**Collections:**

**1. Write a program that asks how many names the user wants to input, asks for those**

**names, and shows the names afterward. But it should only show names that are**

**longer than 5 characters using Array List**

**Sample Input: How many names do you want to input?**

**Enter name #1: Anu**

**Enter name #2: Pragathi**

**Enter name #3: Charlie**

**Enter name #4: Shiny**

**Sample Output: Names longer than 5 characters:**

**Pragathi**

**Charlie**

**package** com.collections;

**import** java.util.\*;

**public** **class** Col01 {

**public** **static** **void** main(String args[]) {

Scanner sc=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of names you want to store");

**int** n=sc.nextInt();

sc.nextLine();

ArrayList<String> arr=**new** ArrayList<>();

**for**(**int** i=0;i<n;i++) {

String name=sc.nextLine();

arr.add(name);

}

System.***out***.println("Names longer than 5 Characters");

**for**(String s:arr) {

**if**(s.length()>5) {

System.***out***.println(s);

}

}

}

}

Output:

Enter the number of names you want to store

4

anu

Pragathi

Charlie

Shiny

Names longer than 5 Characters

Pragathi

Charlie

Shiny

**2. Suppose you have a set s containing some strings. Write a code fragment that**

**uses an iterator to print out each element in s and also compute the result of**

**concatenating all of these strings together.**

**Sample Input & Output: Elements in the Set : Java**

**Elements in the Set : Hello**

**Elements in the Set : Programming**

**Elements in the Set : World**

**Concatenated result: Java Hello Programming World**

**package** com.collections;

**import** java.util.\*;

**public** **class** Col02 {

**public** **static** **void** main(String[] args) {

Set<String> s = **new** LinkedHashSet<>();

s.add("Java");

s.add("Hello");

s.add("Programming");

s.add("World");

String concatenatedResult = "";

Iterator<String> iterator = s.iterator();

**while** (iterator.hasNext()) {

String element = iterator.next();

System.***out***.println("Elements in the Set : " + element);

concatenatedResult += element + " ";

}

System.***out***.println("Concatenated result: " + concatenatedResult.trim());

}

}

Output:

Elements in the Set : Java

Elements in the Set : Hello

Elements in the Set : Programming

Elements in the Set : World

Concatenated result: Java Hello Programming World

**3. Write a code and print the count (number of occurrences) of characters a,c,o, and**

**s in the string (using a hash Map).**

**a. String text = “Write Java code to define yet another Set s. Insert 3 floating**

**point numbers in s, and using an iterator, find the sum of the numbers in s.”**

**Sample Input & Output: Character counts:**

**a: 7**

**c: 1**

**s: 8**

**o: 7**

**package** com.collections;

**import** java.util.\*;

**public** **class** Col03 {

**public** **static** **void** main(String[] args) {

String text = "Write Java code to define yet another Set s. Insert 3 floating \r\n"

+ "point numbers in s, and using an iterator, find the sum of the numbers in s.";

**char** targetChar[]= {'a','c','s','o'};

HashMap<Character, Integer> charCount =**new** HashMap<>();

**for**(**char** ch: targetChar) {

charCount.put(ch,0);

}

**for**(**char** ch:text.toCharArray()) {

**if**(charCount.containsKey(ch)) {

charCount.put(ch, charCount.get(ch)+1);

}

}

System.***out***.println("character count:");

**for**(**char** ch:targetChar) {

System.***out***.println(ch+": "+charCount.get(ch));

}

}

}

Output:

character count:

a: 7

c: 1

s: 8

o: 7

**4. Julie has created a HashMap containing the name and marks as key-value pairs**

**of 6 students. Write a program, with separate user-defined functions to perform the**

**following operations:**

**a. Push the keys (name of the student) of the hash map into a stack, where**

**the corresponding value(marks) is greater than 75.**

**b. Pop and display the content of the stack.**

**Sample Input & Output:**

**Student Marks in the HashMap: {Mano=85, Shan=85, John=55,**

**Anu=60, Aju=90, Frank=80}**

**Student Marks above 75 in the stack:**

**Frank**

**Aju**

**Shan**

**Mano**

**package** com.collections;

**import** java.util.\*;

**public** **class** Col04 {

**public** **static** **void** main(String[] args) {

HashMap<String, Integer> studentMarks = **new** HashMap<>();

studentMarks.put("Mano", 85);

studentMarks.put("Shan", 85);

studentMarks.put("John", 55);

studentMarks.put("Anu", 60);

studentMarks.put("Aju", 90);

studentMarks.put("Frank", 80);

