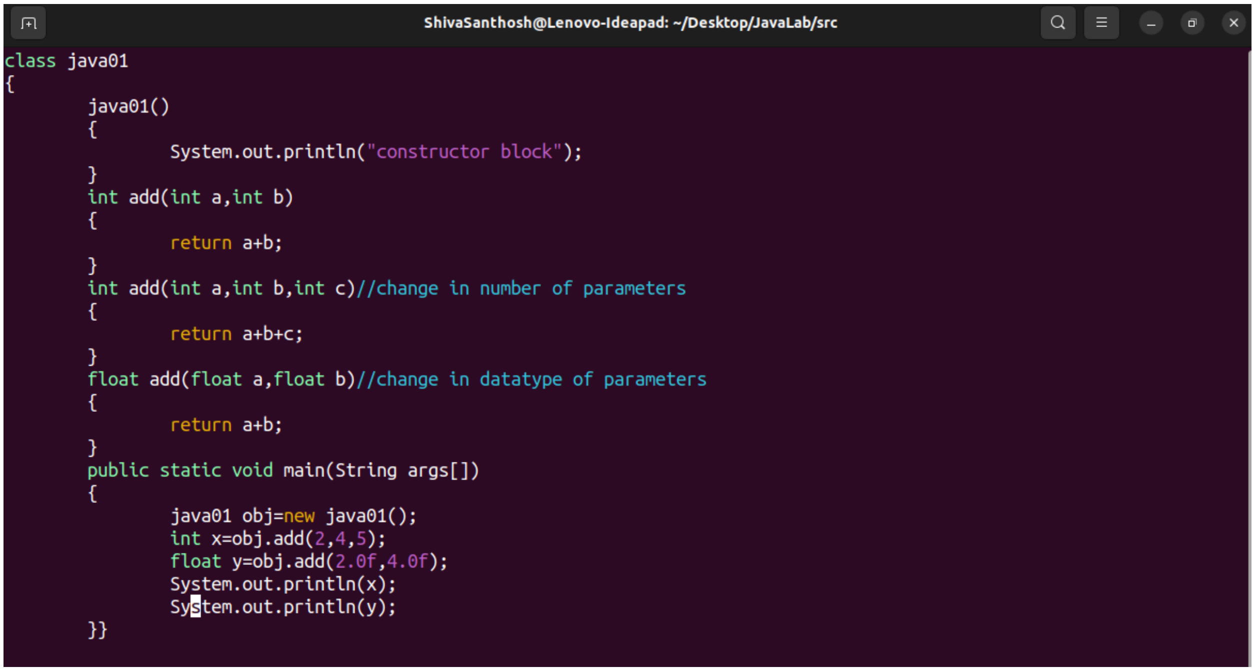
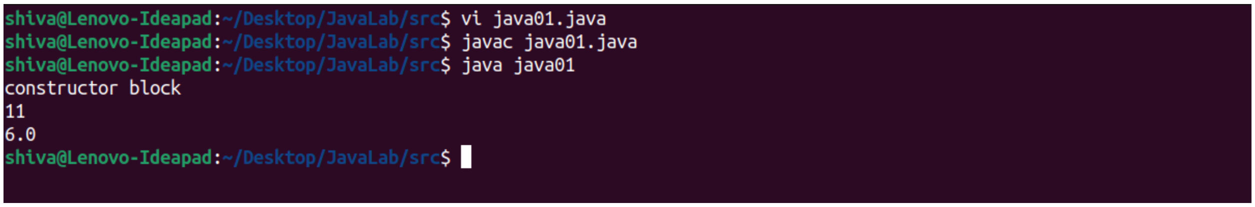
A program to illustrate the concept of class with constructors, methods and overloading.

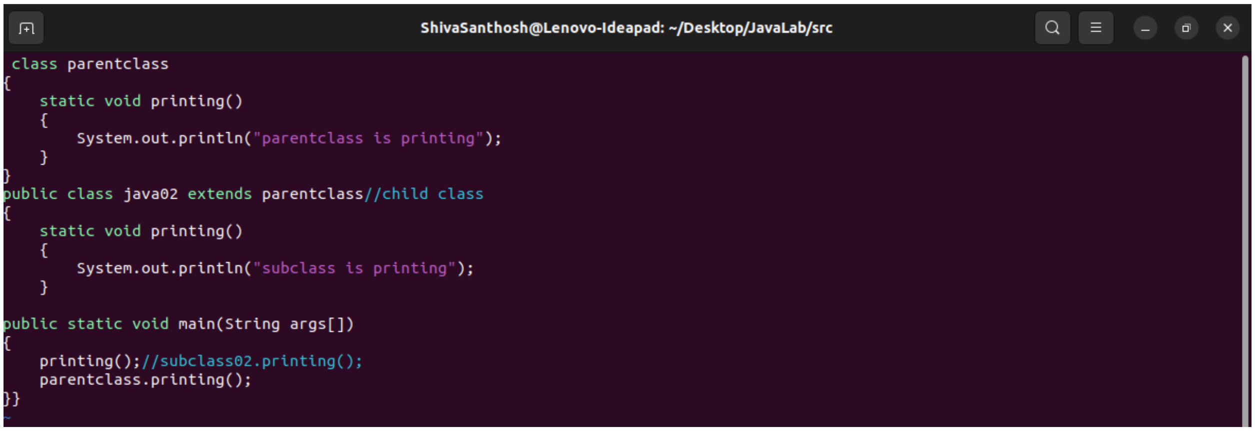


```
class java01
{
        java01()
        {
                System.out.println("constructor block");
        }
        int add(int a,int b)
        {
                return a+b;
        }
        int add(int a,int b,int c)//change in number of parameters
        {
                return a+b+c;
        }
        float add(float a,float b)//change in datatype of parameters
        {
                return a+b;
        }
        public static void main(String args[])
        {
                java01 obj=new java01();
                int x=obj.add(2,4,5);
                float y=obj.add(2.0f,4.0f);
                System.out.println(x);
                System.out.println(y);
        }}
```

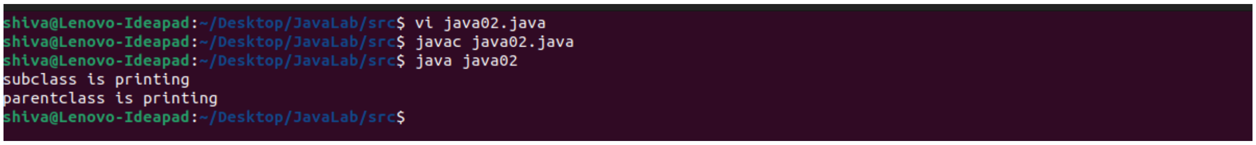


```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java01.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java01.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java01
constructor block
11
6.0
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

A program to illustrate the concept of Inheritance and Dynamic polymorphism.



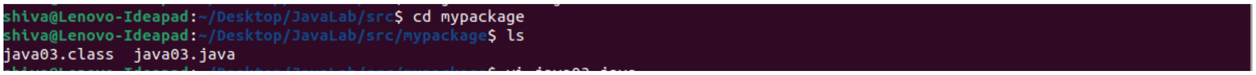
```
 class parentclass
{
    static void printing()
    {
        System.out.println("parentclass is printing");
    }
}
public class java02 extends parentclass//child class
{
    static void printing()
    {
        System.out.println("subclass is printing");
    }

public static void main(String args[])
{
    printing();//subclass02.printing();
    parentclass.printing();
}}
~
```



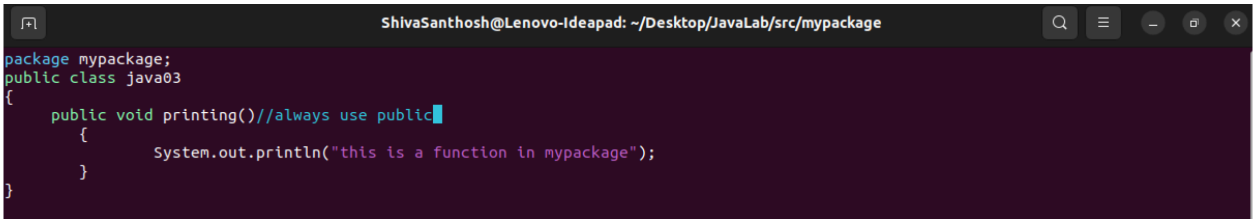
```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java02.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java02.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java02
subclass is printing
parentclass is printing
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

A program to show the concept of packages.

First create a directory “mypackage” and store file java03.java where package is stored



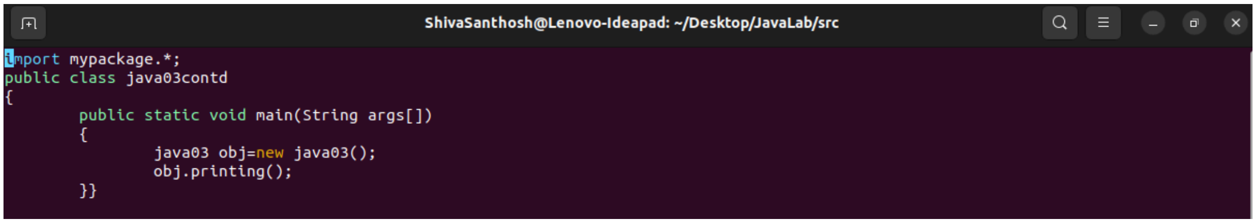
```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ cd mypackage
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src/mypackage$ ls
java03.class  java03.java
```



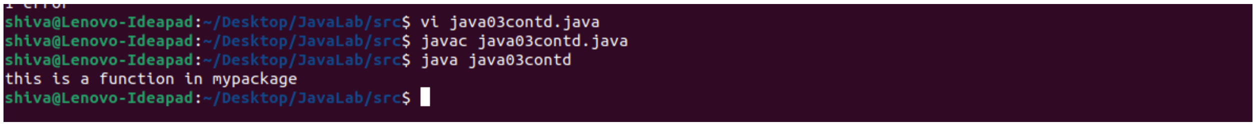
ShivaSanthosh@Lenovo-Ideapad: ~/Desktop/JavaLab/src/mypackage

```
package mypackage;
public class java03
{
     public void printing()//always use public
        {
                System.out.println("this is a function in mypackage");
        }
}
```

Now exit mypackage directory and create a java03cont.java file where u can import mypackage classes

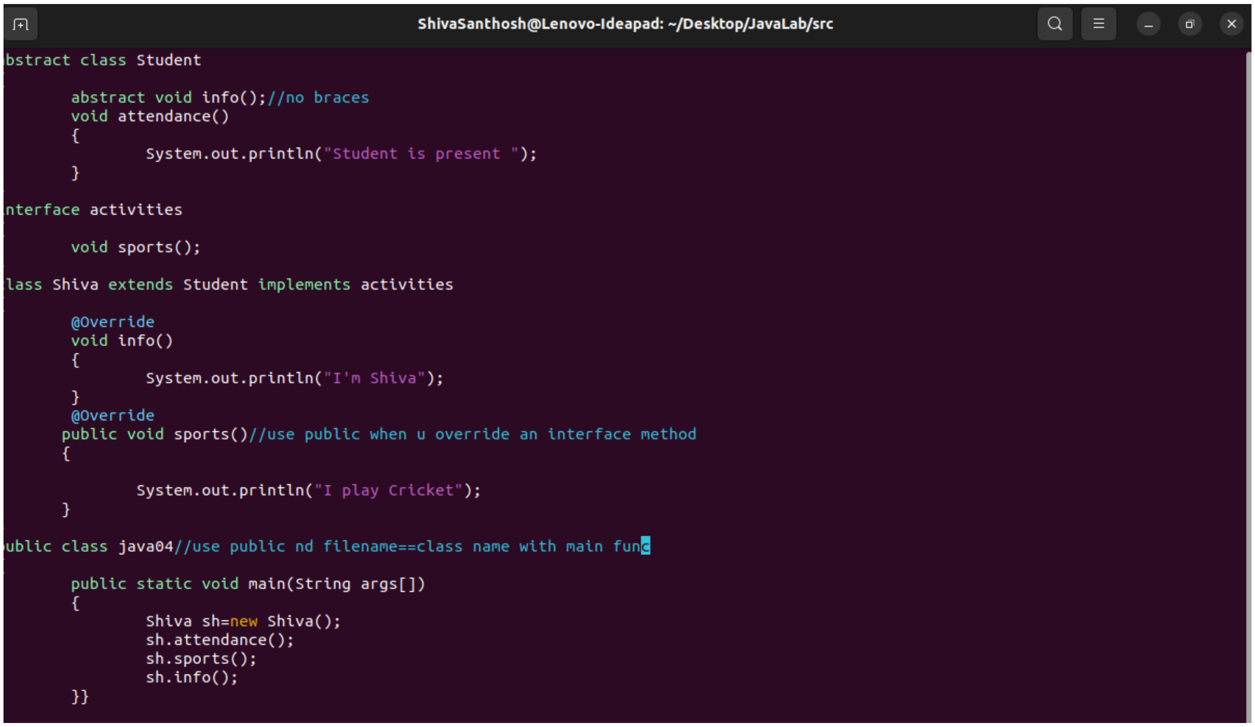


```
import mypackage.*;
public class java03contd
{
        public static void main(String args[])
        {
                java03 obj=new java03();
                obj.printing();
        }}
```



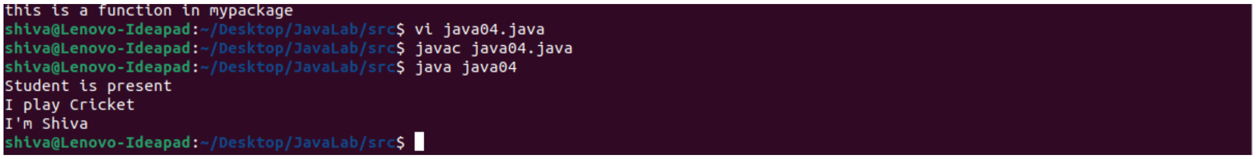
```
1 error
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java03contd.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java03contd.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java03contd
this is a function in mypackage
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

A program to illustrate the usage of interfaces and Abstract class.



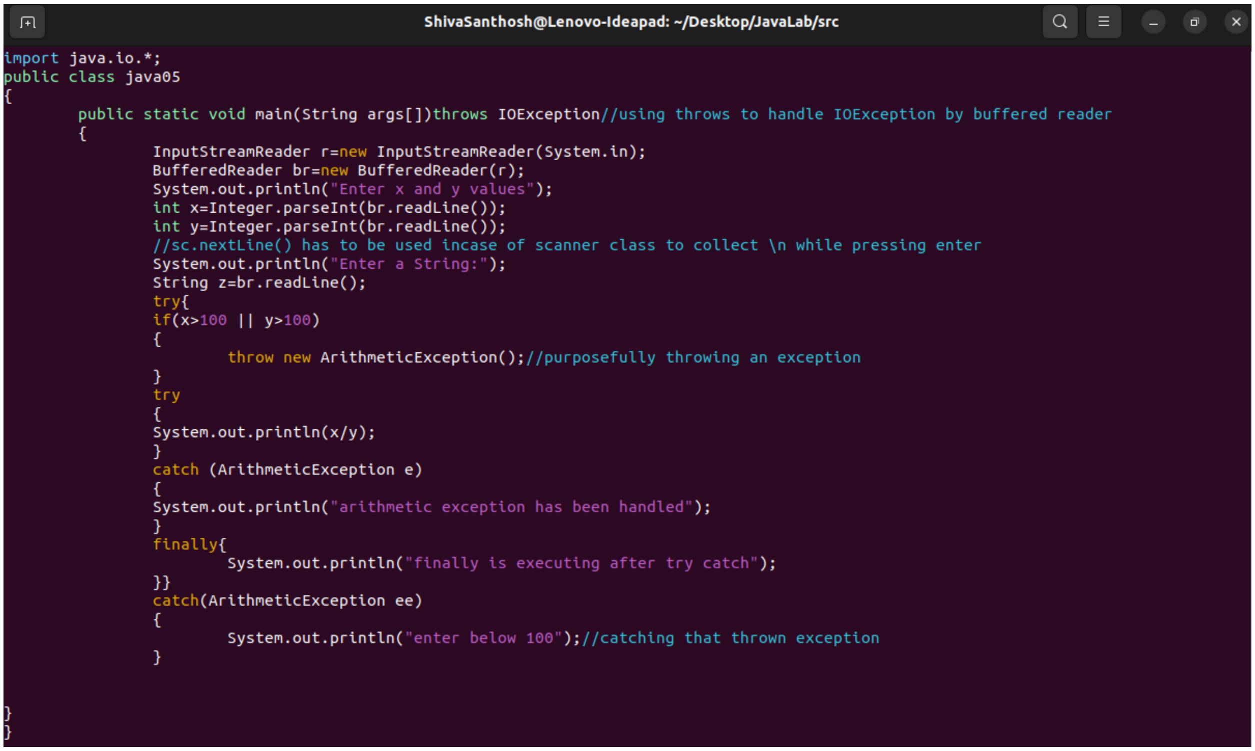
ShivaSanthosh@Lenovo-Ideapad: ~/Desktop/JavaLab/src

