

2.

```
// Write a Resistor class containing the parameterized constructor, which takes in three  
// strings representing the three band colours. Write the methods to calculate and set the  
// resistance for the resistor.
```

```
import java.util.*;
```

```
class Resistor{
```

```
    HashMap<String,Integer> color;
```

```
    // data members
```

```
    private String band1;
```

```
    private String band2;
```

```
    private String band3;
```

```
    private double value; // Resistor value (Ohms)
```

```
    // default constructor
```

```
    public Resistor()
```

```
{
```

```
        band1=band2=band3="";
```

```
        value=0.0;
```

```
}
```

```
    // Constructor taking three strings as parameters
```

```
    // and calculating the resistance
```

```
    public Resistor(String b1, String b2, String b3)
```

```
{
```

```
    // *** to be completed ***
```

```
    band1=b1;
```

```
    band2=b2;
```

```
band3=b3;

color=new HashMap<>();

color.put("black",0);

color.put("brown",1);

color.put("red",2);

color.put("orange",3);

color.put("yellow",4);

color.put("green",5);

color.put("blue",6);

color.put("violet",7);

color.put("grey",8);

color.put("white",9);

}

public String getResistance(){

    int firstdig=color.get(band1);

    int seconddig=color.get(band2);

    int thirddig=(int)Math.pow(10,color.get(band3));

    String value1=Integer.toString(firstdig)+Integer.toString(seconddig);

    value=Integer.parseInt(value1)*thirddig;

    value=value/1000;

    value1=value+"Kilo_ohms";

    return value1;

}

}
```

```

public class lab2{

public static void main(String args[]){

Resistor r1=new Resistor("red","black","orange");

System.out.println(r1.getResistance());

}

}

```

6a.

```

// 6. a. Write a java program that loads names and phone numbers from the text file into Hash
// Table where data is organized as one line per record and each field in record are separated by
// a
// tab(\t). It takes a name or phone number as input and prints the corresponding other value from
// hash table.

import java.util.*;

import java.io.*;

public class lab6A

{ public static void main(String[] args) throws Exception

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

FileReader fr = new FileReader("phone.txt");

BufferedReader br=new BufferedReader(fr);

Hashtable ht=new Hashtable();

String line,name, mobile;

while((line=br.readLine())!=null)

{ String s[] = line.split("\\s+");

ht.put(s[0],s[1]); }

System.out.println(ht);

```

```

System.out.print("Enter name: ");

name = sc.next();

if (ht.containsKey(name))

{ System.out.println("Mobile number is " + ht.get(name)); }

else { System.out.println("Not Found"); }

System.out.print("Enter mobile number: ");

mobile = sc.next();

if (ht.containsValue(mobile))

{

Set<Map.Entry> s1=ht.entrySet();

for (Map.Entry e : s1)

{ if (mobile.equals(e.getValue()))

{ System.out.println("Name is " + e.getKey()); } } }

else { System.out.println("Not Found"); }

br.close(); } }

```

6b.

// Write a Java program to count the votes from the logfile, throwing out duplicate votes

// using the rules above. Display the votes in a table, as shown below for the example:

// PHP 2

// C# 1

// Prolog 1

import java.util.\*;

import java.io.\*;

import java.util.concurrent.CopyOnWriteArrayList;

class Data{

```
String plang,ip,timestamp;

void setData( String plang, String ip,String timestamp)

{

this.plang=plang;

this.ip=ip;

this.timestamp=timestamp;

}

String getPlang()

{


return this.plang;

}

String getIP()

{

return this.ip;

}

String getTimestamp()

{

return this.timestamp;

}

}

public class MyClass

{

public static void main(String[] args) throws Exception

{

Scanner sc = new Scanner(System.in);

FileReader fr = new FileReader("phone.txt");
```

```
BufferedReader br=new BufferedReader(fr);

String line;

String[] data;

//ArrayList<Data> al=new ArrayList<Data>();

List<Data>al = new CopyOnWriteArrayList<Data>();

int a =Integer.parseInt(br.readLine());

for(int i=0;i<a;i++){

    Data d=new Data();

    String r=br.readLine();

    String s=br.readLine();

    String t=br.readLine();

    d.setData(r,s,t);

    al.add(d);

}

//while((line=br.readLine())!=null) {

//data= line.split("\t");

//Data d=new Data();

//System.out.println(data[0]);

//d.setData(data[0],data[1],data[2]);

//al.add(d);

//}

String[] ar=new String[]{"PHP","C#","Prolog"};

int[] ar1=new int[]{1,1,1};

String[] ar2=new String[]{"137.229.156.12","137.229.156.18","156.213.38.31","128.120.56.214"};

int[] ar3=new int[]{0,0,0,0};
```

```
for(Data i : al){  
    String u=i.getPlang();  
    String w=i.getIP();  
    String x=i.getTimestamp();  
    int k =0;  
    for(int j=0;j<ar.length;j++){  
        if(ar[j]==u){  
            k=j;  
            break;  
        }  
    }  
    int z=0;  
  
    for(int j=0;j<ar2.length;j++){  
        if(ar2[j]==w){  
            z=j;  
            break;  
        }  
    }  
}
```

```
        if(ar3[z]==0){

            ar1[k]++;

            ar3[z]=1;

        }

    }

    for(int i=0;i<ar.length;i++){

        System.out.print(ar[i]);

        System.out.print(" ");

        System.out.print(ar1[i]);

        System.out.println();

    }

    for (Data d: al) {

        if (d.getPlang().equals(d.plang)) {

            if (d.getIP().equals(d.ip))

            {

                al.remove(d);

            }

        }

    }

    }}
```