System.***out***.println("Student Marks in the HashMap: " + studentMarks);

Stack<String> stack = **new** Stack<>();

**for** (Map.Entry<String, Integer> entry : studentMarks.entrySet()) {

**if** (entry.getValue() > 75) {

stack.push(entry.getKey());

}

}

System.***out***.println("Student Marks above 75 in the stack:");

**while** (!stack.isEmpty()) {

System.***out***.println(stack.pop());

}

}

}

Output:

Student Marks in the HashMap: {Mano=85, Shan=85, John=55, Anu=60, Aju=90, Frank=80}

Student Marks above 75 in the stack:

Frank

Aju

Shan

Mano

**5. Write a program that reads a sequence of numbers and displays all the numbers**

**in sorted order without duplicates. then: – Prints the average of the numbers. –**

**Prints the highest and lowest number. – Filters out all of the even numbers (ones**

**divisible by 2).**

**Sample Input:**

**Enter a number (or type 'done' to finish): 4**

**Enter a number (or type 'done' to finish): 7**

**Enter a number (or type 'done' to finish): 7**

**Enter a number (or type 'done' to finish): 5**

**Enter a number (or type 'done' to finish): 3**

**Enter a number (or type 'done' to finish): 9**

**Enter a number (or type 'done' to finish): 8**

**Enter a number (or type 'done' to finish): done**

**Sample Output: Sorted unique numbers: [3, 4, 5, 7, 8, 9]**

**Average: 6.0**

**Lowest: 3**

**Highest: 9**

**Odd numbers: [3, 5, 7, 9]**

**package** com.collections;

**import** java.util.\*;

**public** **class** Col05 {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

TreeSet<Integer> numbers = **new** TreeSet<>();

**int** sum = 0, count = 0;

**while** (**true**) {

System.***out***.print("Enter a number (or type 'done' to finish): ");

String input = sc.nextLine();

**if** (input.equalsIgnoreCase("done")) {

**break**;

}

**int** num = Integer.*parseInt*(input);

**if** (numbers.add(num)) {

sum += num;

count++;

}

}

**if** (!numbers.isEmpty()) {

System.***out***.println("Sorted unique numbers: " + numbers);

System.***out***.println("Average: " + ((**double**) sum / count));

System.***out***.println("Lowest: " + numbers.first());

System.***out***.println("Highest: " + numbers.last());

List<Integer> oddNumbers = **new** ArrayList<>();

**for** (**int** num : numbers) {

**if** (num % 2 != 0) {

oddNumbers.add(num);

}

}

System.***out***.println("Odd numbers: " + oddNumbers);

} **else** {

System.***out***.println("No numbers entered.");

}

}

}

Output:

Enter a number (or type 'done' to finish): 1

Enter a number (or type 'done' to finish): 1

Enter a number (or type 'done' to finish): 2

Enter a number (or type 'done' to finish): 4

Enter a number (or type 'done' to finish): 5

Enter a number (or type 'done' to finish): done

Sorted unique numbers: [1, 2, 4, 5]

Average: 3.0

Lowest: 1

Highest: 5

Odd numbers: [1, 5]

**6. Create a class, Person. A person has a first name and a last name. Write a main**

**program that creates an ArrayList with a number of people (e.g., Priscilla Wagner,**

**Tom Parker, Elvis Presley). Sort the ArrayList based on the first name and show**

**the result: Elvis Presley Priscilla Wagner Tom Parker. Use a Comparator or**

**Comparable Interface.**

**Sample Input & Output: Sorted people by first name:**

**Elvis Presley**

**Priscilla Wagner**

**Tom Parker**

package com.collections;

import java.util.\*;

class Person implements Comparable<Person> {

String firstName, lastName;

Person(String firstName, String lastName) {

this.firstName = firstName;

this.lastName = lastName;

}

public int compareTo(Person other) {

return this.firstName.compareTo(other.firstName);

}

public String toString() {

return firstName + " " + lastName;

}

}

public class Col06 {

public static void main(String[] args) {

ArrayList<Person> people = new ArrayList<>();

people.add(new Person("Priscilla", "Wagner"));

people.add(new Person("Tom", "Parker"));

people.add(new Person("Elvis", "Presley"));

Collections.*sort*(people);

System.*out*.println("Sorted people by first name:");

for (Person person : people) {

System.*out*.println(person);

}

}

}

Output:

Sorted people by first name:

Elvis Presley

Priscilla Wagner

Tom Parker