JAVA PROGRAMMING
(CSE-1007)

LAB ASSESSMENT -2

Name: Mathew Jerry Meleth
Reg No: 17BIT0050

SCENARIO – I

Write a program to demonstrate the knowledge of students in Inheritance. Eg: Assume that a bank maintains two kinds of accounts for customers, one called as savings account and the other as current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class account that stores customer name, account number and type of account. From this derive the classes cur_acct and sav_acct to make them more specific to their requirements. Include necessary member functions in order to achieve the following tasks :
a) Accept deposit from a customer and update the balance.
b) Display the balance
c) Compute and deposit interest.
d) Permit withdrawal and update the balance.
e) Check for the minimum balance, impose penalty, necessary, and update the balance.

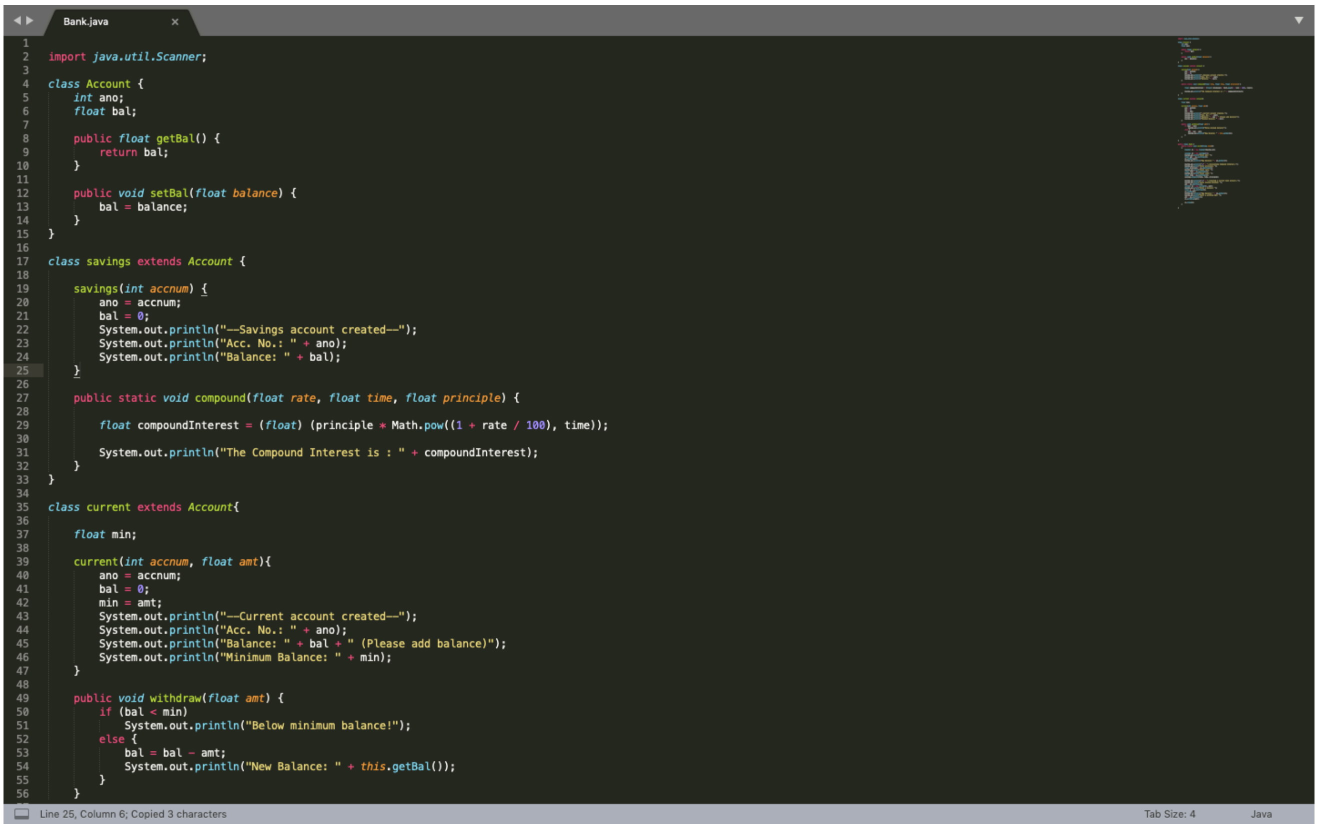
SOURCE CODE:

```
import java.util.Scanner;


class Account {

	int ano;

	float bal;


	public float getBal() {

		return bal;

	}


	public void setBal(float balance) {

		bal = balance;

	}

}


class savings extends Account {


	savings(int accnum) {

		ano = accnum;

		bal = 0;

		System.out.println("--Savings account created--");

		System.out.println("Acc. No.: " + ano);
```

```
        System.out.println("Balance: " + bal);
    }

    public static void compound(float rate, float time, float principle) {

        float compoundInterest = (float) (principle * Math.pow((1 + rate / 100), time));

        System.out.println("The Compound Interest is : " + compoundInterest);
    }
}

class current extends Account{

    float min;

    current(int accnum, float amt){
        ano = accnum;
        bal = 0;
        min = amt;
        System.out.println("--Current account created--");
        System.out.println("Acc. No.: " + ano);
        System.out.println("Balance: " + bal + " (Please add balance)");
        System.out.println("Minimum Balance: " + min);
    }
```

```
        public void withdraw(float amt) {

                if (bal < min)

                        System.out.println("Below minimum balance!");

                else {

                        bal = bal - amt;

                        System.out.println("New Balance: " + this.getBal());

                }

        }


}


public class Bank {

        public static void main(String args[])

        {

                Scanner sc = new Scanner(System.in);



                savings s1 = new savings(1);

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

                float amt = sc.nextFloat();

                s1.setBal(amt);

                System.out.println("New Balance: " + s1.getBal());



                System.out.println("\n" + "--Calculating Compound Interest--");
                System.out.print("Enter principle: ");

                float principle = sc.nextFloat();

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

                float rate = sc.nextFloat();

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

                float time = sc.nextFloat();

                savings.compound(rate, time, principle);



                System.out.println("\n" + "--Creating a current bank account--");

                System.out.print("Enter minimum Balance: ");

                amt = sc.nextFloat();

                current c1 = new current(2, amt);

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

                float b = sc.nextFloat();

                c1.setBal(b);

                System.out.println("New Balance: " + c1.getBal());

                System.out.print("Enter a withdraw Amt: ");

                amt = sc.nextFloat();

                c1.withdraw(amt);



                sc.close();

        }



}
```