7a.

```
// a. Write a Java Class to implement a method Addition() that returns a new Array where each  
// array element at the index k corresponds to the sum of elements of the array (src) starting at  
// index 0 and including element at the index 'k'. For example, for array [2,3,5], the method  
// will return array [2,5,10]. For an array of size '0' or a null parameter, the method will throw  
// exception IllegalArgumentExceptionWith the message "Invalid Argument".
```

```
import java.util.Arrays;  
import java.util.Scanner;  
  
class lab7A  
{  
    public static void main(String[] args)  
    {  
        System.out.println("Enter Array Size: ");  
        Scanner sc=new Scanner (System.in);  
        int n=sc.nextInt();  
        int arr[];  
        try  
        {  
            if (n>0)  
                arr=new int[n];  
            else  
                throw new IllegalArgumentException() ;  
            System.out.println("Enter Array Elements: ");  
            for (int i=0;i<n ;i++ )  
            {
```

```

arr[i]=sc.nextInt();
}

lab7A rs=new lab7A();

int resultarr[]=rs.addition(arr);

System.out.println(Arrays.toString(resultarr));
}

catch (IllegalArgumentException e)
{
System.out.println("Invalid Argument");
}
}

int[] addition(int[] a) {

int sum = 0;

for (int i = 0; i<a.length; i++) {

a[i] += sum;

sum = a[i];
}

return a;
}
}

```

7b.

```

// b. Write a Java Code to implement a multithreaded version of FizzBuzz with four threads.

// If the number is divisible by 3, output "fizz".

```

```

// If the number is divisible by 5, output "buzz".

// If the number is divisible by both 3 and 5, output "fizzbuzz".

// If the number is not divisible by both 3 and 5 print the number

// For instance if n is 15 , we will have the output as " 1, 2, fizz, 4, buzz, fizz, 7, 8, fizz,buzz,
// 11, fizz, 13, 14, fizzbuzz "

// 1. Thread A will call fizz() to check for divisibility of 3 and outputs fizz .

// 2. Thread B will call buzz() to check for divisibility of 5 and outputs buzz .

// 3. Thread C will call fizzbuzz() to check for divisibility of 3 and 5 and outputs fizzbuzz .

// 4. Thread D will call number() which should only output the numbers.

import java.util.Scanner;

class lab7B

{

public static void main(String[] args) throws Exception

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter n value: ");

int n=sc.nextInt();

for(int i=1;i<=n ;i++)

{

if (i%15 == 0)

{

C t1=new C();

t1.start();

```

```
t1.join();

}

else if (i%5==0)

{

    B t2=new B();

    t2.start();

    t2.join();

}

else if (i%3 == 0)

{

    A t3=new A();

    t3.start();

    t3.join();

}

else

System.out.print(i+" ");

}

}

}

class A extends Thread

{

    public void run()

    {

        System.out.print("fizz, ");

    }

}
```

```

}

class B extends Thread
{
    public void run()
    {
        System.out.print("buzz" + ", ");
    }
}

class C extends Thread
{
    public void run()
    {
        System.out.print("fizzbuzz" + ", ");
    }
}

```

8a.

```

// 8.a. Write a java program to store the employee details in an ArrayList and display the
// employee details in ascending order of their experience. Create 'Employee' class with two
// instance variables Employee name and Employee experience (no. of years).

import java.util.*;

class lab8A
{
    public static void main(String[] args)
    {

```

```
ArrayList<Employee>ar = new ArrayList<Employee>();
```

```
ar.add(new Employee("John",5));
```

```
ar.add(new Employee("Nic",3));
```

```
ar.add(new Employee("Roma",9));
```

```
System.out.println("Unsorted Emp List");
```

```
for (int i = 0; i<ar.size(); i++)
```

```
System.out.println(ar.get(i).toString());
```

```
Collections.sort(ar, new Sortbyexp());
```

```
System.out.println("Sorted by Experience");
```

```
for (int i = 0; i<ar.size(); i++)
```

```
System.out.println(ar.get(i).toString());
```

```
}
```

```
}
```

```
class Employee
```

```
{
```

```
String empname;
```

```
int exp;
```

```
Employee(String empname, int exp)
```

```
{
```

```
this.empname= empname;
```

```
this.exp=exp;
```

```
}
```

```
public String toString()
```

```
{
```

```
return this.empname + " " + this.exp ;
```

```

}

}

class Sortbyexp implements Comparator<Employee> {

    public int compare(Employee a, Employee b)

    {

        return a.exp - b.exp;

    }

}