```
bstract class Student

       abstract void info();//no braces
       void attendance()
       {
               System.out.println("Student is present ");
       }

nterface activities

       void sports();

lass Shiva extends Student implements activities

       @Override
       void info()
       {
               System.out.println("I'm Shiva");
       }
       @Override
      public void sports()//use public when u override an interface method
      {

              System.out.println("I play Cricket");
      }

ublic class java04//use public nd filename==class name with main func

       public static void main(String args[])
       {
               Shiva sh=new Shiva();
               sh.attendance();
               sh.sports();
               sh.info();
       }}
```

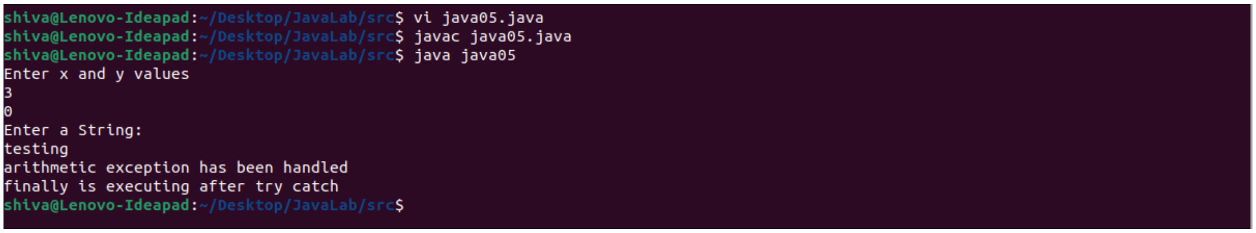


```
this is a function in mypackage
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java04.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java04.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java04
Student is present
I play Cricket
I'm Shiva
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

A program to illustrate exception handling keywords.

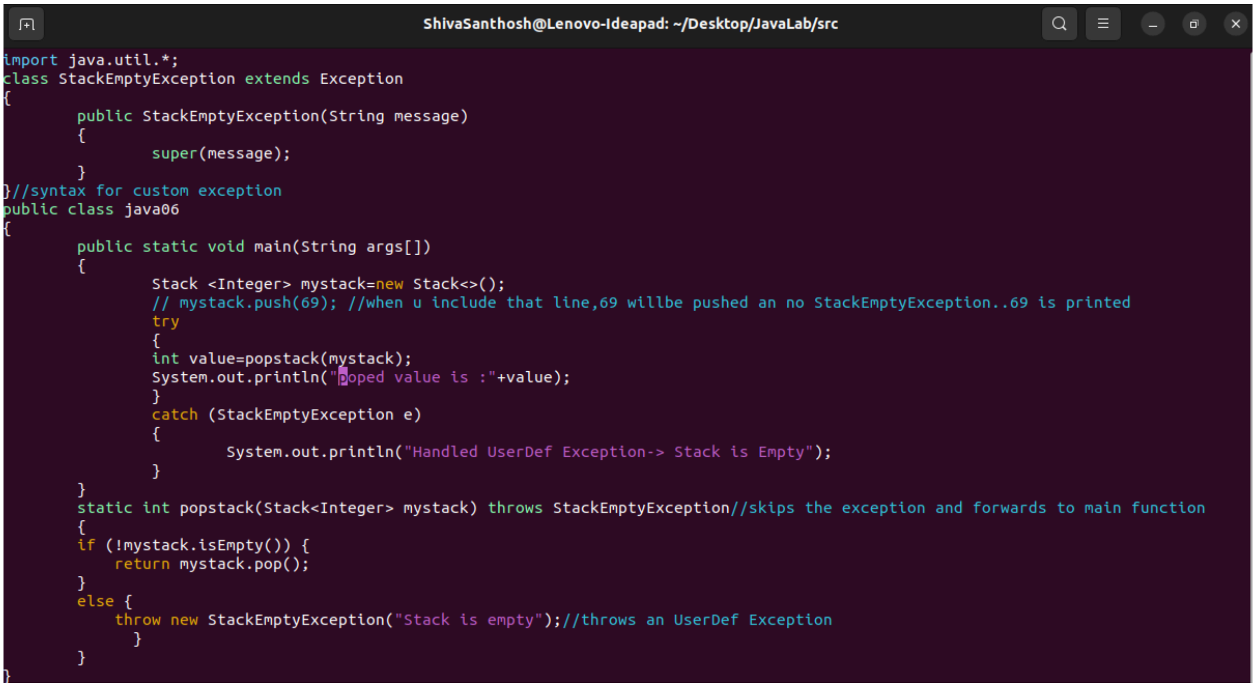


```
import java.io.*;
public class java05
{
        public static void main(String args[])throws IOException//using throws to handle IOException by buffered reader
        {
                InputStreamReader r=new InputStreamReader(System.in);
                BufferedReader br=new BufferedReader(r);
                System.out.println("Enter x and y values");
                int x=Integer.parseInt(br.readLine());
                int y=Integer.parseInt(br.readLine());
                //sc.nextLine() has to be used incase of scanner class to collect \n while pressing enter
                System.out.println("Enter a String:");
                String z=br.readLine();
                try{
                if(x>100 || y>100)
                {
                        throw new ArithmeticException();//purposefully throwing an exception
                }
                try
                {
                System.out.println(x/y);
                }
                catch (ArithmeticException e)
                {
                System.out.println("arithmetic exception has been handled");
                }
                finally{
                        System.out.println("finally is executing after try catch");
                }}
                catch(ArithmeticException ee)
                {
                        System.out.println("enter below 100");//catching that thrown exception
                }


}
}
```

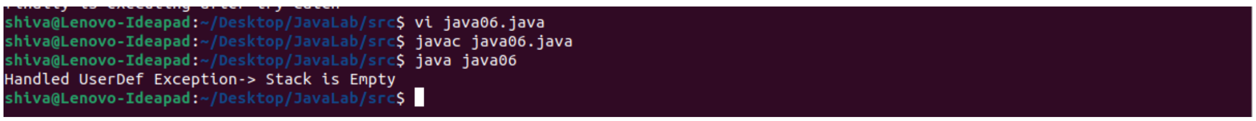


```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java05.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java05.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java05
Enter x and y values
3
0
Enter a String:
testing
arithmetic exception has been handled
finally is executing after try catch
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

A program to illustrate user define exception using stack.

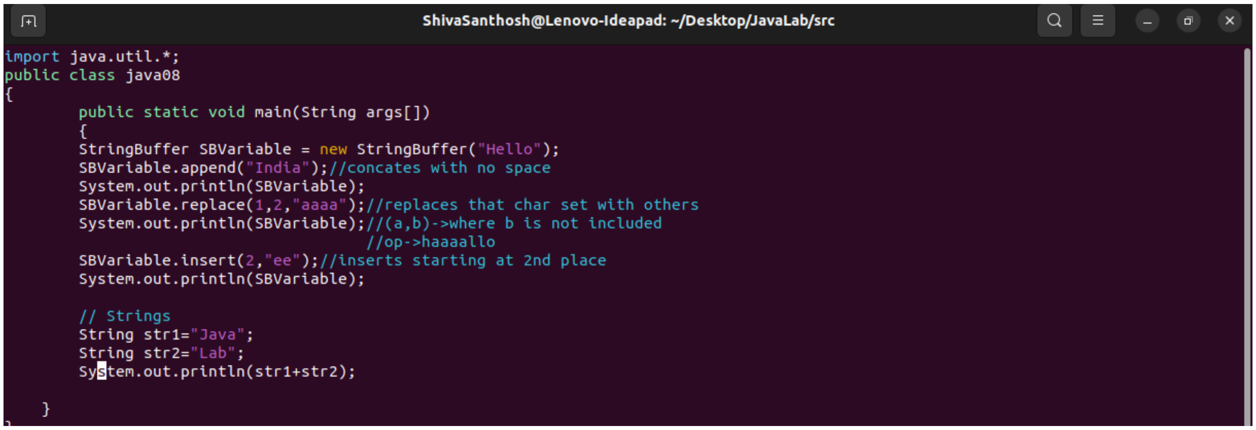


```
import java.util.*;
class StackEmptyException extends Exception
{
        public StackEmptyException(String message)
        {
                super(message);
        }
}//syntax for custom exception
public class java06
{
        public static void main(String args[])
        {
                Stack <Integer> mystack=new Stack<>();
                // mystack.push(69); //when u include that line,69 willbe pushed an no StackEmptyException..69 is printed
                try
                {
                int value=popstack(mystack);
                System.out.println("poped value is :"+value);
                }
                catch (StackEmptyException e)
                {
                        System.out.println("Handled UserDef Exception-> Stack is Empty");
                }
        }
        static int popstack(Stack<Integer> mystack) throws StackEmptyException//skips the exception and forwards to main function
        {
        if (!mystack.isEmpty()) {
            return mystack.pop();
        }
        else {
            throw new StackEmptyException("Stack is empty");//throws an UserDef Exception
              }
        }
}
```

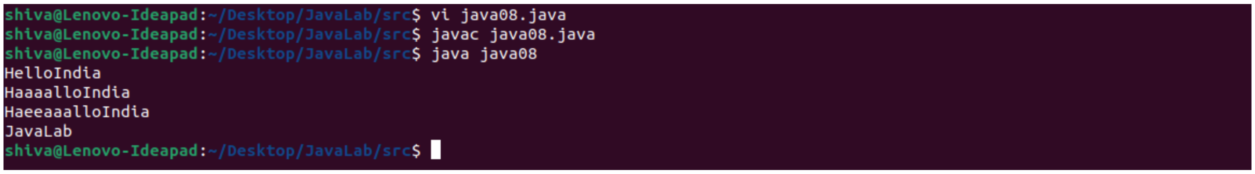


```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java06.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java06.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java06
Handled UserDef Exception-> Stack is Empty
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

A program to illustrate to handle string in java using String and StringBuffer.

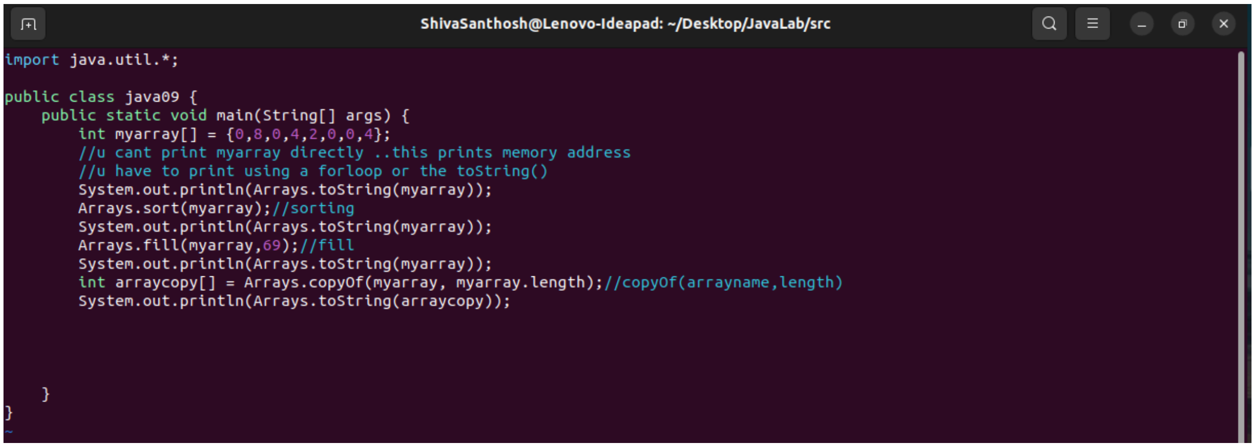


```
import java.util.*;
public class java08
{
        public static void main(String args[])
        {
        StringBuffer SBVariable = new StringBuffer("Hello");
        SBVariable.append("India");//concates with no space
        System.out.println(SBVariable);
        SBVariable.replace(1,2,"aaaa");//replaces that char set with others
        System.out.println(SBVariable);//(a,b)->where b is not included
                                       //op->haaaallo
        SBVariable.insert(2,"ee");//inserts starting at 2nd place
        System.out.println(SBVariable);

        // Strings
        String str1="Java";
        String str2="Lab";
        System.out.println(str1+str2);

    }
}
```

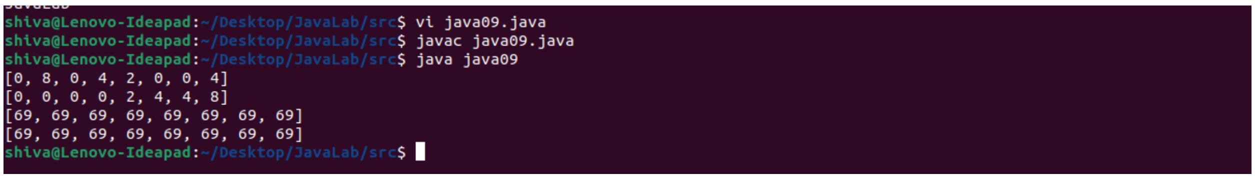


```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java08.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java08.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java08
HelloIndia
HaaaalloIndia
HaeeaaalloIndia
JavaLab
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ 
```