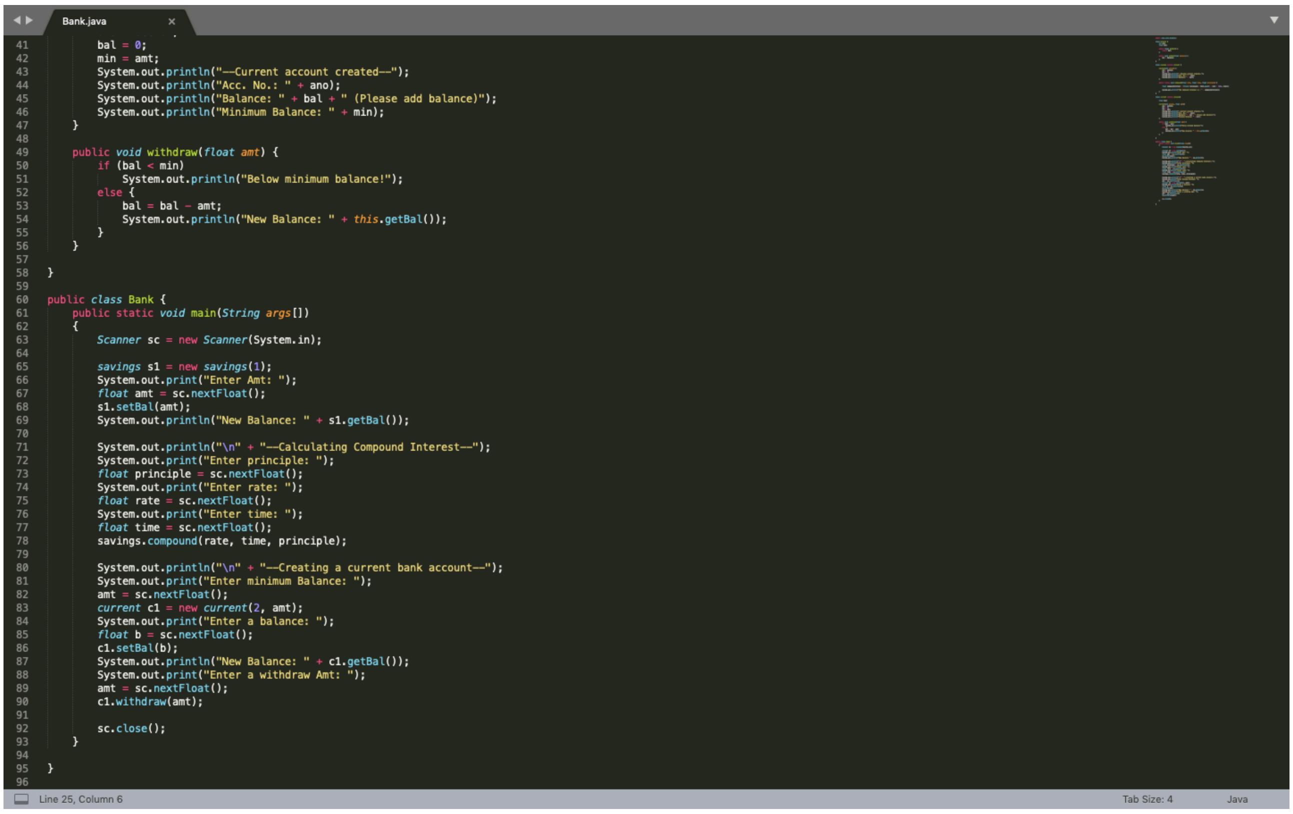
EXECUTION:



Bank.java

```java

import java.util.Scanner;

class Account {
    int ano;
    float bal;

    public float getBal() {
        return bal;
    }

    public void setBal(float balance) {
        bal = balance;
    }
}

class savings extends Account {

    savings(int accnum) {
        ano = accnum;
        bal = 0;
        System.out.println("---Savings account created---");
        System.out.println("Acc. No.: " + ano);
        System.out.println("Balance: " + bal);
    }

    public static void compound(float rate, float time, float principle) {

        float compoundInterest = (float) (principle * Math.pow((1 + rate / 100), time));

        System.out.println("The Compound Interest is : " + compoundInterest);
    }
}

class current extends Account{

    float min;

    current(int accnum, float amt){
        ano = accnum;
        bal = 0;
        min = amt;
        System.out.println("---Current account created---");
        System.out.println("Acc. No.: " + ano);
        System.out.println("Balance: " + bal + " (Please add balance)");
        System.out.println("Minimum Balance: " + min);
    }

    public void withdraw(float amt) {
        if (bal < min)
            System.out.println("Below minimum balance!");
        else {
            bal = bal - amt;
            System.out.println("New Balance: " + this.getBal());
        }
    }
```

Line 25, Column 6; Copied 3 characters   Tab Size: 4   Java

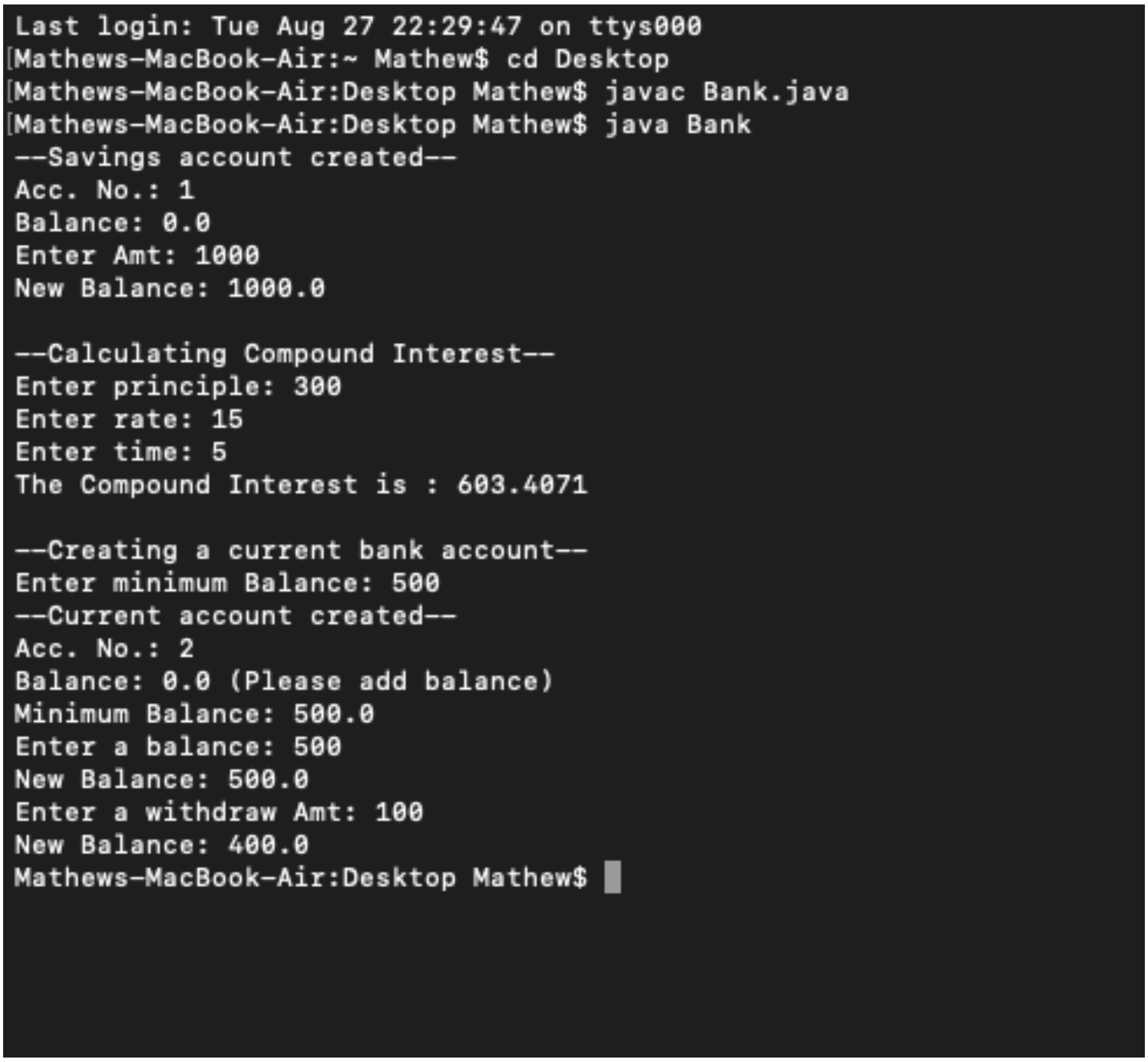


Bank.java

```java
        bal = 0;
        min = amt;
        System.out.println("---Current account created---");