```

8b.

```

// b. Write a program to find the most common words in the list of words given in sorted order
// based on occurrence from largest to smallest. If any of words are having same occurrence then
// consider the smallest character order comes first.

// Input format: First line contains the list of words and next line contains a number (k) which
// represent the top most words to display.

// Output format: display the k top most words.

import java.util.*;

class lab8B{

    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        String s=sc.nextLine();

        String arr[]=s.split(" ");

        Map <String,Integer>ch=new LinkedHashMap<String,Integer>();

        for(int i=0;i<arr.length;i++){

            String x=arr[i];

```

```
if(ch.containsKey(x)){  
  
    int c=ch.get(x);  
  
    c++;  
  
    ch.put(x,c);  
  
}  
  
else{  
  
    ch.put(x,1);  
  
}  
  
}  
  
ArrayList<Integer> al=new ArrayList<Integer>();  
  
Set<String> ss=ch.keySet();  
  
for(String i:ss){  
  
    if(al.contains(ch.get(i))){}  
  
    else{  
  
        al.add(ch.get(i));  
  
    }  
  
  
  
Collections.sort(al);  
  
for(int i=al.size()-1;i>=0;i--){  
  
    int temp= al.get(i);  
  
    ArrayList<String>tal=new ArrayList<>();  
  
    for(Map.Entry<String, Integer> l1 : ch.entrySet())  
  
    {  
  
        if(l1.getValue() == temp)  
  
        tal.add(l1.getKey());  
  
    }  
  
}
```



```

}

Collections.sort(tal);

for( String s1:tal)

System.out.print(s1+" ");

}

}

}

```

9a.

```

// 9. a. Write a java Program to write a method fCount which takes a string as a parameter.

// The Method fCount should return the Map which has the frequency count of the given

// word. For example if the string passed is "hello" the map should return {h-1,e-1,l-2,o-1}.

// The order of the characters should be same as in the string.

import java.util.*;

class lab9A

{

public static void main(String args[])

{

Scanner input = new Scanner(System.in);

System.out.print("Enter a string : ");

String str = input.next();

Map<Character, Integer> result = new HashMap<Character, Integer>();

result = fCount(str);

for(int i = 0; i<str.length(); i++)

{

```

```
if(result.get(str.charAt(i)) != 0)

{

System.out.print(str.charAt(i)+"-"+result.get(str.charAt(i)) + " ");

result.put(str.charAt(i), 0);

}

}

input.close();

}

public static Map<Character, Integer>fCount(String str)

{

Map<Character, Integer> map = new HashMap<Character, Integer>();

for(int i = 0; i<str.length(); i++)

{

if(map.containsKey(str.charAt(i)))

{

map.put(str.charAt(i), map.get(str.charAt(i)) + 1);

}

else

{

map.put(str.charAt(i), 1);

}

}

return map;

}

}
```

9b.

```
// In the example above, these two HashMaps have two key/value pairs in common, namely:  
  
// "Mary"/"Ecstatic" and "Bob"/"Happy". Note that although the key "Felix" is in both  
  
// HashMaps, the associated value with this key is different in the two maps (hence this does  
  
// not count as a key/value pair that is common to both HashMaps). Similarly, just having the  
  
// same value without the same key (such as the value "Fine" which is mapped to by different  
  
// keys in the two different HashMaps) would also not count as a common key/value pair  
  
// between the two HashMaps.
```

```
import java.util.*;  
  
class lab9B  
{  
  
    public static void main(String args[])  
    {  
  
        Scanner input = new Scanner(System.in);  
  
        HashMap<String, String>mapOne = new HashMap<String, String>();  
  
        HashMap<String, String>mapTwo = new HashMap<String, String>();  
  
        System.out.print("Enter how many pairs in HashMap 1 : ");  
  
        int mapOneSize = input.nextInt();  
  
        input.nextLine();  
  
        System.out.println("Enter your " + mapOneSize + " pair of String ");  
  
        for(int i = 0; i<mapOneSize; i++)  
        {  
  
            System.out.println("Enter Pair " + (i+1));  
  
            String pairs[] = input.nextLine().split(" ");
```

```

mapOne.put(pairs[0],pairs[1]);
}

System.out.print("Enter how many pairs in HashMap 2 : ");

int mapTwoSize = input.nextInt();

input.nextLine();

System.out.println("Enter your " + mapTwoSize + " pair of String ");

for(int i = 0; i<mapTwoSize; i++)
{
    System.out.println("Enter Pair " + (i+1));

    String pairs[] = input.nextLine().split(" ");

    mapTwo.put(pairs[0],pairs[1]);
}

int result = commonKeyValuePairs(mapOne,mapTwo);

System.out.println("The number of key/value pairs are : " + result);

input.close();
}

public static int commonKeyValuePairs(HashMap<String, String> map1 ,
HashMap<String, String> map2)
{
    int count = 0;

    for(Map.Entry m1:map1.entrySet())
    {
        for(Map.Entry m2:map2.entrySet())
        {
            if(m1.getKey().equals(m2.getKey())&&m1.getValue().equals(m2.getValue()))

```

```
{  
count++;  
}  
  
}  
  
}  
return count;  
}  
  
}
```