A program to illustrate manipulating array in java

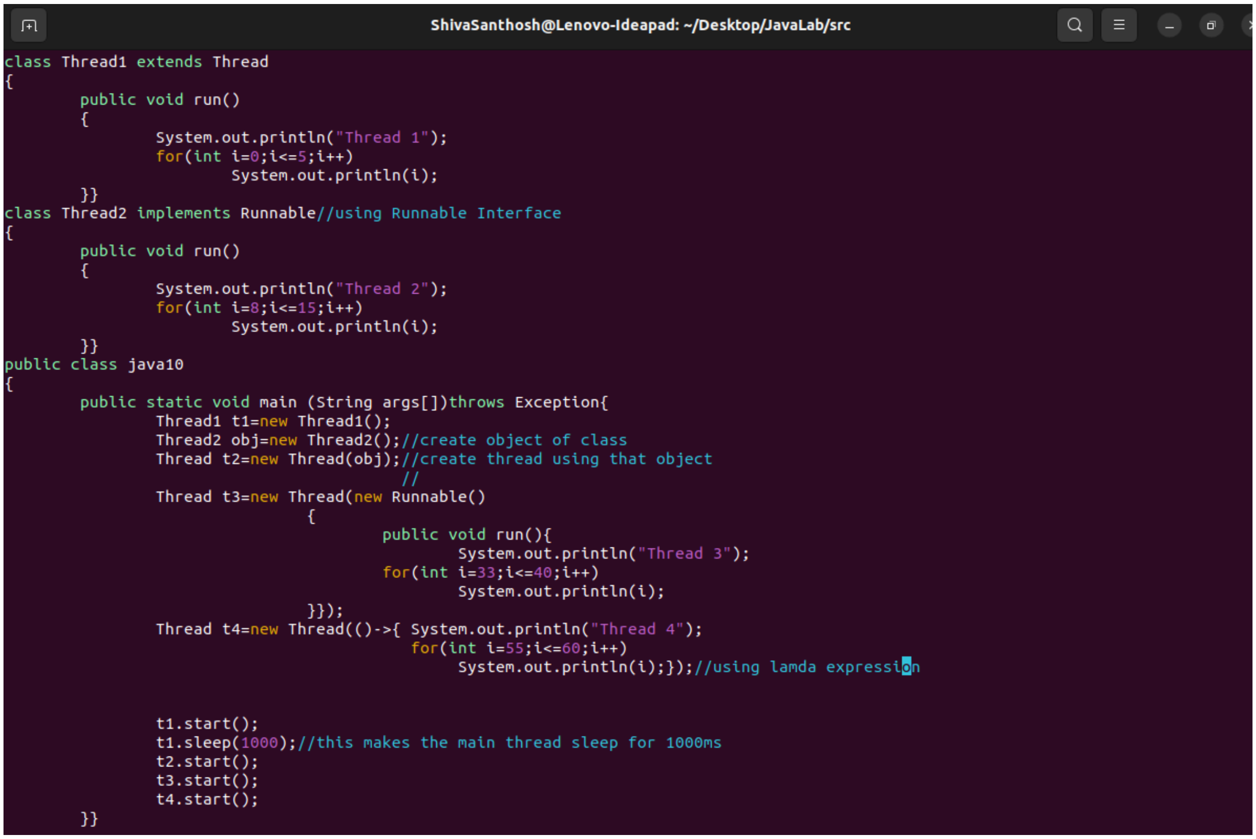


```
import java.util.*;

public class java09 {
    public static void main(String[] args) {
        int myarray[] = {0,8,0,4,2,0,0,4};
        //u cant print myarray directly ..this prints memory address
        //u have to print using a forloop or the toString()
        System.out.println(Arrays.toString(myarray));
        Arrays.sort(myarray);//sorting
        System.out.println(Arrays.toString(myarray));
        Arrays.fill(myarray,69);//fill
        System.out.println(Arrays.toString(myarray));
        int arraycopy[] = Arrays.copyOf(myarray, myarray.length);//copyOf(arrayname,length)
        System.out.println(Arrays.toString(arraycopy));



    }
}
~
```

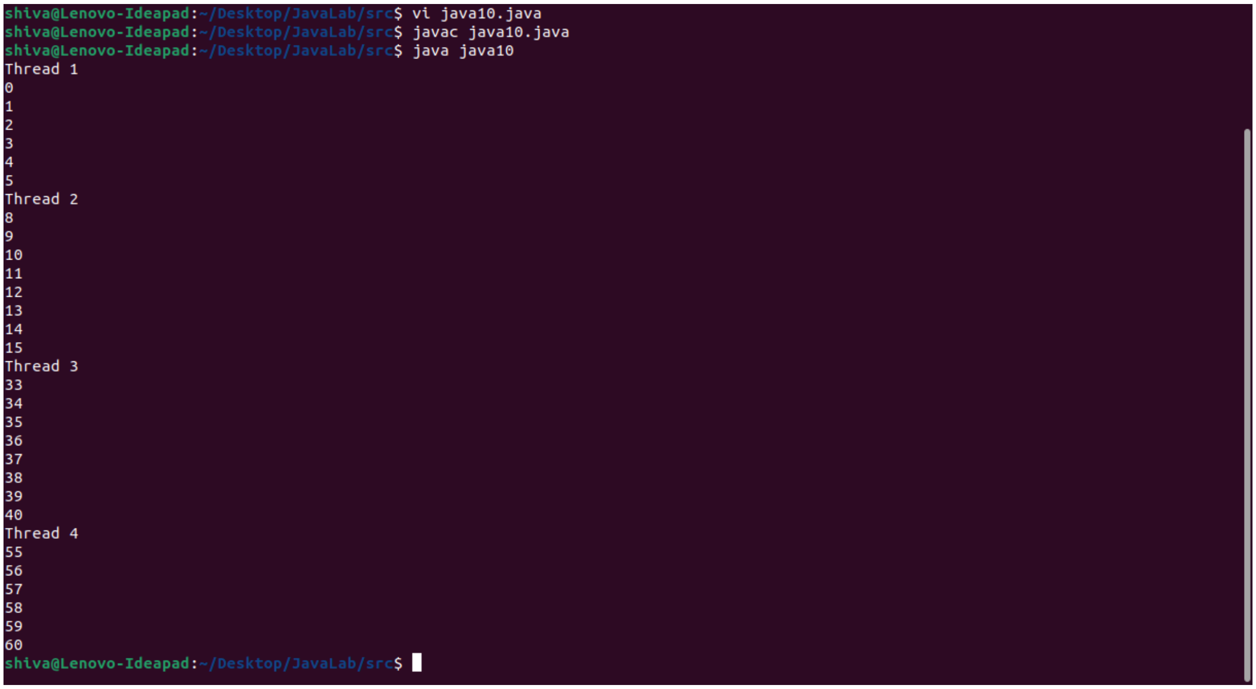


```
JavaLab
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java09.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java09.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java09
[0, 8, 0, 4, 2, 0, 0, 4]
[0, 0, 0, 0, 2, 4, 4, 8]
[69, 69, 69, 69, 69, 69, 69, 69]
[69, 69, 69, 69, 69, 69, 69, 69]
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ 
```

A program to illustrate Multithreading.

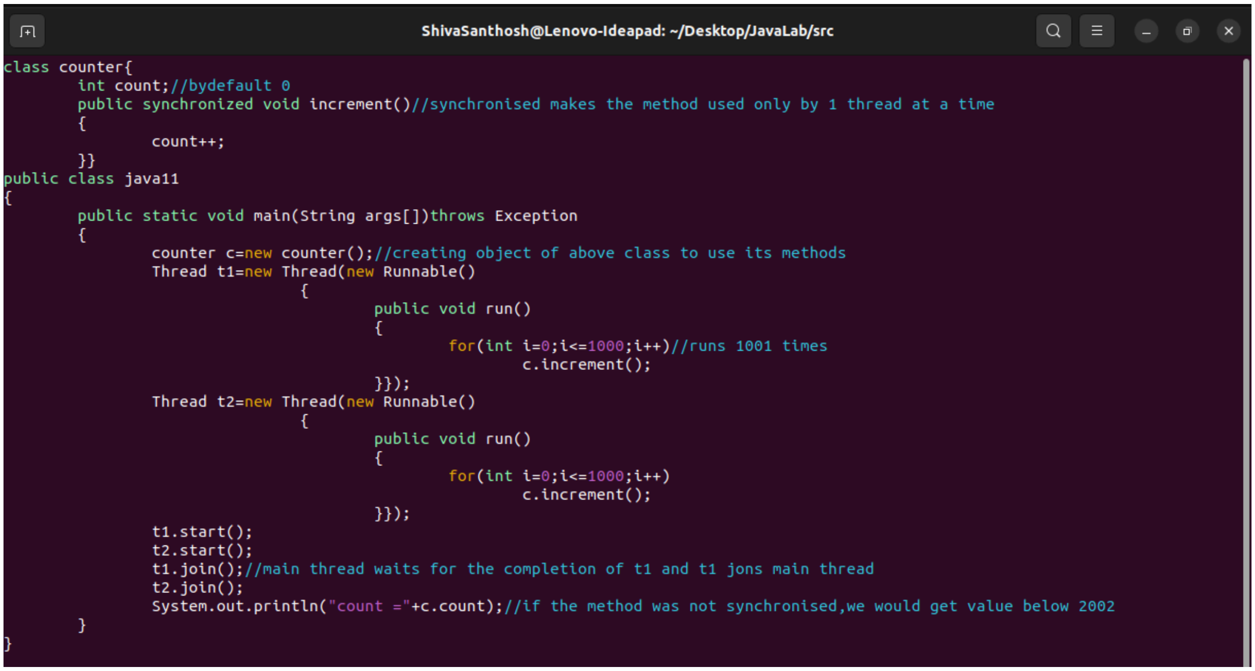


```
class Thread1 extends Thread
{
        public void run()
        {
                System.out.println("Thread 1");
                for(int i=0;i<=5;i++)
                        System.out.println(i);
        }}
class Thread2 implements Runnable//using Runnable Interface
{
        public void run()
        {
                System.out.println("Thread 2");
                for(int i=8;i<=15;i++)
                        System.out.println(i);
        }}
public class java10
{
        public static void main (String args[])throws Exception{
                Thread1 t1=new Thread1();
                Thread2 obj=new Thread2();//create object of class
                Thread t2=new Thread(obj);//create thread using that object
                                          //
                Thread t3=new Thread(new Runnable()
                                {
                                        public void run(){
                                                System.out.println("Thread 3");
                                        for(int i=33;i<=40;i++)
                                                System.out.println(i);
                                }});
                Thread t4=new Thread(()->{ System.out.println("Thread 4");
                                           for(int i=55;i<=60;i++)
                                                System.out.println(i);});//using lamda expression


                t1.start();
                t1.sleep(1000);//this makes the main thread sleep for 1000ms
                t2.start();
                t3.start();
                t4.start();
        }}
```

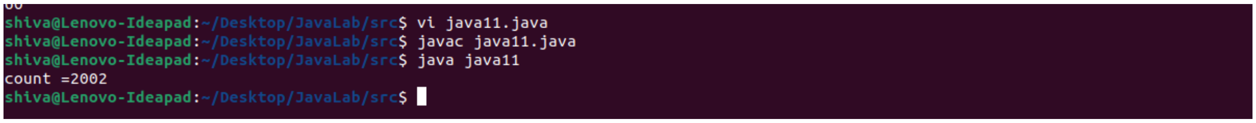


```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java10.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java10.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java10
Thread 1
0
1
2
3
4
5
Thread 2
8
9
10
11
12
13
14
15
Thread 3
33
34
35
36
37
38
39
40
Thread 4
55
56
57
58
59
60
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

A program to illustrate Thread synchronization.

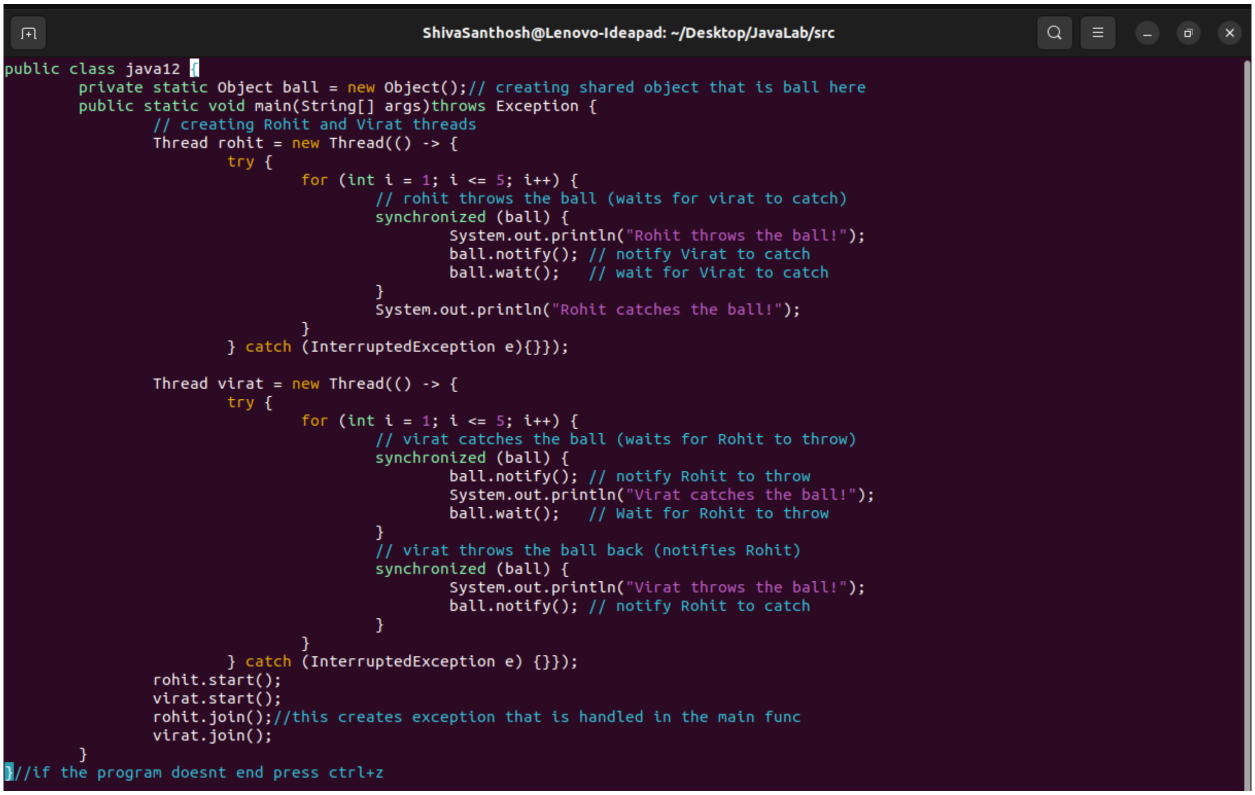


```
class counter{
        int count;//bydefault 0
        public synchronized void increment()//synchronised makes the method used only by 1 thread at a time
        {
                count++;
        }}
public class java11
{
        public static void main(String args[])throws Exception
        {
                counter c=new counter();//creating object of above class to use its methods
                Thread t1=new Thread(new Runnable()
                                {
                                        public void run()
                                        {
                                                for(int i=0;i<=1000;i++)//runs 1001 times
                                                        c.increment();
                                        }});
                Thread t2=new Thread(new Runnable()
                                {
                                        public void run()
                                        {
                                                for(int i=0;i<=1000;i++)
                                                        c.increment();
                                        }});
                t1.start();
                t2.start();
                t1.join();//main thread waits for the completion of t1 and t1 jons main thread
                t2.join();
                System.out.println("count ="+c.count);//if the method was not synchronised,we would get value below 2002
        }
}
```

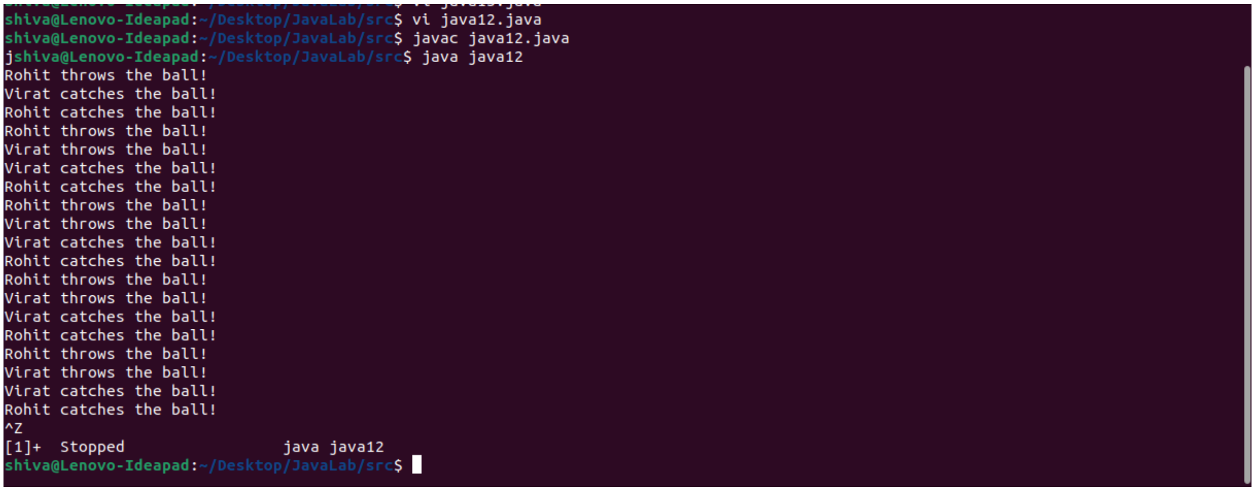