        System.out.println("Acc. No.: " + ano);
        System.out.println("Balance: " + bal + " (Please add balance)");
        System.out.println("Minimum Balance: " + min);
    }

    public void withdraw(float amt) {
        if (bal < min)
            System.out.println("Below minimum balance!");
        else {
            bal = bal - amt;
            System.out.println("New Balance: " + this.getBal());
        }
    }

}

public class Bank {
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);

        savings s1 = new savings(1);
        System.out.print("Enter Amt: ");
        float amt = sc.nextFloat();
        s1.setBal(amt);
        System.out.println("New Balance: " + s1.getBal());

        System.out.println("\n" + "---Calculating Compound Interest---");
        System.out.print("Enter principle: ");
        float principle = sc.nextFloat();
        System.out.print("Enter rate: ");
        float rate = sc.nextFloat();
        System.out.print("Enter time: ");
        float time = sc.nextFloat();
        savings.compound(rate, time, principle);

        System.out.println("\n" + "---Creating a current bank account---");
        System.out.print("Enter minimum Balance: ");
        amt = sc.nextFloat();
        current c1 = new current(2, amt);
        System.out.print("Enter a balance: ");
        float b = sc.nextFloat();
        c1.setBal(b);
        System.out.println("New Balance: " + c1.getBal());
        System.out.print("Enter a withdraw Amt: ");
        amt = sc.nextFloat();
        c1.withdraw(amt);

        sc.close();
    }

}

