Assignment 1

Set a

1) Write a java program to accept names of “n” cities , insert same into arraylist collection and display the contents of same arraylist, also remove all these elements.

```
import java.util.*;
import java.io.*;
public class ArrayListDemo
{
public static void main(String args[])throws Exception
{
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
ArrayList al=new ArrayList();
System.out.println(“\nHow many City?”);
Int n =Integer.parseInt(br.readLine());
System.out.println(“\n Enter City names:”);
for(int i=;i<=n;i++)
{
al.add(br.readLine());
}
System.out.println(“Entered cities are :”+al);
al.removeAll(al);
System.out.println(“All cities are removed from the ArrayList:”+a);
}
}
```

---

2)Write a java program to read ‘n’ names of your friends , store it into linked list , also display contents of the same.

```
import java.util.*;
import java.io.*;
public class LinkedListDemo
{
Public static void main(String args[])throws Exception
{
int n;
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
LinkedList li=new LinkedList();
System.out.println(“\Enter number of your friends:”);
n=Interger.parseInt(br.readLine()));
System.out.println(“\Enter your friends names:”);
for(int i=1;i<=n;i++)
{
li.add(br.readLine());
}
System.out.println(“\Linked List content is:”);
}
}
```

---

3) Write a program to create a new tree set , add some colors(String)and print out the tree set.

```
import java.util.*;
class TreeSetDemo
{
Public static void main(String args[])
{
TreeSet ts=new TreeSet();
ts.add(“Red”);
ts.add(“Yellow”);
ts.add(“Blue”);
ts.add(“Black”);
ts.add(“Orange”);
ts.add(“Green”);
System.out.println(“TreeSet is:”+ts);
}
}
```

---

4) Create the hastable that will maintain the mobile number and student name. Display the contact list .

```
import java.util.*;
import java.io.*;
class HashTableDemo
{
Public static void main(String args[])
{
Hashtable ht= new Hashtable();
Enumeration names;
String str;
ht.put(“Amar”,new Long(2222222222L));
ht.put(“Anil”,new Long(3333333333L));
ht.put(“Soham”,new Long(8888888888L));
ht.put(“Kiran”,new Long(9999999999L));
ht.put(“Williams”,new Long(4444444444L));
names=ht.keys();
while(names.hasMoreElements());
{
str= (String) names.nextElement();
System.out.println(str+”:”+ht.get(str));
}}}
```

Set b

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

class SETB1
{
    public static void main(String[] args) throws Exception
    {
        int no,element,i;
            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
            TreeSet ts=new TreeSet();
            System.out.println("Enter the of elements :");
            no=Integer.parseInt(br.readLine());
            for(i=0;i<no;i++)
            {
                System.out.println("Enter the element : ");
                    element=Integer.parseInt(br.readLine());
                    ts.add(element);
            }

            System.out.println("The elements in sorted order :"+ts);
        System.out.println("Enter element to be serach : ");
        element = Integer.parseInt(br.readLine());
        if(ts.contains(element))
            System.out.println("Element is found");
        else
            System.out.println("Element is NOT found");
    }
}
```

2) Write a program to sort HashMap by keys and display the details before sorting and after sorting.

```
import java.util.*;
class HashMapDemo
{
Public static void main(String args[])
{
HashMap <Interger,String> ha=new HashMap<Interger,String>();
ha.put(10,”Java”);
ha.put(20,”Operating System”);
ha.put(30,”SoftwareTesting Tools”);
ha.put(40,”Complier Constructor”);
ha.put(50,”Web Technologies”);
System.out.println( “Before Sorting:”);
Set set=ha.entrySet();
Iterator it=set.iterator();
While(it.hasNext())
{
Map.Entry me=(Map.Entry)it.next();
System.out.print(me.getKeys()+”:”);

System.out.println(me.getValues());
}
Map<Integer,String> map=new HashMap<Integer,String>(hm);
System.out.println(“After Sorting:”);
Set set1=map.entrySet();
Iterator it2=set2.it();
While(it2.hasNext())
{
Map.Entry me2=(Map.Entry)it2.next();
System.out.print(me2.getKeys()+”:”);
System.out.prinln(me2.getValues());
}
}
}
```

Set-B-3)

```
import java.util.*;
import java.io.*;
public class Phonebook
{
Public static void main(String args[])
{
Try
{
FileInputStream fis=new FileInputStream(“home/desktop/myfile.txt”);
Scanner(fis).useDelimiter(“\t”);
Hashtable<String,String> ht=new Hashtable<String,String>();
String [] strarray;
String a,str;
While(sc.hasNext())
{
a=sc.nextLine();
strarray=a.split(\t”);
ht.put(strarray[0],strarray[1]);

