set.size();
   }
}
```

**OUTPUT**



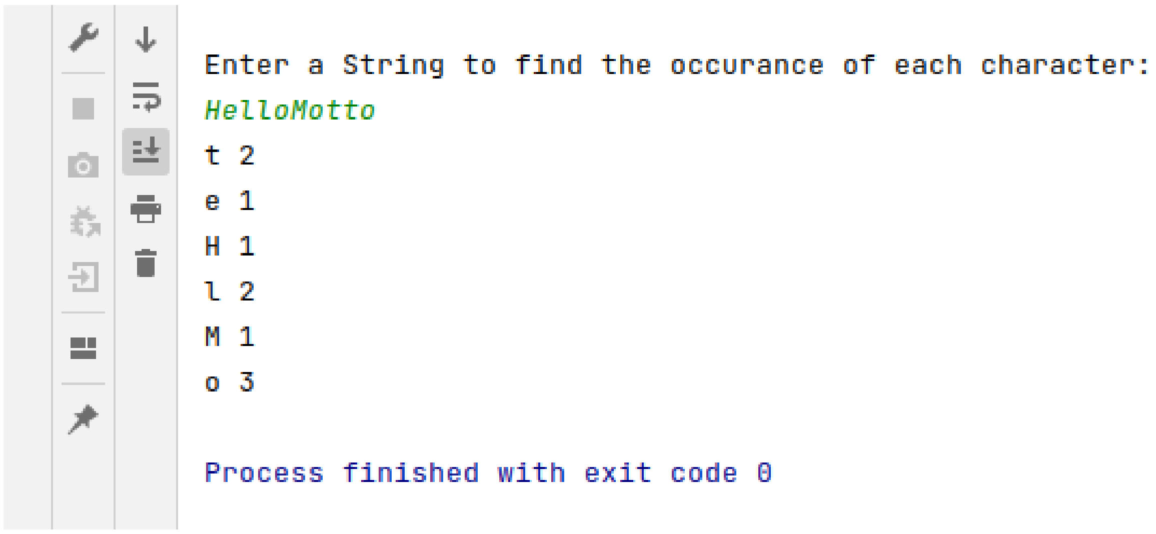
```
Enter a String to find number of unique Characters in it:
aaakkkoooeeeyyy
No of Unique characters is: 5

Process finished with exit code 0
```

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

**CODE**

```
package com.company;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class Ques3 {
   public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("\nEnter a String to find the occurance of each character: ");
        String str = scanner.nextLine();
        charOccur(str);
    }
    public static void charOccur(String str)
    {
        HashMap<Character, Integer> hashMap = new HashMap<Character,Integer>();
        char arr[] = str.toCharArray();
        for(char val: arr)
        {
            if(hashMap.containsKey(val))
            {
                hashMap.put(val, hashMap.get(val)+1);
            }
            else{
                hashMap.put(val,1);
            }
        }
        for(Map.Entry entry : hashMap.entrySet())
        {
            System.out.println(entry.getKey()+ " "+ entry.getValue());
        }
    }
}
```

## OUTPUT



```
Enter a String to find the occurance of each character:
HelloMotto
t 2
e 1
H 1
l 2
M 1
o 3

Process finished with exit code 0
```

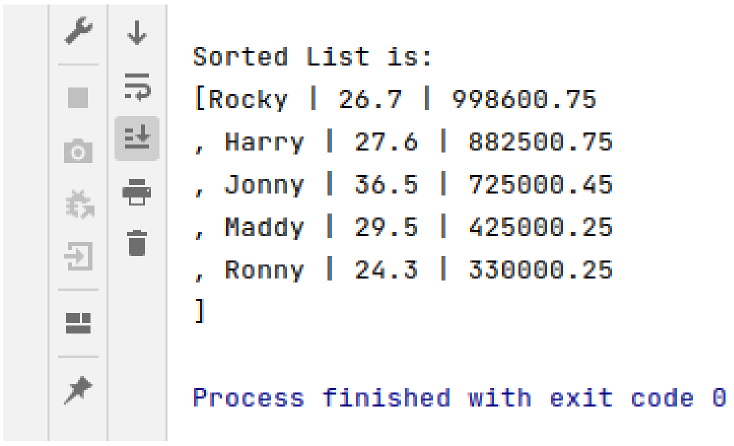
4. Write a program to sort Employee objects based on highest salary using Comparator. Employee class{ Double Age; Double Salary; String Name}

**CODE**