```

Line 25, Column 6   Tab Size: 4   Java

RESULT:



```
Last login: Tue Aug 27 22:29:47 on ttys000
[Mathews-MacBook-Air:~ Mathew$ cd Desktop
[Mathews-MacBook-Air:Desktop Mathew$ javac Bank.java
[Mathews-MacBook-Air:Desktop Mathew$ java Bank
--Savings account created--
Acc. No.: 1
Balance: 0.0
Enter Amt: 1000
New Balance: 1000.0

--Calculating Compound Interest--
Enter principle: 300
Enter rate: 15
Enter time: 5
The Compound Interest is : 603.4071

--Creating a current bank account--
Enter minimum Balance: 500
--Current account created--
Acc. No.: 2
Balance: 0.0 (Please add balance)
Minimum Balance: 500.0
Enter a balance: 500
New Balance: 500.0
Enter a withdraw Amt: 100
New Balance: 400.0
Mathews-MacBook-Air:Desktop Mathew$
```

SCENARIO – II

1. 3. Design a class to display the schedule of trains in MGR Central railway station. The class can have its own member variable like, train_no, src, dest, time, traveltime, platformno, traveltype (A-Arrival, D-Departure). Create an array of objects in main function. Perform the following tasks.

- SearchFunction ( ) – Takes trainno has input and perfroms a search with all objects. If found display all variable details of the class like train_no, src, dest, time, traveltime, platformno, traveltype. If not found throw arrayIndexOutofBound exception and handle it.

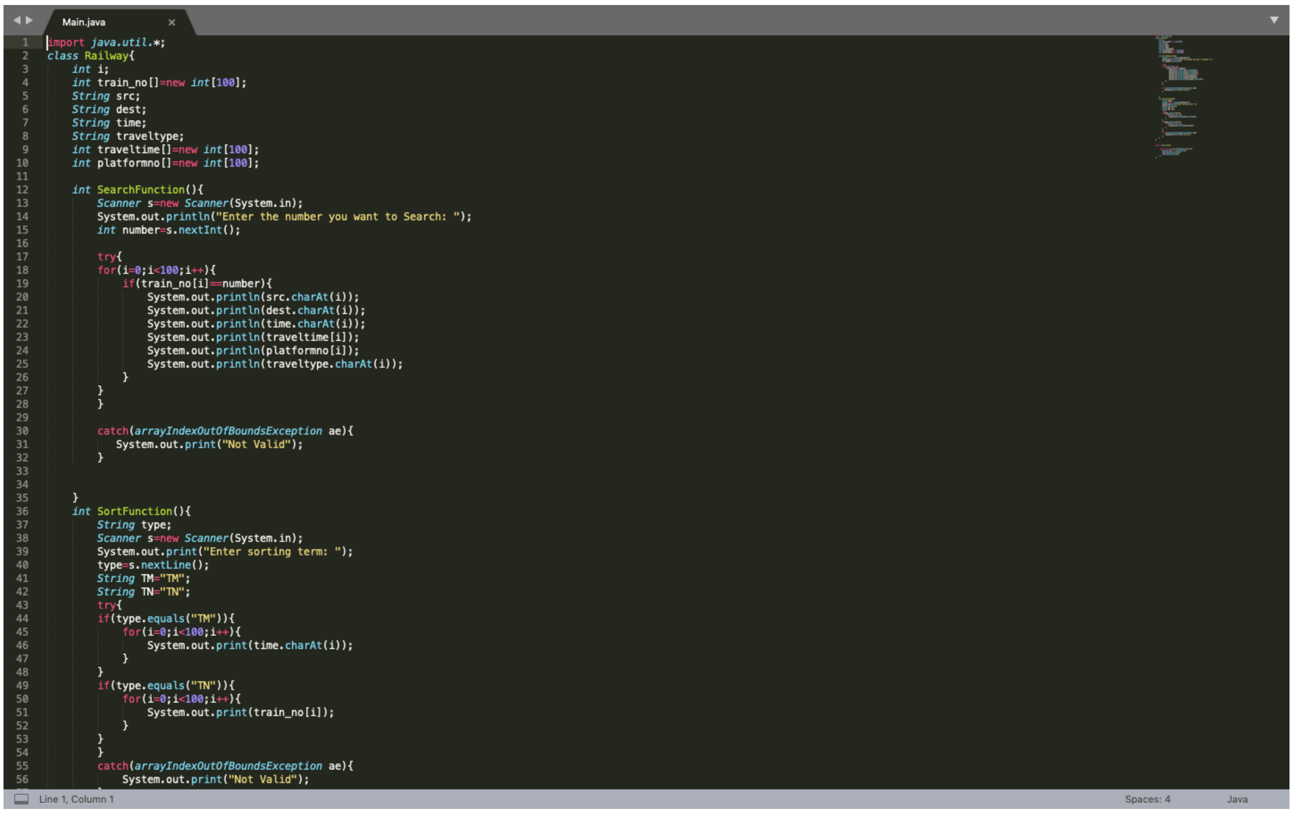
- SortFunction ( ) – Takes the input of sorting type (TM- time based, TN – Train no based). Sort and display the records. If the user enters other than TN or TM input throw arithematic or arrayIndexOutofBound exception and handle it.