System.out.println(“Hashtable values are:”+strarray[0]+”:”+strarray[1]);
Scanner s=new Scanner(System.in);
System.out.println("Enter the name as given in the phone book");
Str=s.next();
If(ht.containsKey(str))
{
System.out.println("Phone no is:"+ht,get(str));
}
else
{
System.out.println("Name is not Matching with the phone book");
}
}
Catch(Exception e)
{
System.out.println(e);
}
}
}
```

Assignment 2

1)

```
Class TestPrint extends Thread
{
String msg="";
int n;
TestPrint(String msg,int n)
{
this.msg=msg;
this.n=n;
}
Public void run()

{
Try
{
For(int i=1;i<=n;i++)
{
System.out.println(msg+” “+i+”times”);
}
System.out.println(“\n”);
}
Catch(Exception e)
{
}}}
Class DemoMythread
{
Public static void main(String args[]);
{
Int n = Integer.parseInt(args[0])
TestPrint t1=new TestPrint(“COVID”,n);
T1.start();
TestPrint t2=new TestPrint(“LOCKDOWN2020”,n+10);
T2.start();
TestPrint t3=new TestPrint(“Vaccinated2021”,n+20);
T3.start();
}
}
```

2)

```
class MythreadDemo
{
public static void main(String args[])
{
Thread t=Thread.currentThread();
System.out.prinltn("Current Thread is:"+t);
try
{
for(int n=100;n>0;n--)
{
System.out.println(n);
Thread.sleep(6000);
}
t.setName("Demo Thread");

System.out.prinltn("After changing the name thread is :"+t);
}

catch(InterruptedException e)
{

System.out.println("Thread interrupted");
}
}}
```

---

3)

```
public class Communication
{
```

```
public static void main(String args[])
{
    Producer p=new Producer();
    Consumer c=new Consumer(p);
    Thread t1=new Thread(p);
    Thread t2=new Thread(c);
    t1.start();
    t2.start();


}
}
class Producer extends Thread
{
    StringBuffer sb;
    boolean dataprodover=false;
    Producer()
    {
        sb=new StringBuffer();
    }
    public void run()
    {synchronized(sb)
        for(int i=1;i<=10;i++)
        {
            try
            {
                sb.append(i+":");
                Thread.sleep(100);
                System.out.println("appending");
```

```
                        }
                        catch(InterruptedException ae)
                        {
                                System.out.println(ae);
                        }
                }sb.notify();//dataprodover=true;
        }
}
class Consumer extends Thread
{
        Producer prod;
        Consumer(Producer prod)
        {
                this.prod=prod;
        }
        public void run()
        {synchronized(prod.sb)
                try
                {prod.sb.wait();
                        //while(!prod.dataprodover)
                                //Thread.sleep(100);
                }
                catch(InterruptedException ae)
                {
                        System.out.println(ae);
                }
                System.out.println(prod.sb);
        }
```

```
}
```

SET-B

1) Write a program to calculate the sum and average of an array of 1000 integers

(genearated randomly) using 10 threads. Each thread calculates the sum of 100 integers .

Use these values to calculate average(Use join method).

```
import java.util.*;
class thread implements Runnable
{
Thread t;
int i,no,sum;
int a[]=new int[1000];
thread(String s,int n)
{
Random rs = new Random();
t=new Thread(this,s);

no=n;
int j=0;

for(i=1;i<=1000;i++)
{
a[j]=rs.nextInt()%100;;
j++;
}
t.start();
}
public void run() {
```

```
for(i=0;i<100;i++)

{

sum=sum+a[no];

no++;

}

System.out.println("Sum = "+sum);

System.out.println("Avg ="+sum/100);


}

}

public class threaddemo

{

public static void main(String[] arg) throws InterruptedException

{

thread t1=new thread("g",1);

t1.t.join();

thread t2=new thread("r",100);

t2.t.join();

thread t3=new thread("s",200);

t3.t.join();

thread t4=new thread("t",300);

t4.t.join();

thread t5=new thread("p",400);

t5.t.join();

thread t6=new thread("p",500);

t5.t.join();

thread t7=new thread("p",600);

t5.t.join();

thread t8=new thread("p",700);
```

```
t5.t.join();
thread t9=new thread("p",800);
t5.t.join();
thread t10=new thread("p",900);
t5.t.join();
}
}
```

3) Write a java program that implements a multithread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

```
import java.util.Random;

class RandomNumberThread extends Thread
{
public void run()
{
Random random = new Random();
for (int i = 0; i &lt; 10; i++) {
int randomInteger = random.nextInt(100);
System.out.println(&quot;Random Integer generated : &quot; + randomInteger);
if((randomInteger%2) == 0) {
SquareThread sThread = new

SquareThread(randomInteger);

sThread.start();
}
else {
CubeThread cThread = new CubeThread(randomInteger);
cThread.start();
}
try {
Thread.sleep(1000);
}
catch (InterruptedException ex) {
System.out.println(ex);

}
}
}
}
```