```
package com.company;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
class Employee
{
   String name;
   Double age, salary;

   Employee(String name, Double age, Double salary)
   {
      this.name = name;
      this.age = age;
      this.salary = salary;
   }
   public Double getSalary()
   {
      return salary;
   }
   public String toString()
   {
      return ""+name+" | "+age+" | "+salary+"\n";
   }
}

public class Ques4 {
   public static void main(String[] args) {

      ArrayList<Employee> arrayList = new ArrayList<Employee>();
      arrayList.add(new Employee("Jonny" , 36.5, 725000.45));
      arrayList.add(new Employee("Rocky" , 26.7, 998600.75));
      arrayList.add(new Employee("Maddy" , 29.5, 425000.25));
      arrayList.add(new Employee("Ronny" , 24.3, 330000.25));
      arrayList.add(new Employee("Harry" , 27.6, 882500.75));

      Comparator<Employee> salComp = new Comparator<Employee>() {
         @Override
         public int compare(Employee employee, Employee t1) {
            return t1.getSalary().compareTo(employee.getSalary()) ;
         }
```

```
        };
        Collections.sort(arrayList, salComp);
        System.out.println("\nSorted List is: \n"+arrayList);
    }
}
```

**OUTPUT**



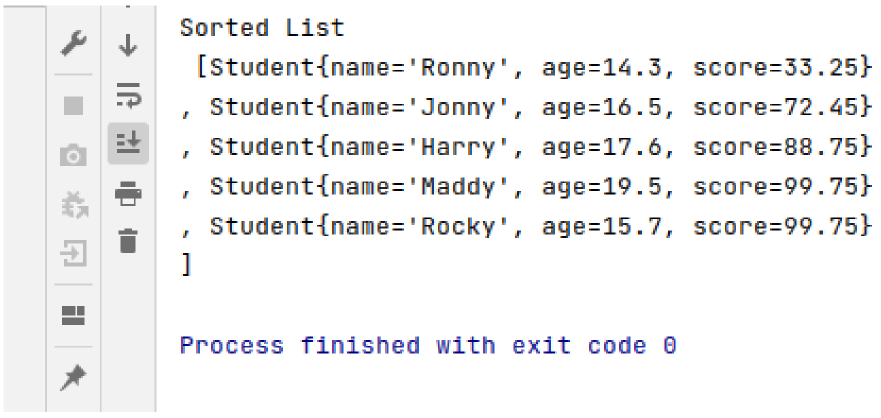
```
Sorted List is:
[Rocky | 26.7 | 998600.75
, Harry | 27.6 | 882500.75
, Jonny | 36.5 | 725000.45
, Maddy | 29.5 | 425000.25
, Ronny | 24.3 | 330000.25
]

Process finished with exit code 0
```

5. 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}

**CODE**

```
package com.company;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;

class Student
{
    String name;
    Double age, score;

    Student(String name, double age, double score)
    {
        this.name = name;
        this.age = age;
        this.score = score;
    }
    public Double getScore()
    {
        return score;
    }
    public String getName()
    {
        return name;
    }

    @Override
    public String toString() {
        return "Student{" +
                "name='" + name + '\'' +
                ", age=" + age +
                ", score=" + score +
                '}'+"\n";
    }
}

public class Ques5 {
    public static void main(String[] args) {

        ArrayList<Student> arrayList = new ArrayList<Student>();
        arrayList.add(new Student("Jonny" , 16.5, 72.45));
        arrayList.add(new Student("Rocky" , 15.7, 99.75));
        arrayList.add(new Student("Maddy" , 19.5, 99.75));
        arrayList.add(new Student("Ronny" , 14.3, 33.25));
        arrayList.add(new Student("Harry" , 17.6, 88.75));