SOURCE CODE

```
import java.util.*;

class Railway{

   int i;

   int train_no[]=new int[100];

   String src;

   String dest;

   String time;

   String traveltype;

   int traveltime[]=new int[100];

   int platformno[]=new int[100];
int SearchFunction(){
    Scanner s=new Scanner(System.in);
    System.out.println("Enter the number you want to Search: ");
    int number=s.nextInt();

    try{
    for(i=0;i<100;i++){
        if(train_no[i]==number){
            System.out.println(src.charAt(i));
            System.out.println(dest.charAt(i));
            System.out.println(time.charAt(i));
            System.out.println(traveltime[i]);
            System.out.println(platformno[i]);
            System.out.println(traveltype.charAt(i));
        }
    }
    }

    catch(arrayIndexOutOfBoundsException ae){
       System.out.print("Not Valid");
    }
}
int SortFunction(){
    String type;
    Scanner s=new Scanner(System.in);
    System.out.print("Enter sorting term: ");
    type=s.nextLine();
    String TM="TM";
    String TN="TN";
    try{
    if(type.equals("TM")){
        for(i=0;i<100;i++){
            System.out.print(time.charAt(i));
        }
    }
    if(type.equals("TN")){
        for(i=0;i<100;i++){
            System.out.print(train_no[i]);
        }
    }
    }
    catch(arrayIndexOutOfBoundsException ae){
        System.out.print("Not Valid");
    }
}
```

```
}



public class Main{

    public static void main(String []args){
      Railway r[]=new Railway[100];
      r[0].SearchFunction();
      r[0].SortFunction();
    }
}
```

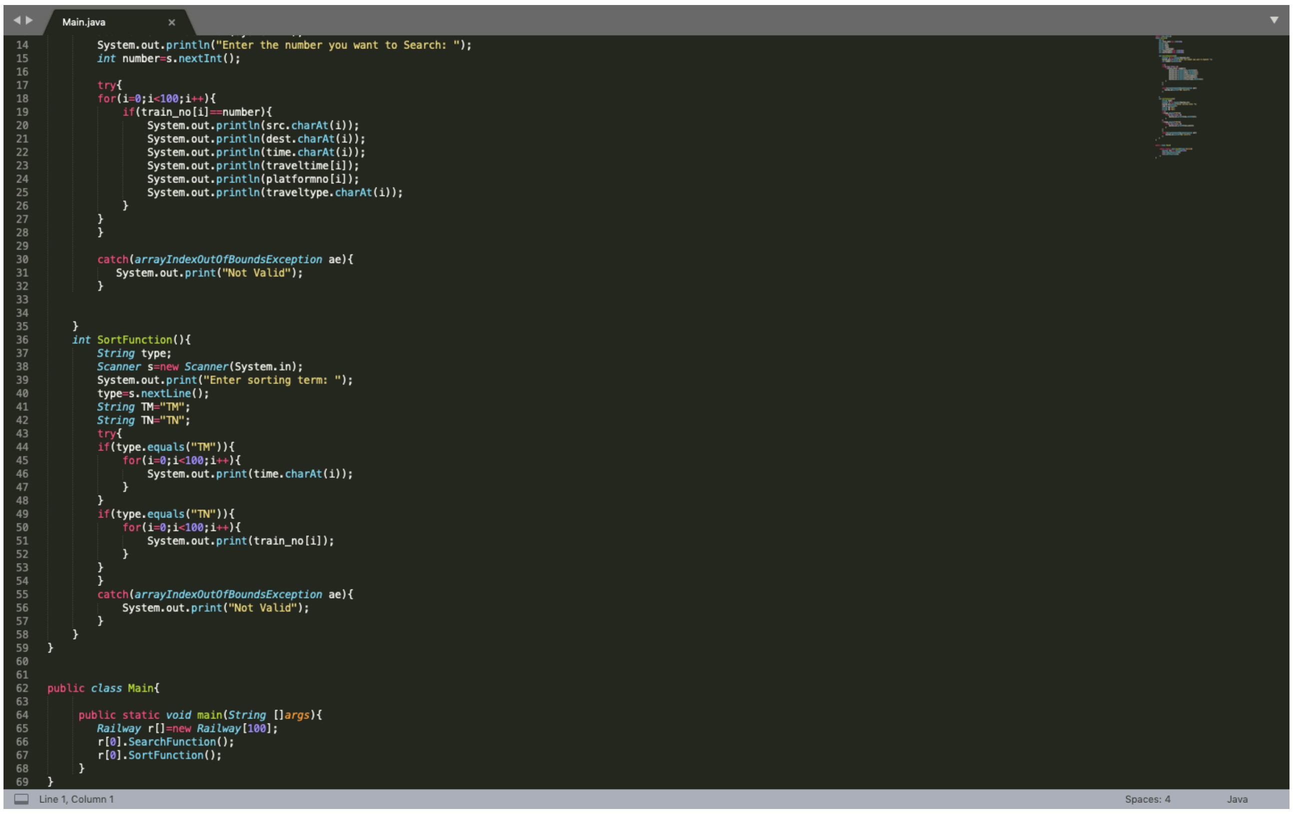
EXECUTION:



Main.java

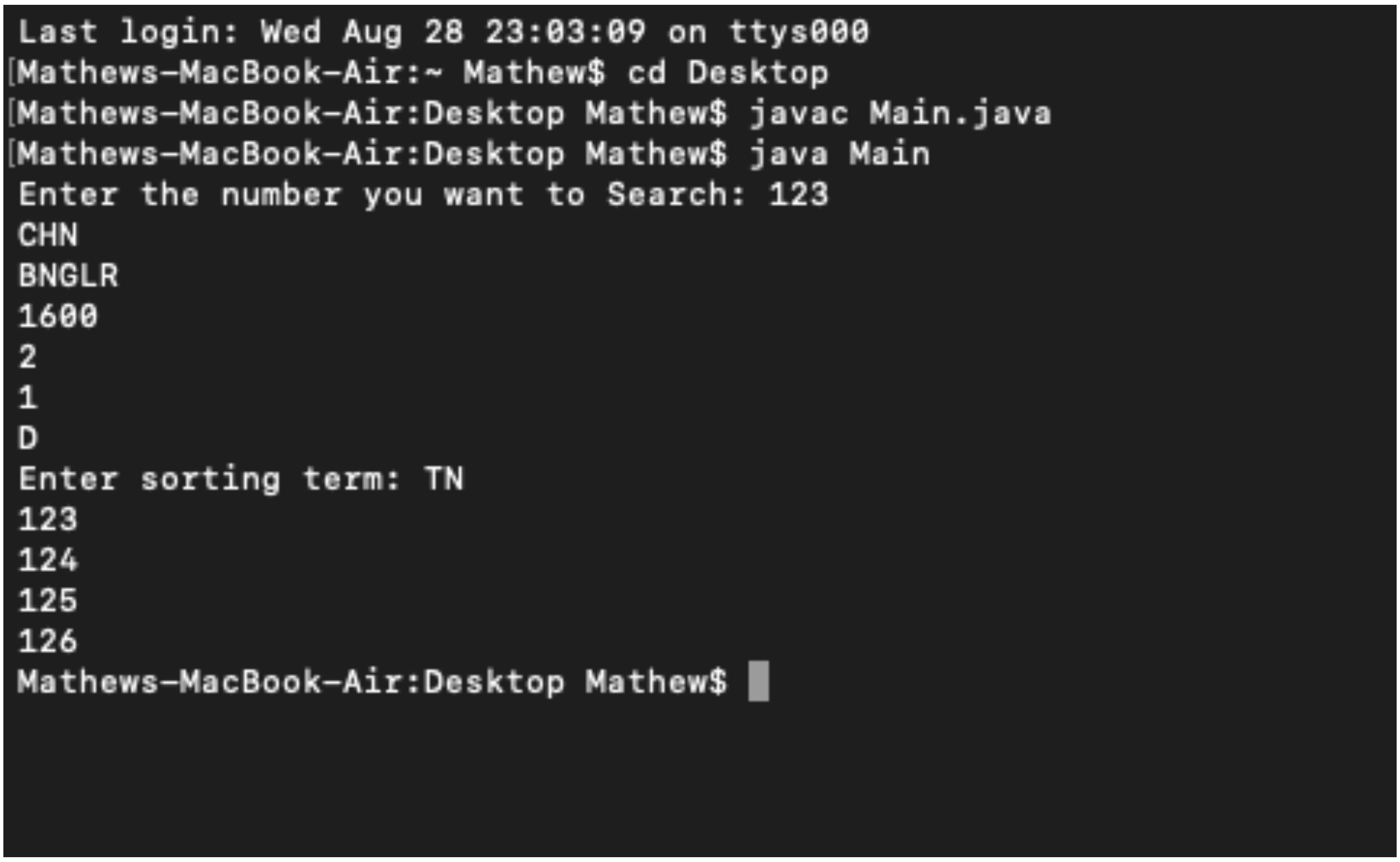
```
import java.util.*;
class Railway{
    int i;
    int train_no[]=new int[100];
    String src;
    String dest;
    String time;
    String traveltype;
    int traveltime[]=new int[100];
    int platformno[]=new int[100];

    int SearchFunction(){
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the number you want to Search: ");
        int number=s.nextInt();

        try{
        for(i=0;i<100;i++){
            if(train_no[i]==number){
                System.out.println(src.charAt(i));
                System.out.println(dest.charAt(i));
                System.out.println(time.charAt(i));
                System.out.println(traveltime[i]);
                System.out.println(platformno[i]);
                System.out.println(traveltype.charAt(i));
            }
        }
        }

        catch(arrayIndexOutOfBoundsException ae){
           System.out.print("Not Valid");
        }


    }
    int SortFunction(){
        String type;
        Scanner s=new Scanner(System.in);
        System.out.print("Enter sorting term: ");
        type=s.nextLine();
        String TM="TM";
        String TN="TN";
        try{
        if(type.equals("TM")){
            for(i=0;i<100;i++){
                System.out.print(time.charAt(i));
            }
        }
        if(type.equals("TN")){
            for(i=0;i<100;i++){
                System.out.print(train_no[i]);
            }
        }
        }
        catch(arrayIndexOutOfBoundsException ae){
            System.out.print("Not Valid");
```