```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java11.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java11.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java11
count =2002
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ 
```

A program to illustrate inter thread communication.

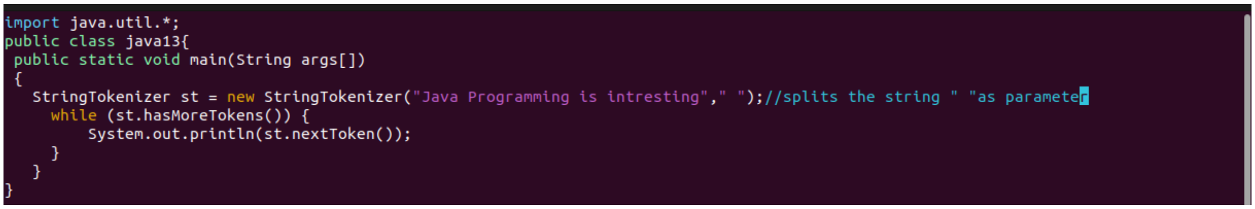


```
public class java12 {
        private static Object ball = new Object();// creating shared object that is ball here
        public static void main(String[] args)throws Exception {
                // creating Rohit and Virat threads
                Thread rohit = new Thread(() -> {
                        try {
                                for (int i = 1; i <= 5; i++) {
                                        // rohit throws the ball (waits for virat to catch)
                                        synchronized (ball) {
                                                System.out.println("Rohit throws the ball!");
                                                ball.notify(); // notify Virat to catch
                                                ball.wait();   // wait for Virat to catch
                                        }
                                        System.out.println("Rohit catches the ball!");
                                }
                        } catch (InterruptedException e){}});

                Thread virat = new Thread(() -> {
                        try {
                                for (int i = 1; i <= 5; i++) {
                                        // virat catches the ball (waits for Rohit to throw)
                                        synchronized (ball) {
                                                ball.notify(); // notify Rohit to throw
                                                System.out.println("Virat catches the ball!");
                                                ball.wait();   // Wait for Rohit to throw
                                        }
                                        // virat throws the ball back (notifies Rohit)
                                        synchronized (ball) {
                                                System.out.println("Virat throws the ball!");
                                                ball.notify(); // notify Rohit to catch
                                        }
                                }
                        } catch (InterruptedException e) {}});
                rohit.start();
                virat.start();
                rohit.join();//this creates exception that is handled in the main func
                virat.join();
        }
}//if the program doesnt end press ctrl+z
```

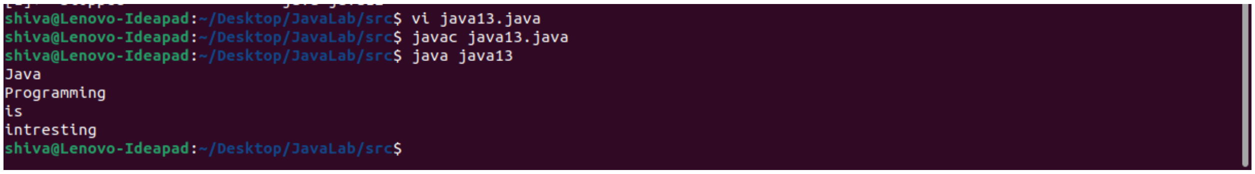


```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java12.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java12.java
jshiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java12
Rohit throws the ball!
Virat catches the ball!
Rohit catches the ball!
Rohit throws the ball!
Virat throws the ball!
Virat catches the ball!
Rohit catches the ball!
Rohit throws the ball!
Virat throws the ball!
Virat catches the ball!
Rohit catches the ball!
Rohit throws the ball!
Virat throws the ball!
Virat catches the ball!
Rohit catches the ball!
Rohit throws the ball!
Virat throws the ball!
Virat catches the ball!
Rohit catches the ball!
^Z
[1]+  Stopped                 java java12
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ 
```

A program using String tokenizer.

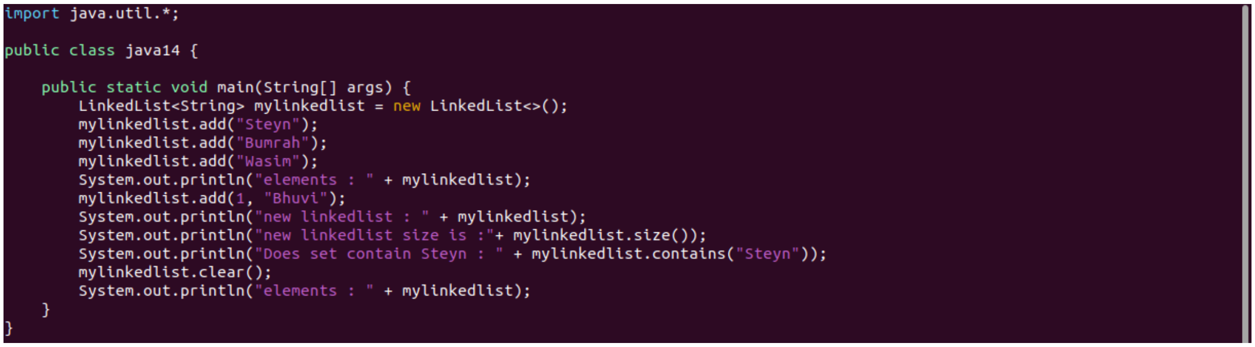


```
import java.util.*;
public class java13{
 public static void main(String args[])
 {
   StringTokenizer st = new StringTokenizer("Java Programming is intresting"," ");//splits the string " "as parameter
     while (st.hasMoreTokens()) {
         System.out.println(st.nextToken());
     }
   }
}
```

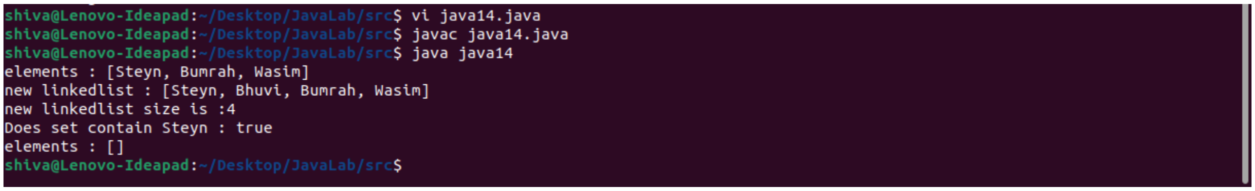


```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java13.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java13.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java13
Java
Programming
is
intresting
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

A program using Linked list class.

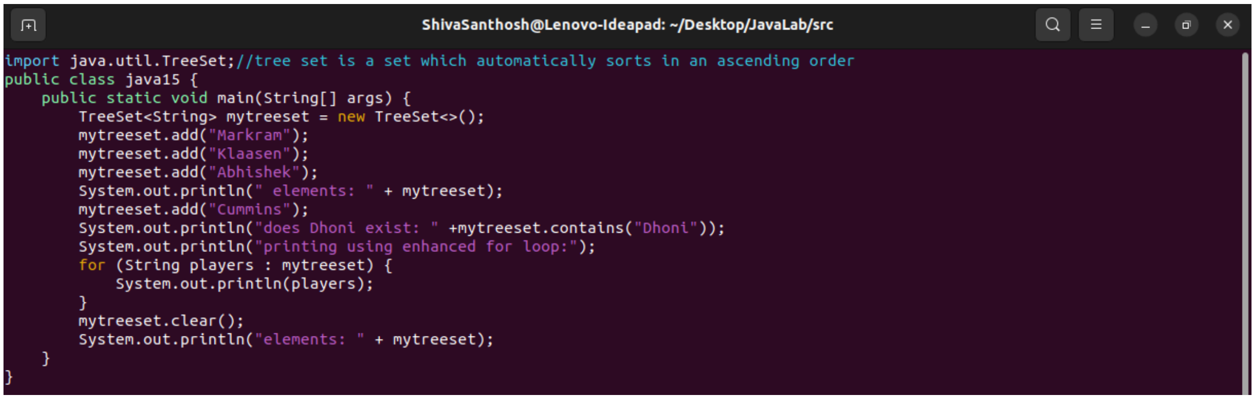


```
import java.util.*;

public class java14 {

    public static void main(String[] args) {
        LinkedList<String> mylinkedlist = new LinkedList<>();
        mylinkedlist.add("Steyn");
        mylinkedlist.add("Bumrah");
        mylinkedlist.add("Wasim");
        System.out.println("elements : " + mylinkedlist);
        mylinkedlist.add(1, "Bhuvi");
        System.out.println("new linkedlist : " + mylinkedlist);
        System.out.println("new linkedlist size is :"+ mylinkedlist.size());
        System.out.println("Does set contain Steyn : " + mylinkedlist.contains("Steyn"));
        mylinkedlist.clear();
        System.out.println("elements : " + mylinkedlist);
    }
}
```

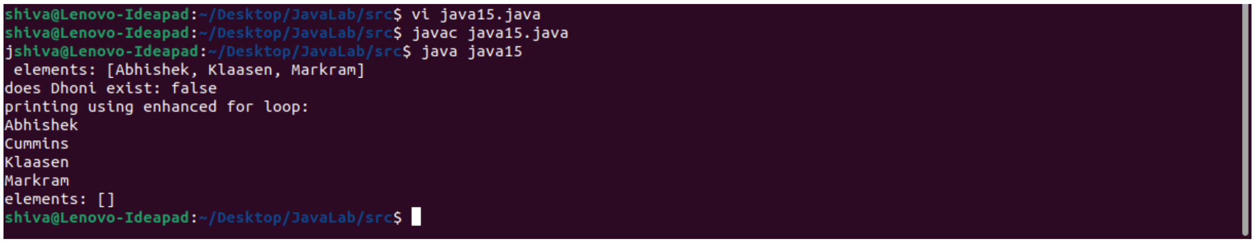


```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java14.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java14.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java14
elements : [Steyn, Bumrah, Wasim]
new linkedlist : [Steyn, Bhuvi, Bumrah, Wasim]
new linkedlist size is :4
Does set contain Steyn : true
elements : []
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

A program using Tree set class.

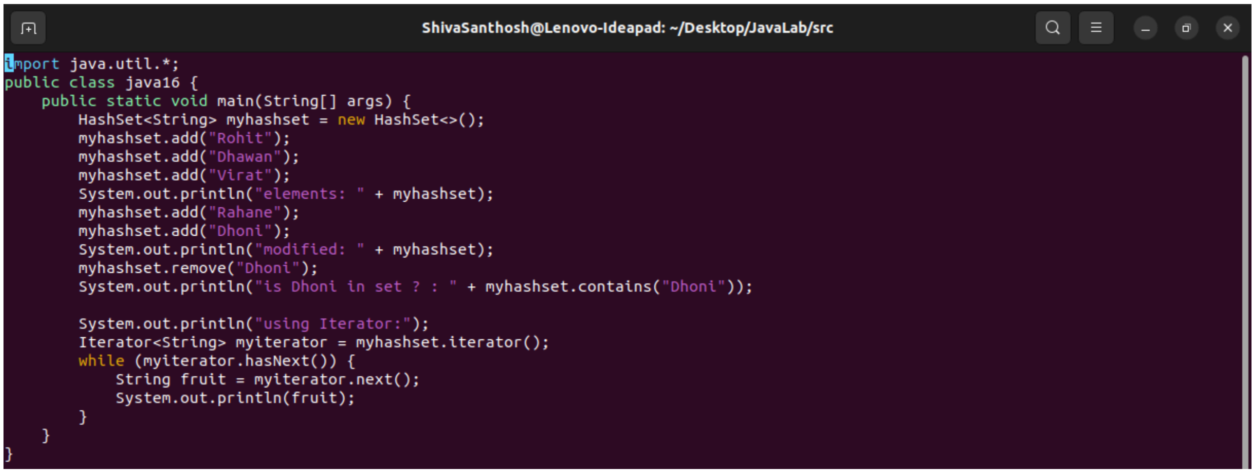


```
import java.util.TreeSet;//tree set is a set which automatically sorts in an ascending order
public class java15 {
    public static void main(String[] args) {
        TreeSet<String> mytreeset = new TreeSet<>();
        mytreeset.add("Markram");
        mytreeset.add("Klaasen");
        mytreeset.add("Abhishek");
        System.out.println(" elements: " + mytreeset);
        mytreeset.add("Cummins");
        System.out.println("does Dhoni exist: " +mytreeset.contains("Dhoni"));
        System.out.println("printing using enhanced for loop:");
        for (String players : mytreeset) {
            System.out.println(players);
        }
        mytreeset.clear();
        System.out.println("elements: " + mytreeset);
    }
}
```

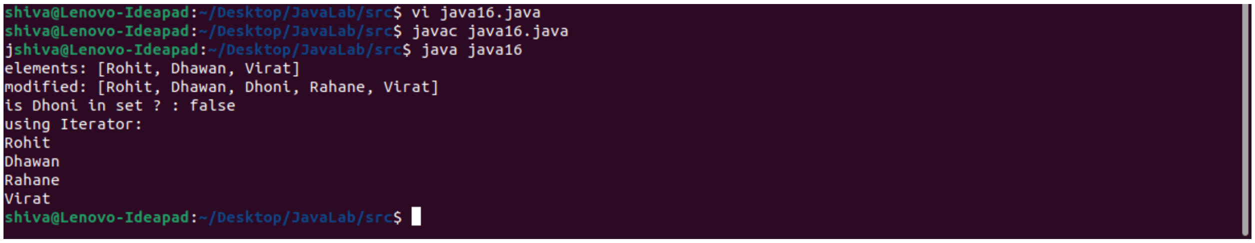


```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java15.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java15.java
jshiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java15
 elements: [Abhishek, Klaasen, Markram]
does Dhoni exist: false
printing using enhanced for loop:
Abhishek
Cummins
Klaasen
Markram
elements: []
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

A program using Hash set and Iterator classes.

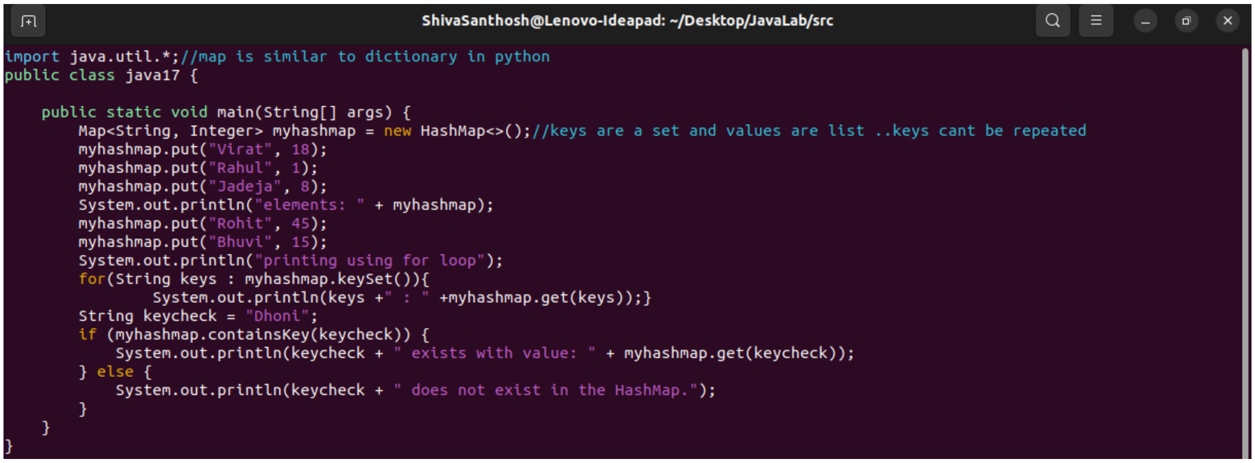


```
import java.util.*;
public class java16 {
    public static void main(String[] args) {
        HashSet<String> myhashset = new HashSet<>();
        myhashset.add("Rohit");
        myhashset.add("Dhawan");
        myhashset.add("Virat");
        System.out.println("elements: " + myhashset);
        myhashset.add("Rahane");
        myhashset.add("Dhoni");
        System.out.println("modified: " + myhashset);
        myhashset.remove("Dhoni");
        System.out.println("is Dhoni in set ? : " + myhashset.contains("Dhoni"));

        System.out.println("using Iterator:");
        Iterator<String> myiterator = myhashset.iterator();
        while (myiterator.hasNext()) {
            String fruit = myiterator.next();
            System.out.println(fruit);
        }
    }
}
```