        Comparator<Student> scoreComp = new Comparator<Student>() {
            @Override
            public int compare(Student student, Student t1) {
                return student.getScore().compareTo(t1.getScore()) ;
            }
        };
        Comparator<Student>       nameComp       =       scoreComp.thenComparing(new
Comparator<Student>() {
            @Override
            public int compare(Student student, Student t1) {
                return student.getName().compareTo(t1.getName()) ;
            }
        });
        Collections.sort(arrayList, nameComp);
        System.out.println("Sorted List \n "+arrayList);
    }
}
```

**OUTPUT**



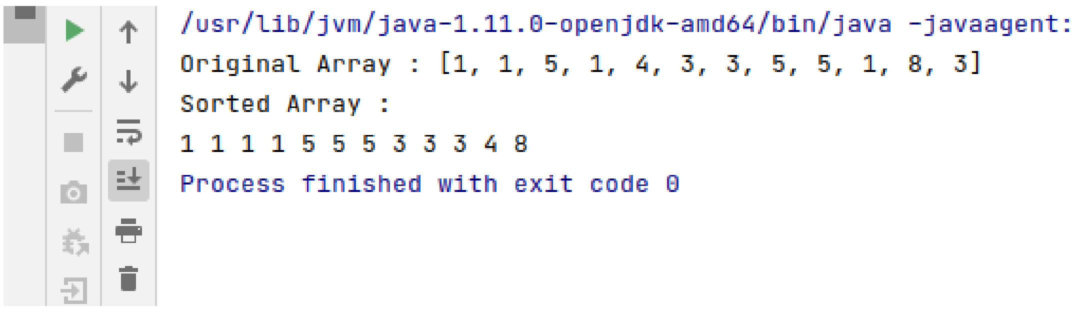
```
Sorted List
 [Student{name='Ronny', age=14.3, score=33.25}
, Student{name='Jonny', age=16.5, score=72.45}
, Student{name='Harry', age=17.6, score=88.75}
, Student{name='Maddy', age=19.5, score=99.75}
, Student{name='Rocky', age=15.7, score=99.75}
]

Process finished with exit code 0
```

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

**CODE**

```
package com.company;

import java.util.*;

public class Ques6 {
    public static void main(String[] args) {

        int[] arr1 = new int[]{1, 1, 5, 1, 4, 3, 3, 5, 5, 1, 8, 3};
        System.out.println("Original Array : " + Arrays.toString(arr1));
        sortByFreq(arr1);

    }
    public static void sortByFreq(int[] arr) {
        Map<Integer, Integer> mp = new LinkedHashMap<>();
        for (int i = 0; i < arr.length; i++) {
            if (mp.containsKey(arr[i])) {
                mp.put(arr[i], mp.get(arr[i]) + 1);
            } else {
                mp.put(arr[i], 1);
            }
        }
        ArrayList<Map.Entry<Integer, Integer>> list = new ArrayList<>(mp.entrySet());
        Comparator<Map.Entry<Integer, Integer>> comp = new
Comparator<Map.Entry<Integer, Integer>>() {
            @Override
            public int compare(Map.Entry<Integer, Integer> o1, Map.Entry<Integer, Integer>
o2) {
                return o2.getValue().compareTo(o1.getValue());
            }
        };
        Collections.sort(list, comp);
        System.out.println("Sorted Array : ");
        for (Map.Entry<Integer, Integer> entry : list) {
            int freq = entry.getValue();
            while (freq >= 1) {
                System.out.print(entry.getKey() + " ");
                freq--;
            }
        }
    }
}
```

**OUTPUT**