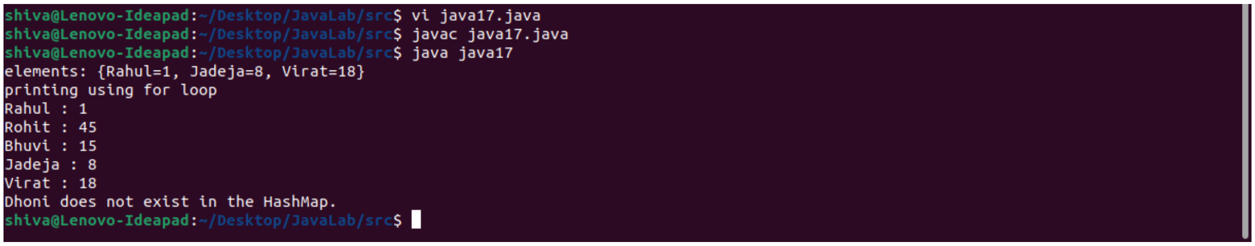


```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java16.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java16.java
jshiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java16
elements: [Rohit, Dhawan, Virat]
modified: [Rohit, Dhawan, Dhoni, Rahane, Virat]
is Dhoni in set ? : false
using Iterator:
Rohit
Dhawan
Rahane
Virat
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

A program using Map classes.

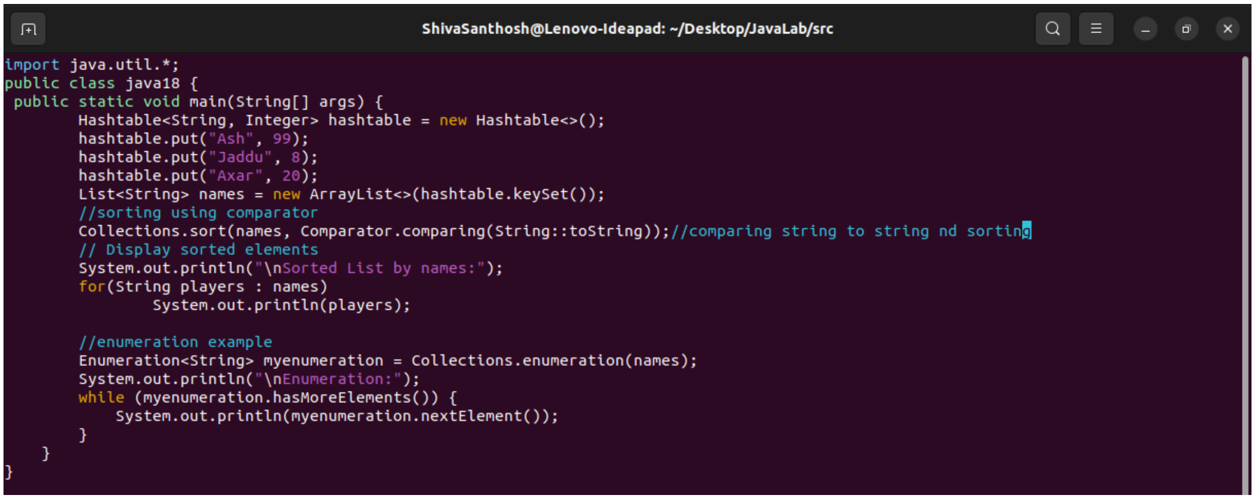


```
import java.util.*;//map is similar to dictionary in python
public class java17 {

    public static void main(String[] args) {
        Map<String, Integer> myhashmap = new HashMap<>();//keys are a set and values are list ..keys cant be repeated
        myhashmap.put("Virat", 18);
        myhashmap.put("Rahul", 1);
        myhashmap.put("Jadeja", 8);
        System.out.println("elements: " + myhashmap);
        myhashmap.put("Rohit", 45);
        myhashmap.put("Bhuvi", 15);
        System.out.println("printing using for loop");
        for(String keys : myhashmap.keySet()){
                System.out.println(keys +" : " +myhashmap.get(keys));}
        String keycheck = "Dhoni";
        if (myhashmap.containsKey(keycheck)) {
            System.out.println(keycheck + " exists with value: " + myhashmap.get(keycheck));
        } else {
            System.out.println(keycheck + " does not exist in the HashMap.");
        }
    }
}
```

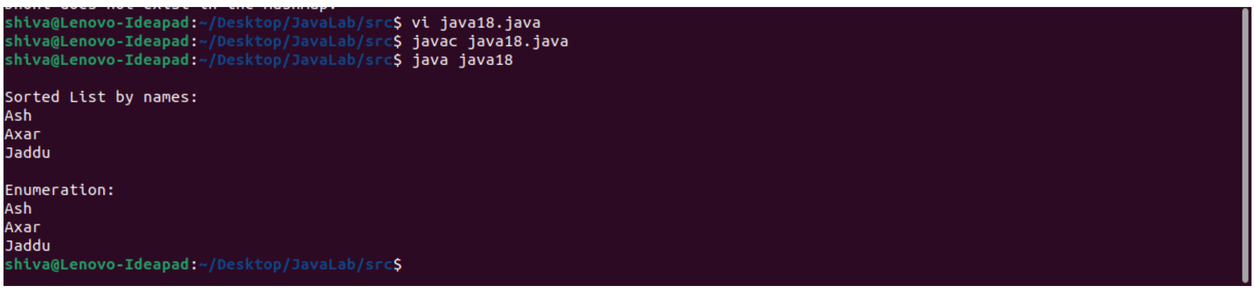


```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java17.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java17.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java17
elements: {Rahul=1, Jadeja=8, Virat=18}
printing using for loop
Rahul : 1
Rohit : 45
Bhuvi : 15
Jadeja : 8
Virat : 18
Dhoni does not exist in the HashMap.
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

A program using Enumeration and Comparator interfaces.

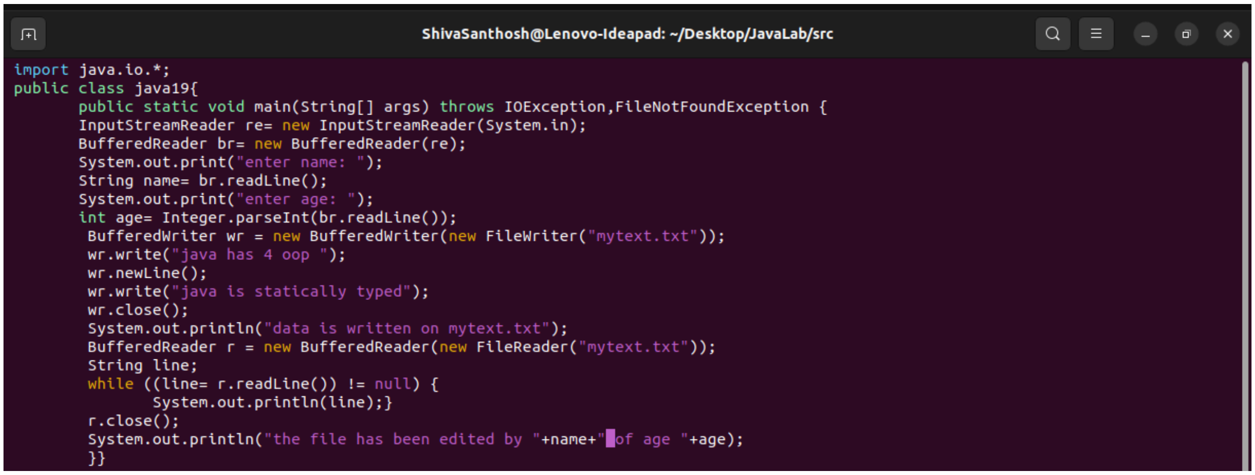


```
import java.util.*;
public class java18 {
 public static void main(String[] args) {
        Hashtable<String, Integer> hashtable = new Hashtable<>();
        hashtable.put("Ash", 99);
        hashtable.put("Jaddu", 8);
        hashtable.put("Axar", 20);
        List<String> names = new ArrayList<>(hashtable.keySet());
        //sorting using comparator
        Collections.sort(names, Comparator.comparing(String::toString));//comparing string to string nd sorting
        // Display sorted elements
        System.out.println("\nSorted List by names:");
        for(String players : names)
                System.out.println(players);

        //enumeration example
        Enumeration<String> myenumeration = Collections.enumeration(names);
        System.out.println("\nEnumeration:");
        while (myenumeration.hasMoreElements()) {
            System.out.println(myenumeration.nextElement());
        }
    }
}
```

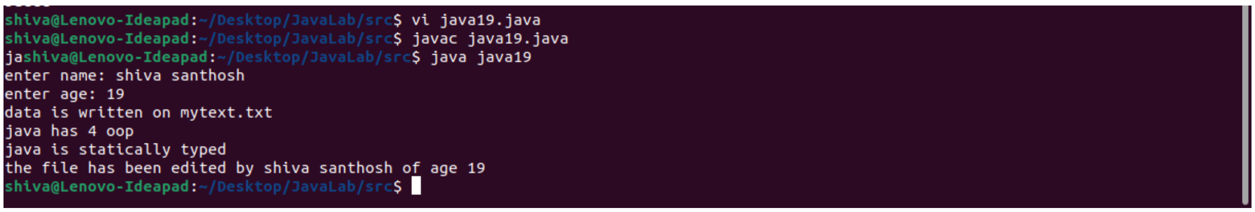


```
Dhoni does not exist in the HashMap.
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java18.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java18.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java18

Sorted List by names:
Ash
Axar
Jaddu

Enumeration:
Ash
Axar
Jaddu
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

A program to illustrate Buffered I/O streams and Buffered reader.

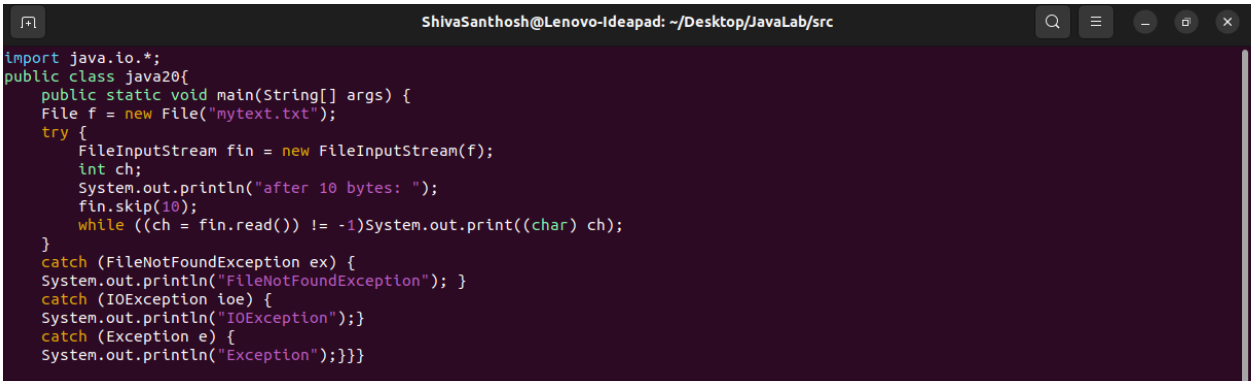


```
import java.io.*;
public class java19{
       public static void main(String[] args) throws IOException,FileNotFoundException {
       InputStreamReader re= new InputStreamReader(System.in);
       BufferedReader br= new BufferedReader(re);
       System.out.print("enter name: ");
       String name= br.readLine();
       System.out.print("enter age: ");
       int age= Integer.parseInt(br.readLine());
        BufferedWriter wr = new BufferedWriter(new FileWriter("mytext.txt"));
        wr.write("java has 4 oop ");
        wr.newLine();
        wr.write("java is statically typed");
        wr.close();
        System.out.println("data is written on mytext.txt");
        BufferedReader r = new BufferedReader(new FileReader("mytext.txt"));
        String line;
        while ((line= r.readLine()) != null) {
               System.out.println(line);}
        r.close();
        System.out.println("the file has been edited by "+name+" of age "+age);
        }}
```

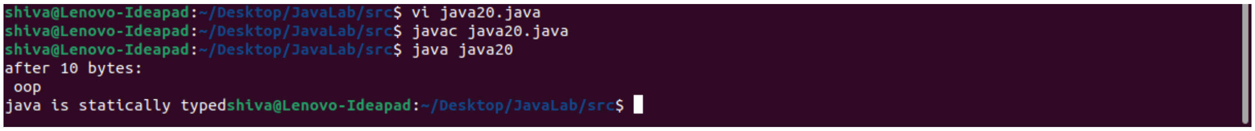


```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java19.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java19.java
jashiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java19
enter name: shiva santhosh
enter age: 19
data is written on mytext.txt
java has 4 oop
java is statically typed
the file has been edited by shiva santhosh of age 19
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

Write a Java program to read text from file from a specify index or skipping byte using file Input stream.

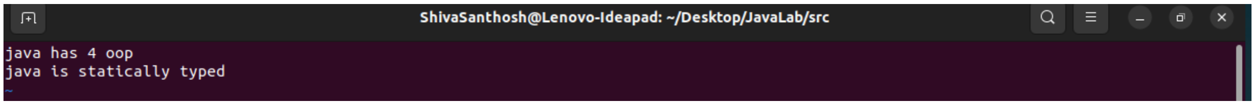


```
import java.io.*;
public class java20{
    public static void main(String[] args) {
    File f = new File("mytext.txt");
    try {
        FileInputStream fin = new FileInputStream(f);
        int ch;
        System.out.println("after 10 bytes: ");
        fin.skip(10);
        while ((ch = fin.read()) != -1)System.out.print((char) ch);
    }
    catch (FileNotFoundException ex) {
    System.out.println("FileNotFoundException"); }
    catch (IOException ioe) {
    System.out.println("IOException");}
    catch (Exception e) {
    System.out.println("Exception");}}}
```



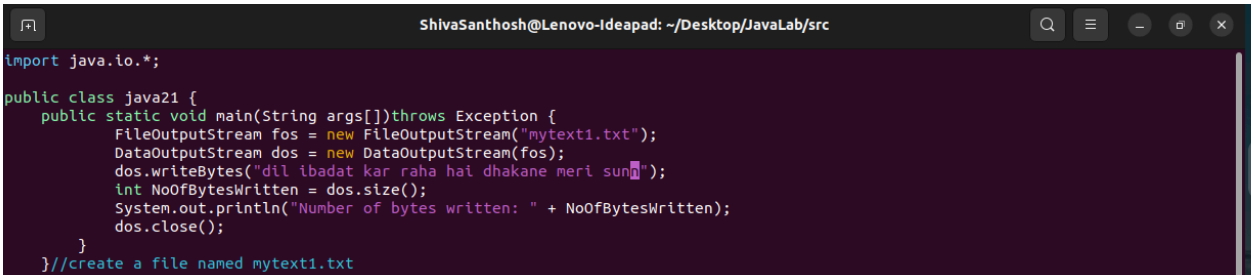
```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java20.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java20.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java20
after 10 bytes:
 oop
java is statically typedshiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

File used -> mytext.txt



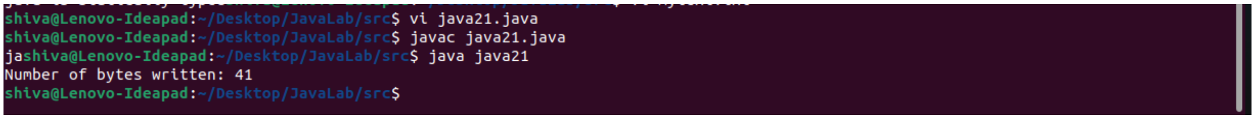
```
java has 4 oop
java is statically typed
~
```

Write a Java program to determine number of byte return to file using data output stream.



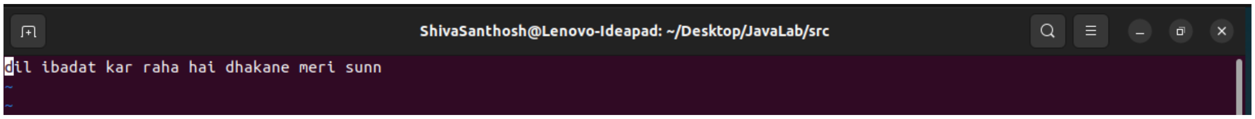
ShivaSanthosh@Lenovo-Ideapad: ~/Desktop/JavaLab/src

```
import java.io.*;

public class java21 {
    public static void main(String args[])throws Exception {
            FileOutputStream fos = new FileOutputStream("mytext1.txt");
            DataOutputStream dos = new DataOutputStream(fos);
            dos.writeBytes("dil ibadat kar raha hai dhakane meri sunn");
            int NoOfBytesWritten = dos.size();
            System.out.println("Number of bytes written: " + NoOfBytesWritten);
            dos.close();
        }
    }//create a file named mytext1.txt
```



```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java21.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java21.java
jashiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java21
Number of bytes written: 41
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

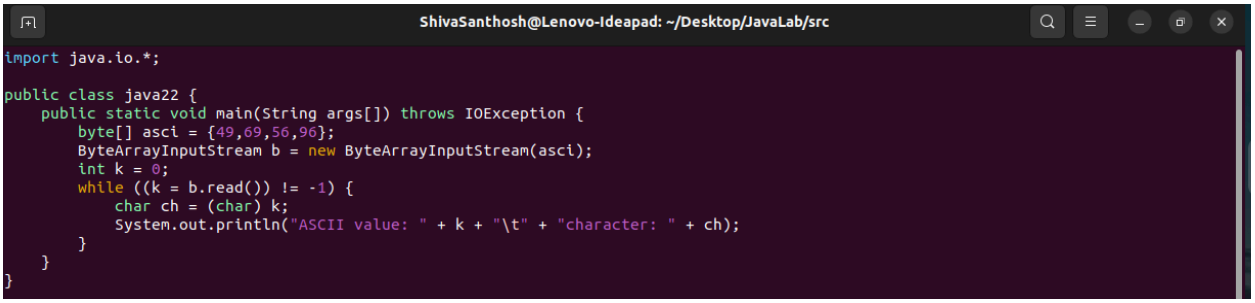
File -> mytext1.txt



ShivaSanthosh@Lenovo-Ideapad: ~/Desktop/JavaLab/src

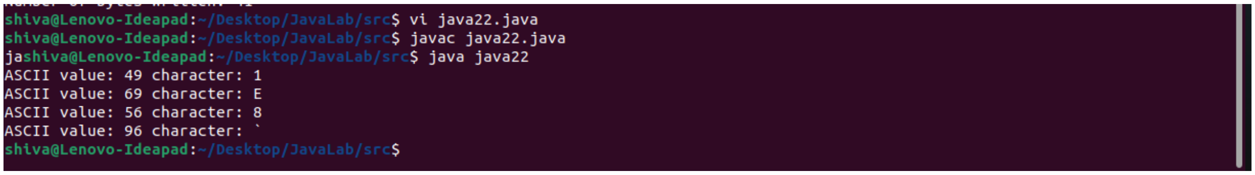
```
dil ibadat kar raha hai dhakane meri sunn
~
~
```

A program to illustrate ByteArrayI/O Streams.



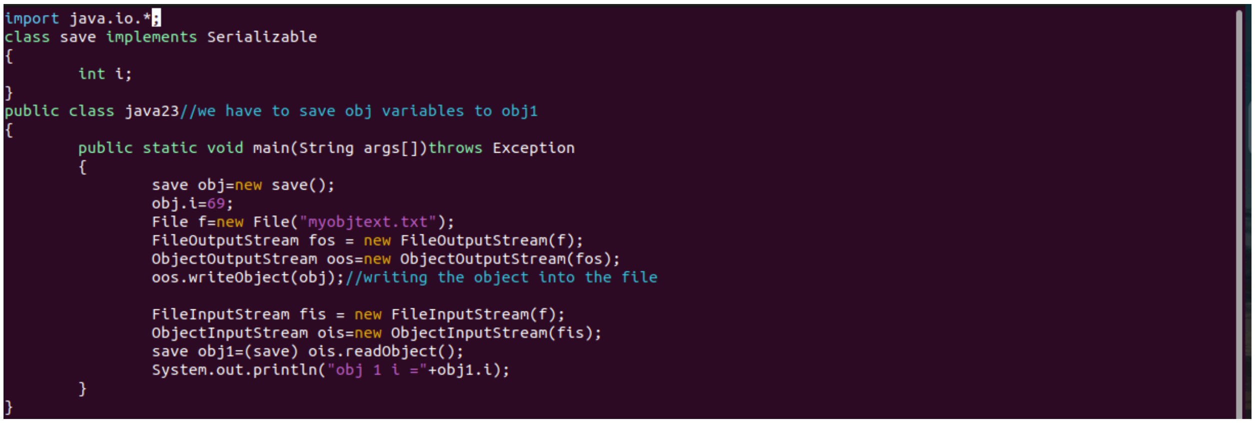
ShivaSanthosh@Lenovo-Ideapad: ~/Desktop/JavaLab/src

```
import java.io.*;

public class java22 {
    public static void main(String args[]) throws IOException {
        byte[] asci = {49,69,56,96};
        ByteArrayInputStream b = new ByteArrayInputStream(asci);
        int k = 0;
        while ((k = b.read()) != -1) {
            char ch = (char) k;
            System.out.println("ASCII value: " + k + "\t" + "character: " + ch);
        }
    }
}
```

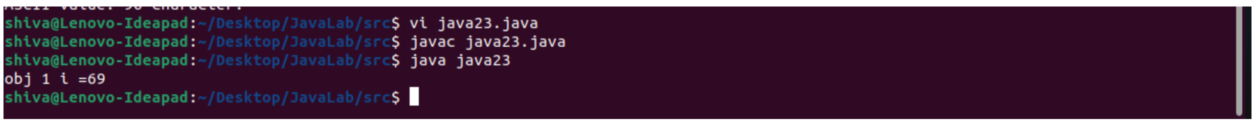


```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java22.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java22.java
jashiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java22
ASCII value: 49 character: 1
ASCII value: 69 character: E
ASCII value: 56 character: 8
ASCII value: 96 character: `
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$
```

A program to illustrate the usage of Serialization.

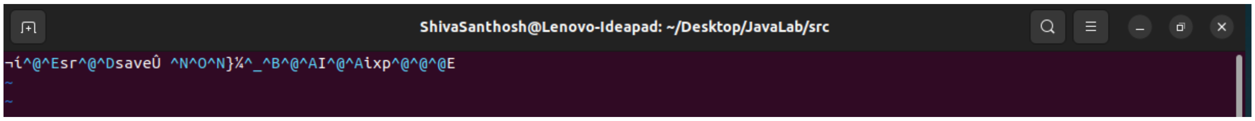


```
import java.io.*;
class save implements Serializable
{
        int i;
}
public class java23//we have to save obj variables to obj1
{
        public static void main(String args[])throws Exception
        {
                save obj=new save();
                obj.i=69;
                File f=new File("myobjtext.txt");
                FileOutputStream fos = new FileOutputStream(f);
                ObjectOutputStream oos=new ObjectOutputStream(fos);
                oos.writeObject(obj);//writing the object into the file

                FileInputStream fis = new FileInputStream(f);
                ObjectInputStream ois=new ObjectInputStream(fis);
                save obj1=(save) ois.readObject();
                System.out.println("obj 1 i ="+obj1.i);
        }
}
```



```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java23.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java23.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java23
obj 1 i =69
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ 
```

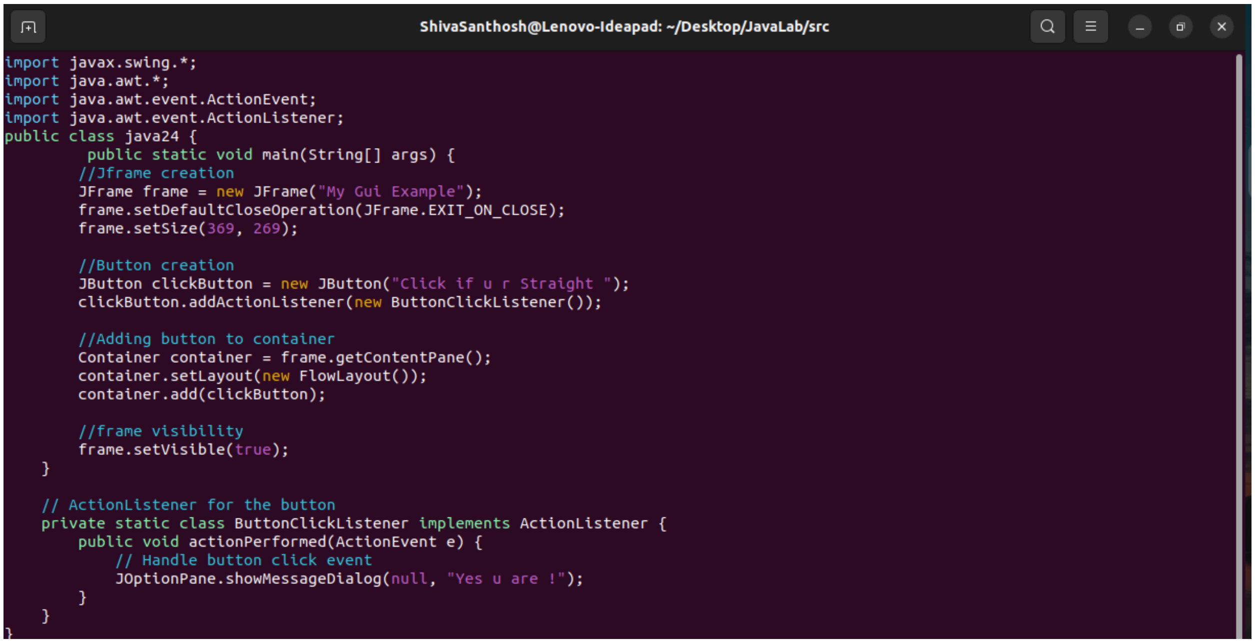
Variables of Object obj is stored in file myobjtext.txt



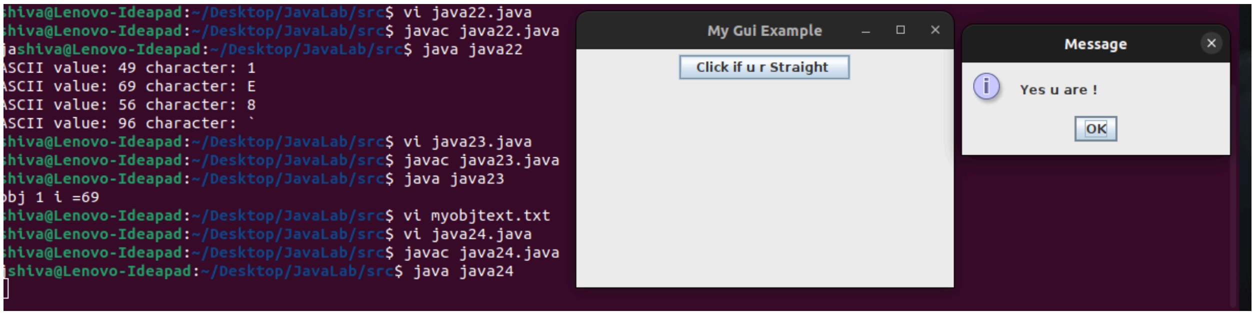
```
¬í^@^Esr^@^DsaveÛ ^N^O^N}¼^_^B^@^AI^@^Aixp^@^@^@E
~
~
```

Java can understand that text

An application involving GUI with different controls, menus and event handling.



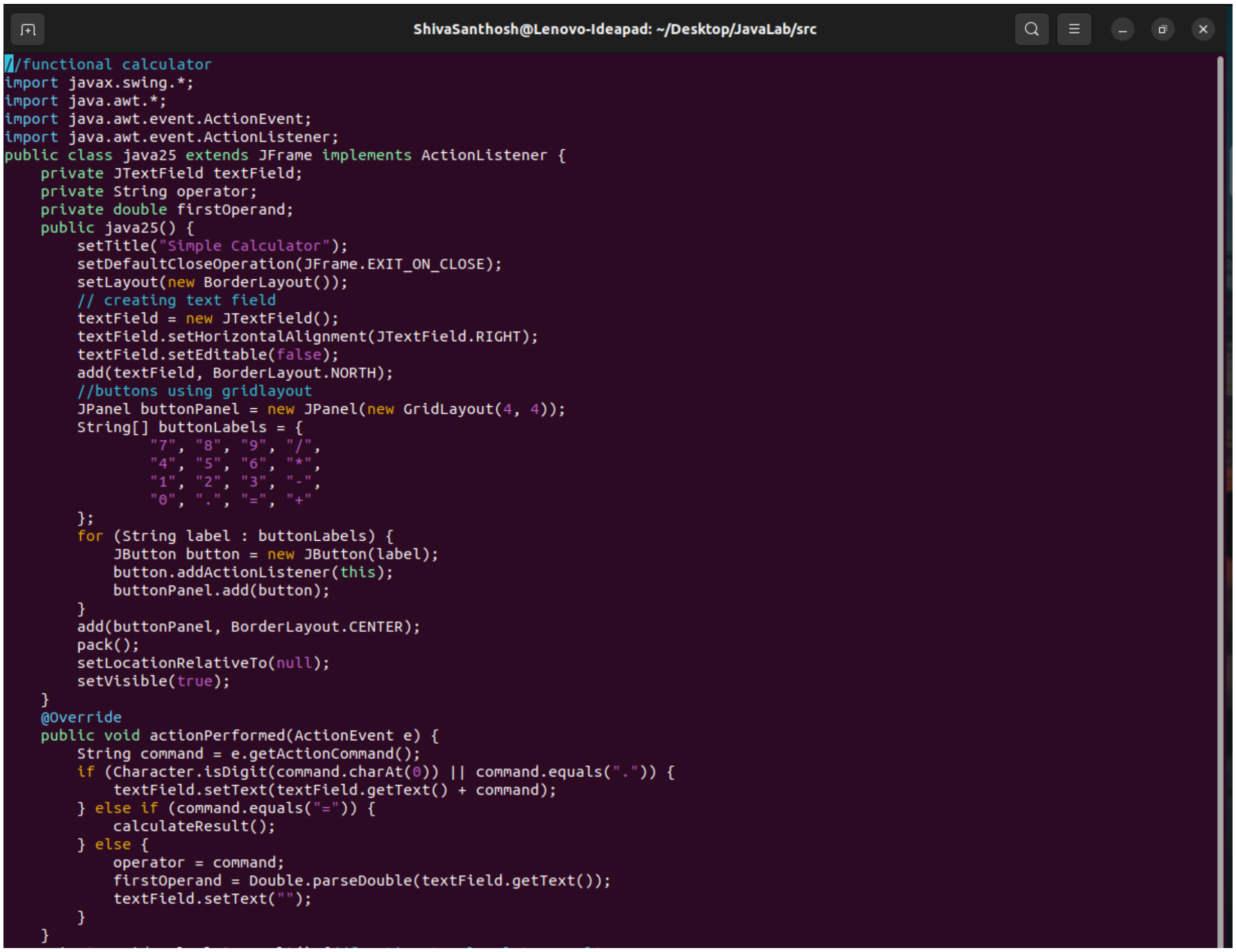
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class java24 {
         public static void main(String[] args) {
        //Jframe creation
        JFrame frame = new JFrame("My Gui Example");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(369, 269);

        //Button creation
        JButton clickButton = new JButton("Click if u r Straight ");
        clickButton.addActionListener(new ButtonClickListener());

        //Adding button to container
        Container container = frame.getContentPane();
        container.setLayout(new FlowLayout());
        container.add(clickButton);

        //frame visibility
        frame.setVisible(true);
    }

    // ActionListener for the button
    private static class ButtonClickListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            // Handle button click event
            JOptionPane.showMessageDialog(null, "Yes u are !");
        }
    }
}
```