```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:
Original Array : [1, 1, 5, 1, 4, 3, 3, 5, 5, 1, 8, 3]
Sorted Array :
1 1 1 1 5 5 5 3 3 3 4 8
Process finished with exit code 0
```

7. 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))

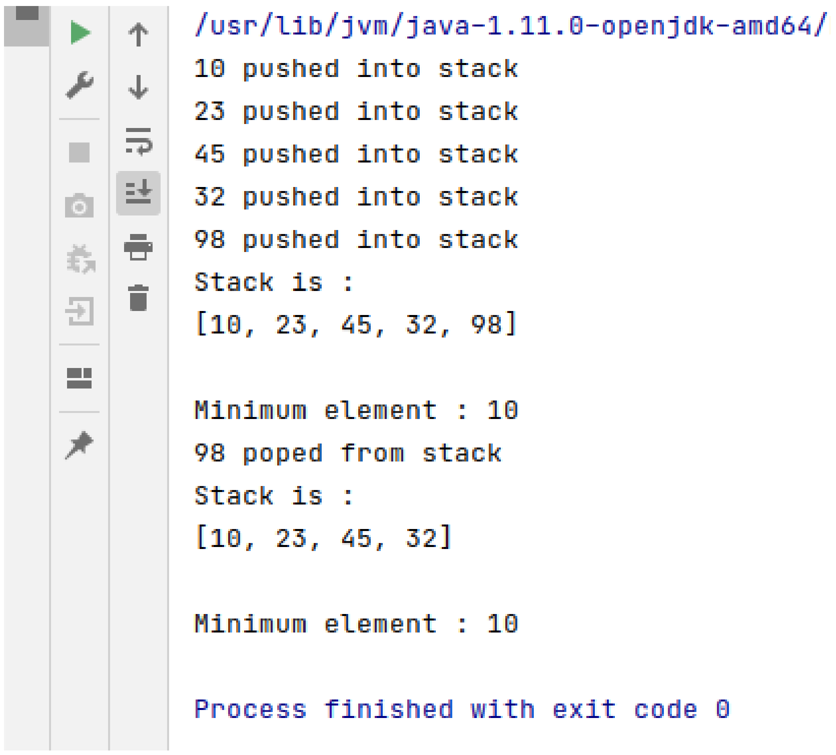
**CODE**

```
package com.company;

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

class SpecialStack
{
    private int size;
    private List<Integer> stack = new ArrayList<Integer>(size);
    private int min=0;

    public SpecialStack(int size) {
        this.size = size;
```

```
}

public boolean isEmpty()
{
   return stack.isEmpty();
}

public boolean isFull()
{
   if(stack.size()==size)
      return true;
   else
      return false;
}

public void push(int num)
{
   if(isFull())
   {
      System.out.println("Stack is full. " +num+" cannot be added.");
   }
   else
   {
      stack.add(num);
      System.out.println(num+" pushed into stack");
   }
}

public int pop()
{
   if(isEmpty())
   {
      System.out.println("Stack is empty. No more elements can be pop.");
      return 0;
   }
   else{
      int item = stack.get(stack.size()-1);
      stack.remove(stack.size()-1);
      System.out.println(item+" poped from stack");
      return item;
   }
}

public void show()
{
   System.out.println("Stack is :");
   System.out.println(stack);
   System.out.println();
```

```
    }

    public void getMin()
    {
        System.out.println("Minimum element : " + Collections.min(stack));
    }