```
hiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java22.java
hiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java22.java
jashiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java22
SCII value: 49 character: 1
SCII value: 69 character: E
SCII value: 56 character: 8
SCII value: 96 character: `
hiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java23.java
hiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java23.java
hiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java23
bj 1 i =69
hiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi myobjtext.txt
hiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java24.java
hiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java24.java
ishiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java24
```

Yes u are !

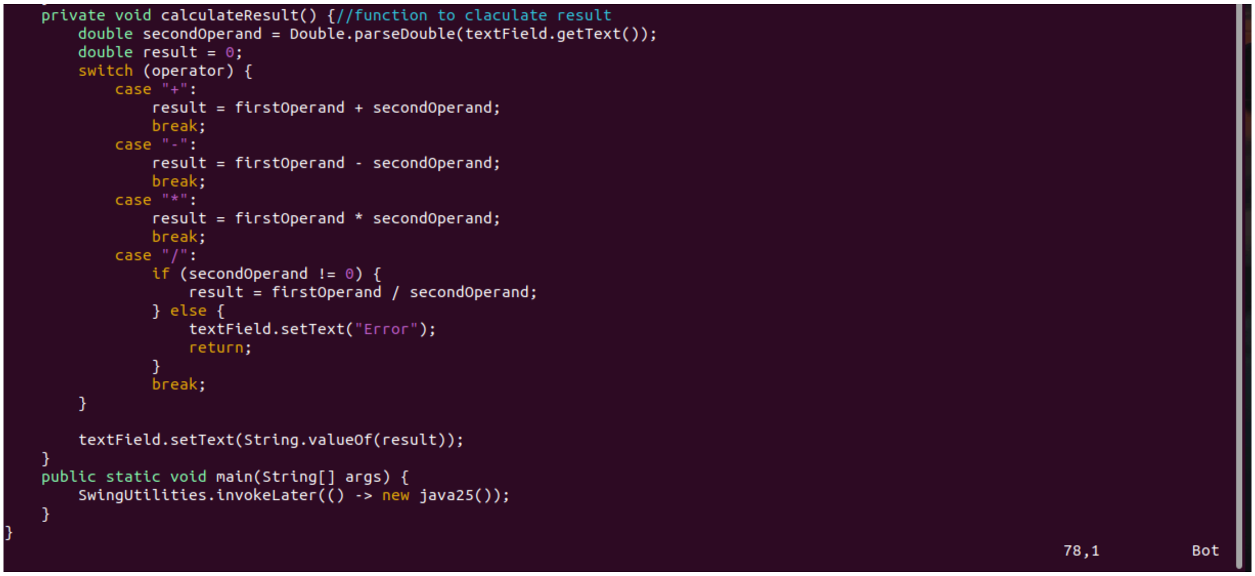
A program to implement a simple calculator using grid layout manager.



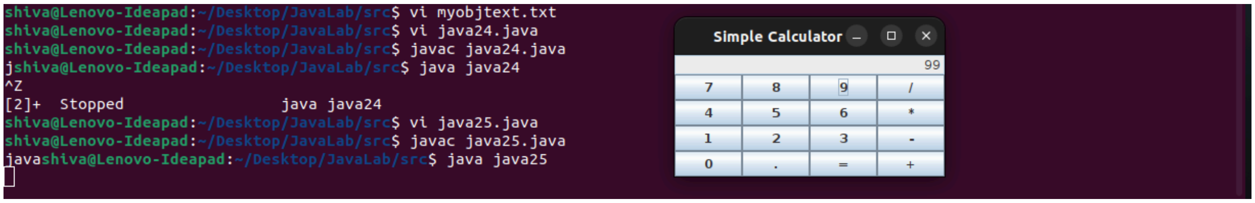
```
//functional calculator
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class java25 extends JFrame implements ActionListener {
    private JTextField textField;
    private String operator;
    private double firstOperand;
    public java25() {
        setTitle("Simple Calculator");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
        // creating text field
        textField = new JTextField();
        textField.setHorizontalAlignment(JTextField.RIGHT);
        textField.setEditable(false);
        add(textField, BorderLayout.NORTH);
        //buttons using gridlayout
        JPanel buttonPanel = new JPanel(new GridLayout(4, 4));
        String[] buttonLabels = {
                "7", "8", "9", "/",
                "4", "5", "6", "*",
                "1", "2", "3", "-",
                "0", ".", "=", "+"
        };
        for (String label : buttonLabels) {
            JButton button = new JButton(label);
            button.addActionListener(this);
            buttonPanel.add(button);
        }
        add(buttonPanel, BorderLayout.CENTER);
        pack();
        setLocationRelativeTo(null);
        setVisible(true);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        String command = e.getActionCommand();
        if (Character.isDigit(command.charAt(0)) || command.equals(".")) {
            textField.setText(textField.getText() + command);
        } else if (command.equals("=")) {
            calculateResult();
        } else {
            operator = command;
            firstOperand = Double.parseDouble(textField.getText());
            textField.setText("");
        }
    }
```

Code continues

P.T.O

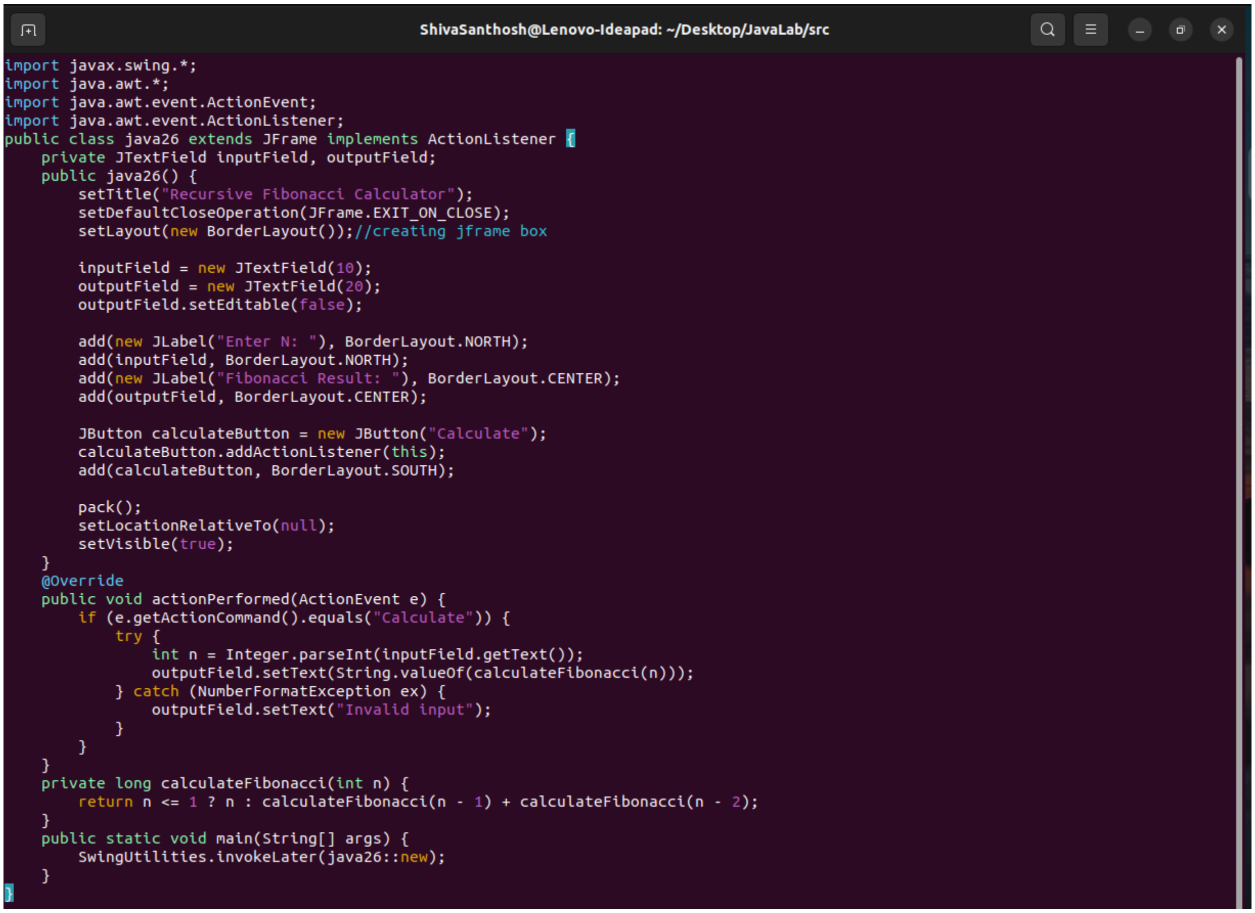


```
    private void calculateResult() {//function to claculate result
        double secondOperand = Double.parseDouble(textField.getText());
        double result = 0;
        switch (operator) {
            case "+":
                result = firstOperand + secondOperand;
                break;
            case "-":
                result = firstOperand - secondOperand;
                break;
            case "*":
                result = firstOperand * secondOperand;
                break;
            case "/":
                if (secondOperand != 0) {
                    result = firstOperand / secondOperand;
                } else {
                    textField.setText("Error");
                    return;
                }
                break;
        }

        textField.setText(String.valueOf(result));
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new java25());
    }
}
                                                                      78,1          Bot
```

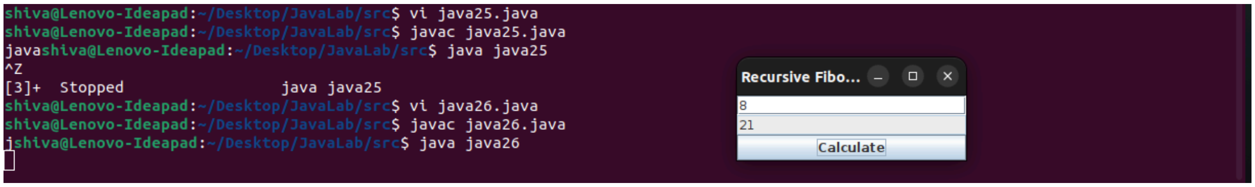


```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi myobjtext.txt
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java24.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java24.java
jshiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java24
^Z
[2]+  Stopped                 java java24
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java25.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java25.java
javashiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java25
```

A program to implement Recursive Fibonacci method using swing



```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class java26 extends JFrame implements ActionListener {
    private JTextField inputField, outputField;
    public java26() {
        setTitle("Recursive Fibonacci Calculator");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());//creating jframe box

        inputField = new JTextField(10);
        outputField = new JTextField(20);
        outputField.setEditable(false);

        add(new JLabel("Enter N: "), BorderLayout.NORTH);
        add(inputField, BorderLayout.NORTH);
        add(new JLabel("Fibonacci Result: "), BorderLayout.CENTER);
        add(outputField, BorderLayout.CENTER);

        JButton calculateButton = new JButton("Calculate");
        calculateButton.addActionListener(this);
        add(calculateButton, BorderLayout.SOUTH);

        pack();
        setLocationRelativeTo(null);
        setVisible(true);
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getActionCommand().equals("Calculate")) {
            try {
                int n = Integer.parseInt(inputField.getText());
                outputField.setText(String.valueOf(calculateFibonacci(n)));
            } catch (NumberFormatException ex) {
                outputField.setText("Invalid input");
            }
        }
    }
    private long calculateFibonacci(int n) {
        return n <= 1 ? n : calculateFibonacci(n - 1) + calculateFibonacci(n - 2);
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(java26::new);
    }
}
```

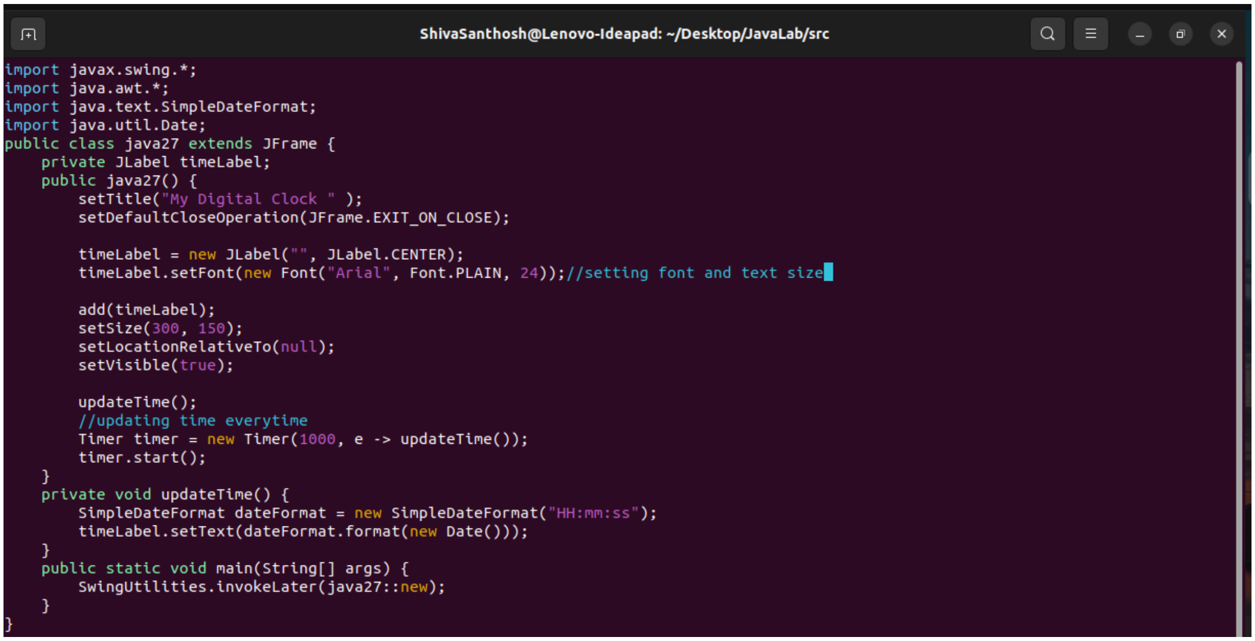


```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java25.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java25.java
javashiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java25
^Z
[3]+  Stopped                 java java25
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java26.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java26.java
jshiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java26
```

8

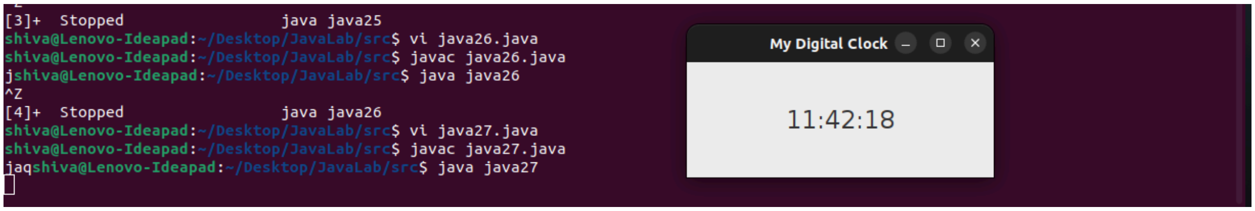
21

A program to display digital clock using swing



ShivaSanthosh@Lenovo-Ideapad: ~/Desktop/JavaLab/src