}

public class Ques7 {
    public static void main(String[] args) {

        SpecialStack specialStack = new SpecialStack(7);
        specialStack.push(10);
        specialStack.push(23);
        specialStack.push(45);
        specialStack.push(32);
        specialStack.push(98);
        specialStack.show();
        specialStack.getMin();
        specialStack.pop();
        specialStack.show();
        specialStack.getMin();
    }
}
```

## OUTPUT



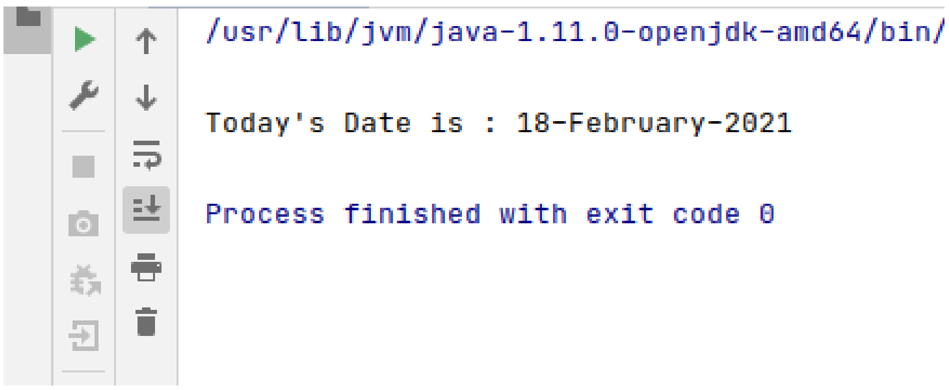
```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/
10 pushed into stack
23 pushed into stack
45 pushed into stack
32 pushed into stack
98 pushed into stack
Stack is :
[10, 23, 45, 32, 98]

Minimum element : 10
98 poped from stack
Stack is :
[10, 23, 45, 32]

Minimum element : 10

Process finished with exit code 0
```

8. Write a program to format date as example "21-March-2016"

**CODE**

```
package com.company;
import java.text.SimpleDateFormat;
import java.util.Date;
public class Ques8 {
    public static void main(String[] args) {
        String format = "dd-MMMM-yyyy";
        SimpleDateFormat simpleDateFormat = new SimpleDateFormat(format);

        String date = simpleDateFormat.format(new Date());
        System.out.println("\nToday's Date is : " +date);
    }
}
```

**OUTPUT**



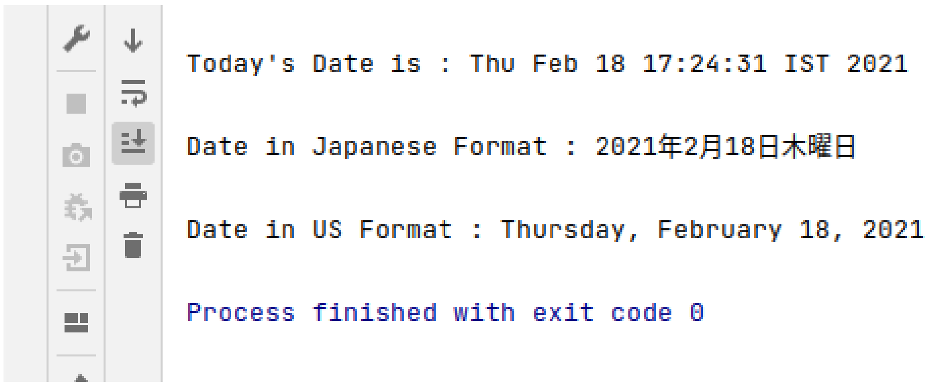
```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/

Today's Date is : 18-February-2021

Process finished with exit code 0
```

9. Write a program to display times in different country format.

**CODE**

```
package com.company;
import java.text.DateFormat;
import java.util.*;
public class Ques9 {
    public static void main(String[] args) {

        Date date = new Date();
        System.out.println("\nToday's Date is : "+ date.toString());
        Locale local = new Locale("ja","JA");
        DateFormat df = DateFormat.getDateInstance
                (DateFormat.FULL, local);
        System.out.println("\nDate in Japanese Format : "+ df.format(date));
        local = new Locale("en","US");
        df = DateFormat.getDateInstance
              (DateFormat.FULL, local);
        System.out.println("\nDate in US Format : "+ df.format(date));
    }
}
```

## OUTPUT



```
Today's Date is : Thu Feb 18 17:24:31 IST 2021

Date in Japanese Format : 2021年2月18日木曜日

Date in US Format : Thursday, February 18, 2021

Process finished with exit code 0
```