```
import javax.swing.*;
import java.awt.*;
import java.text.SimpleDateFormat;
import java.util.Date;
public class java27 extends JFrame {
    private JLabel timeLabel;
    public java27() {
        setTitle("My Digital Clock " );
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        timeLabel = new JLabel("", JLabel.CENTER);
        timeLabel.setFont(new Font("Arial", Font.PLAIN, 24));//setting font and text size

        add(timeLabel);
        setSize(300, 150);
        setLocationRelativeTo(null);
        setVisible(true);

        updateTime();
        //updating time everytime
        Timer timer = new Timer(1000, e -> updateTime());
        timer.start();
    }
    private void updateTime() {
        SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm:ss");
        timeLabel.setText(dateFormat.format(new Date()));
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(java27::new);
    }
}
```

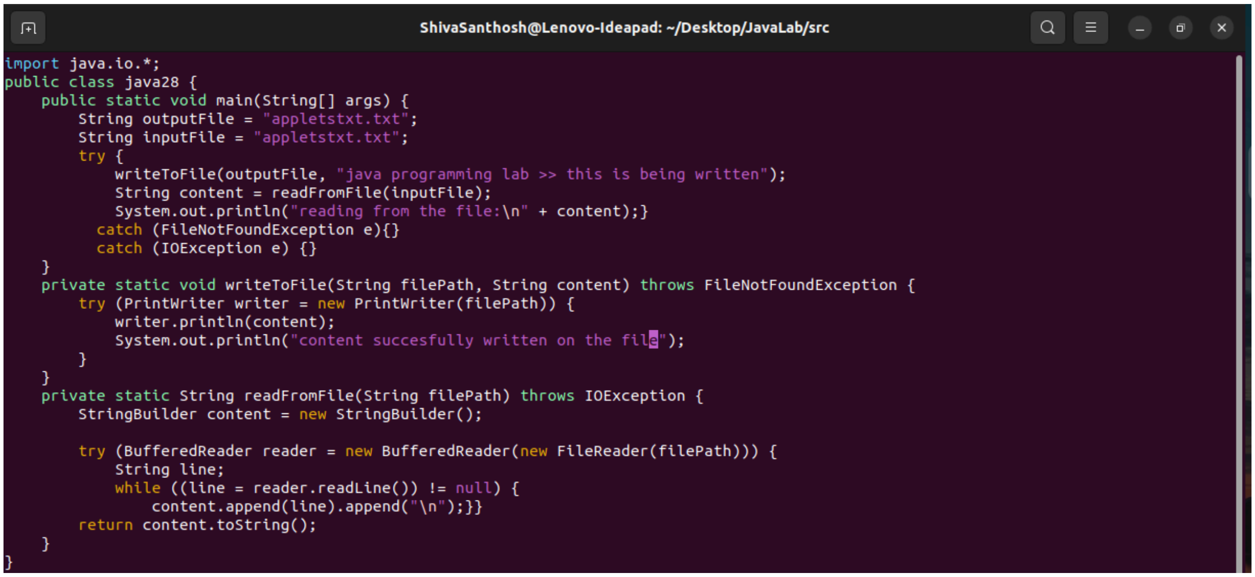


```
[3]+  Stopped                 java java25
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java26.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java26.java
jshiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java26
^Z
[4]+  Stopped                 java java26
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java27.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java27.java
jaqshiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java27
```

11:42:18

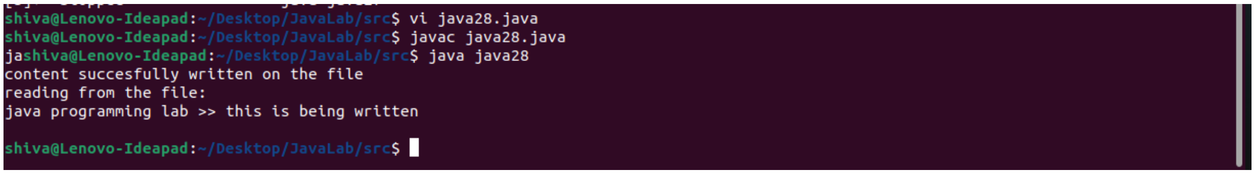
Displays current time

A program to read from a file and write to a file using Applet



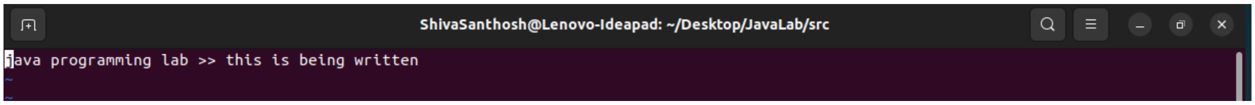
ShivaSanthosh@Lenovo-Ideapad: ~/Desktop/JavaLab/src

```
import java.io.*;
public class java28 {
    public static void main(String[] args) {
        String outputFile = "appletstxt.txt";
        String inputFile = "appletstxt.txt";
        try {
            writeToFile(outputFile, "java programming lab >> this is being written");
            String content = readFromFile(inputFile);
            System.out.println("reading from the file:\n" + content);}
          catch (FileNotFoundException e){}
          catch (IOException e) {}
    }
    private static void writeToFile(String filePath, String content) throws FileNotFoundException {
        try (PrintWriter writer = new PrintWriter(filePath)) {
            writer.println(content);
            System.out.println("content succesfully written on the file");
        }
    }
    private static String readFromFile(String filePath) throws IOException {
        StringBuilder content = new StringBuilder();

        try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) {
            String line;
            while ((line = reader.readLine()) != null) {
                content.append(line).append("\n");}}
        return content.toString();
    }
}
```



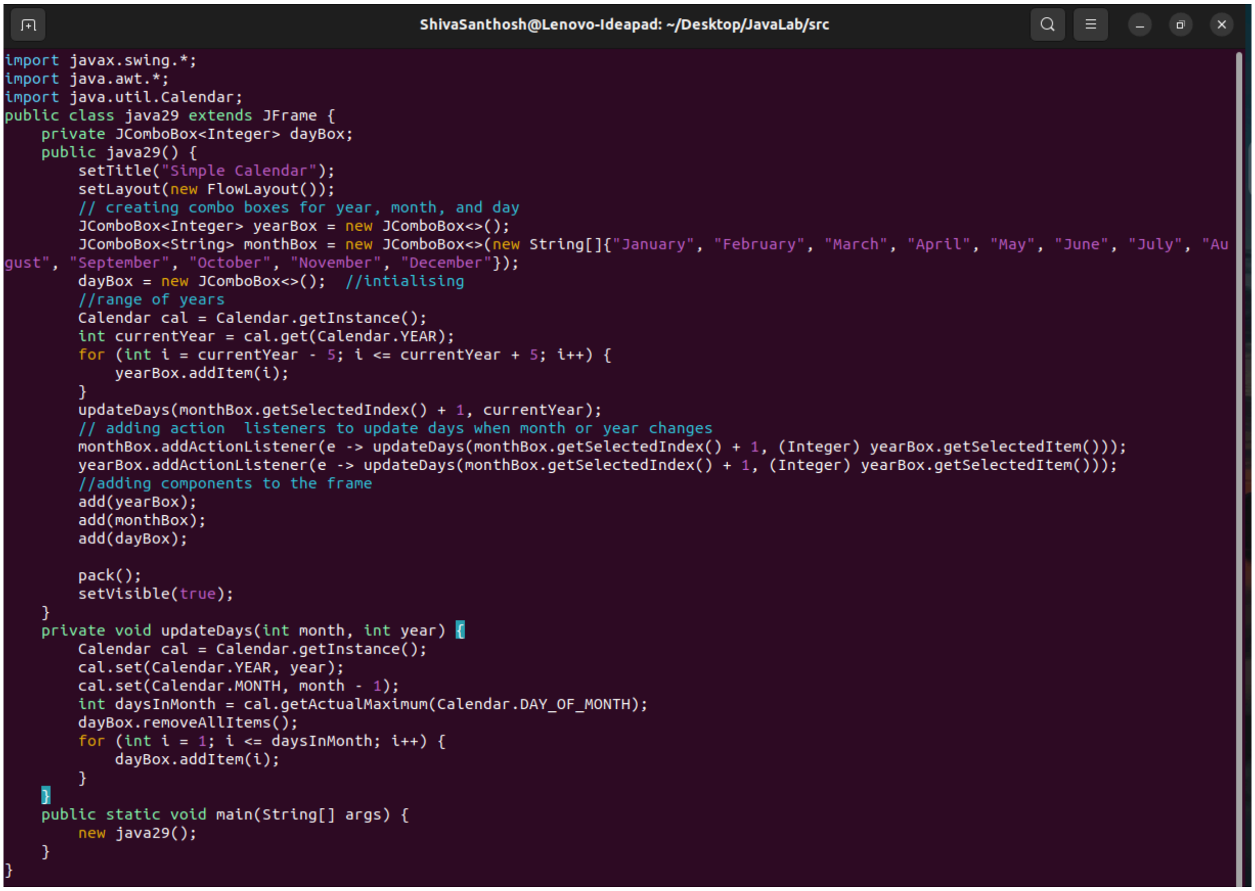
```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java28.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java28.java
jashiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java28
content succesfully written on the file
reading from the file:
java programming lab >> this is being written

shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ 
```

File -> appletstxt.txt

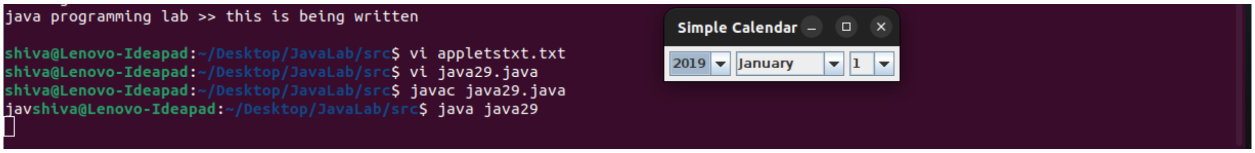


```
java programming lab >> this is being written
~
~
```

A program to display a calendar using JCombo box.



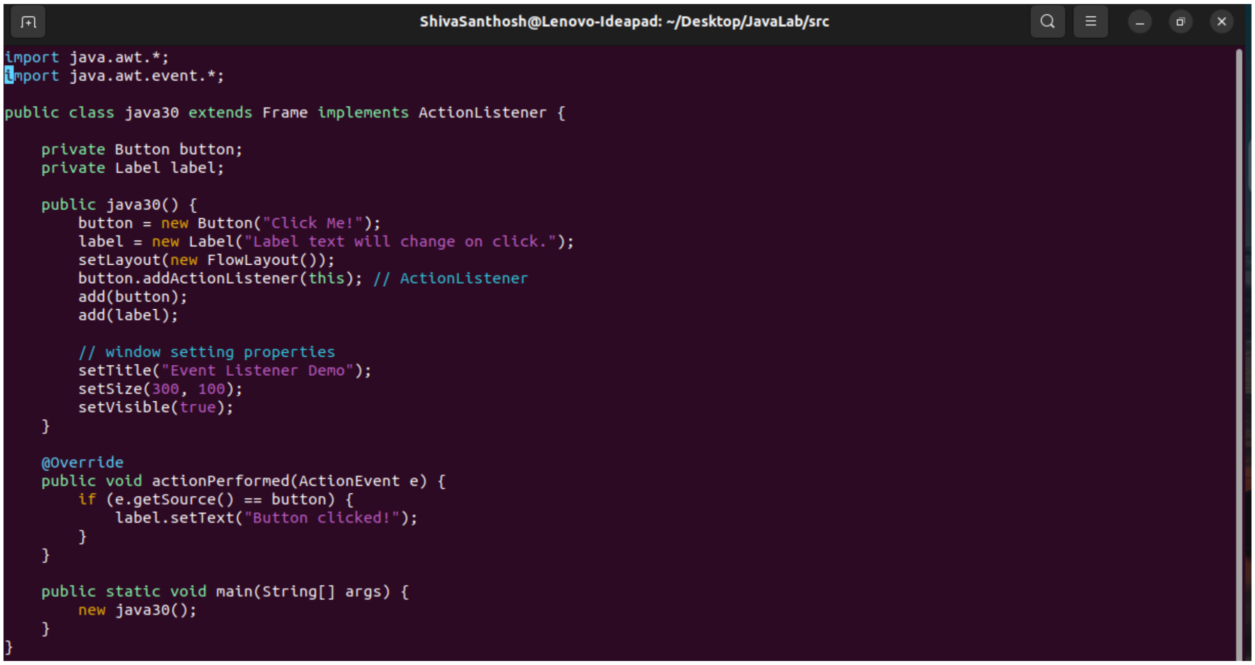
```java
import javax.swing.*;
import java.awt.*;
import java.util.Calendar;
public class java29 extends JFrame {
    private JComboBox<Integer> dayBox;
    public java29() {
        setTitle("Simple Calendar");
        setLayout(new FlowLayout());
        // creating combo boxes for year, month, and day
        JComboBox<Integer> yearBox = new JComboBox<>();
        JComboBox<String> monthBox = new JComboBox<>(new String[]{"January", "February", "March", "April", "May", "June", "July", "Au
gust", "September", "October", "November", "December"});
        dayBox = new JComboBox<>();  //intialising
        //range of years
        Calendar cal = Calendar.getInstance();
        int currentYear = cal.get(Calendar.YEAR);
        for (int i = currentYear - 5; i <= currentYear + 5; i++) {
            yearBox.addItem(i);
        }
        updateDays(monthBox.getSelectedIndex() + 1, currentYear);
        // adding action  listeners to update days when month or year changes
        monthBox.addActionListener(e -> updateDays(monthBox.getSelectedIndex() + 1, (Integer) yearBox.getSelectedItem()));
        yearBox.addActionListener(e -> updateDays(monthBox.getSelectedIndex() + 1, (Integer) yearBox.getSelectedItem()));
        //adding components to the frame
        add(yearBox);
        add(monthBox);
        add(dayBox);

        pack();
        setVisible(true);
    }
    private void updateDays(int month, int year) {
        Calendar cal = Calendar.getInstance();
        cal.set(Calendar.YEAR, year);
        cal.set(Calendar.MONTH, month - 1);
        int daysInMonth = cal.getActualMaximum(Calendar.DAY_OF_MONTH);
        dayBox.removeAllItems();
        for (int i = 1; i <= daysInMonth; i++) {
            dayBox.addItem(i);
        }
    }
    public static void main(String[] args) {
        new java29();
    }
}
```



```
java programming lab >> this is being written

shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi appletstxt.txt
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java29.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java29.java
javshiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java29
```

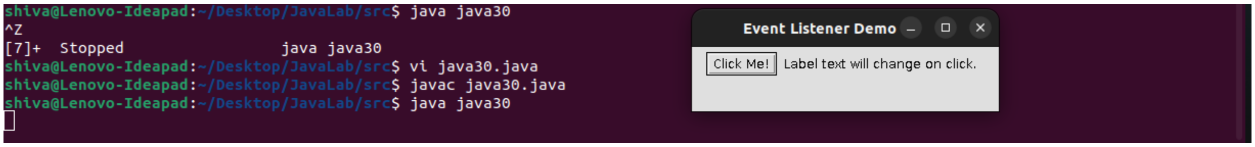
2019 ▾ January ▾ 1 ▾

A program to illustrate event listener interfaces.



ShivaSanthosh@Lenovo-Ideapad: ~/Desktop/JavaLab/src

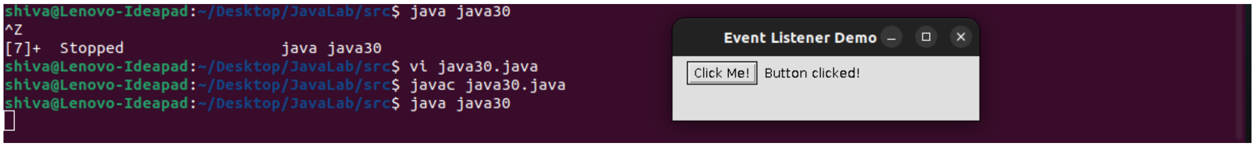
```
import java.awt.*;
import java.awt.event.*;

public class java30 extends Frame implements ActionListener {

    private Button button;
    private Label label;

    public java30() {
        button = new Button("Click Me!");
        label = new Label("Label text will change on click.");
        setLayout(new FlowLayout());
        button.addActionListener(this); // ActionListener
        add(button);
        add(label);

        // window setting properties
        setTitle("Event Listener Demo");
        setSize(300, 100);
        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == button) {
            label.setText("Button clicked!");
        }
    }

    public static void main(String[] args) {
        new java30();
    }
}
```



```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java30
^Z
[7]+  Stopped                 java java30
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java30.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java30.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java30
```

Event Listener Demo

Label text will change on click.



```
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java30
^Z
[7]+  Stopped                 java java30
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ vi java30.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ javac java30.java
shiva@Lenovo-Ideapad:~/Desktop/JavaLab/src$ java java30
```

Event Listener Demo

Button clicked!