Line 1, Column 1   Spaces: 4   Java



```
        System.out.println("Enter the number you want to Search: ");
        int number=s.nextInt();

        try{
        for(i=0;i<100;i++){
            if(train_no[i]==number){
                System.out.println(src.charAt(i));
                System.out.println(dest.charAt(i));
                System.out.println(time.charAt(i));
                System.out.println(traveltime[i]);
                System.out.println(platformno[i]);
                System.out.println(traveltype.charAt(i));
            }
        }
        }

        catch(arrayIndexOutOfBoundsException ae){
           System.out.print("Not Valid");
        }


    }
    int SortFunction(){
        String type;
        Scanner s=new Scanner(System.in);
        System.out.print("Enter sorting term: ");
        type=s.nextLine();
        String TM="TM";
        String TN="TN";
        try{
        if(type.equals("TM")){
            for(i=0;i<100;i++){
                System.out.print(time.charAt(i));
            }
        }
        if(type.equals("TN")){
            for(i=0;i<100;i++){
                System.out.print(train_no[i]);
            }
        }
        }
        catch(arrayIndexOutOfBoundsException ae){
            System.out.print("Not Valid");
        }
    }
}


public class Main{

     public static void main(String []args){
        Railway r[]=new Railway[100];
        r[0].SearchFunction();
        r[0].SortFunction();
     }
}
```

RESULT:



```
Last login: Wed Aug 28 23:03:09 on ttys000
[Mathews-MacBook-Air:~ Mathew$ cd Desktop
[Mathews-MacBook-Air:Desktop Mathew$ javac Main.java
[Mathews-MacBook-Air:Desktop Mathew$ java Main
Enter the number you want to Search: 123
CHN
BNGLR
1600
2
1
D
Enter sorting term: TN
123
124
125
126
Mathews-MacBook-Air:Desktop Mathew$ 
```

SCENARIO – III

Write a program to demonstrate the knowledge of students in Java Exception handling.

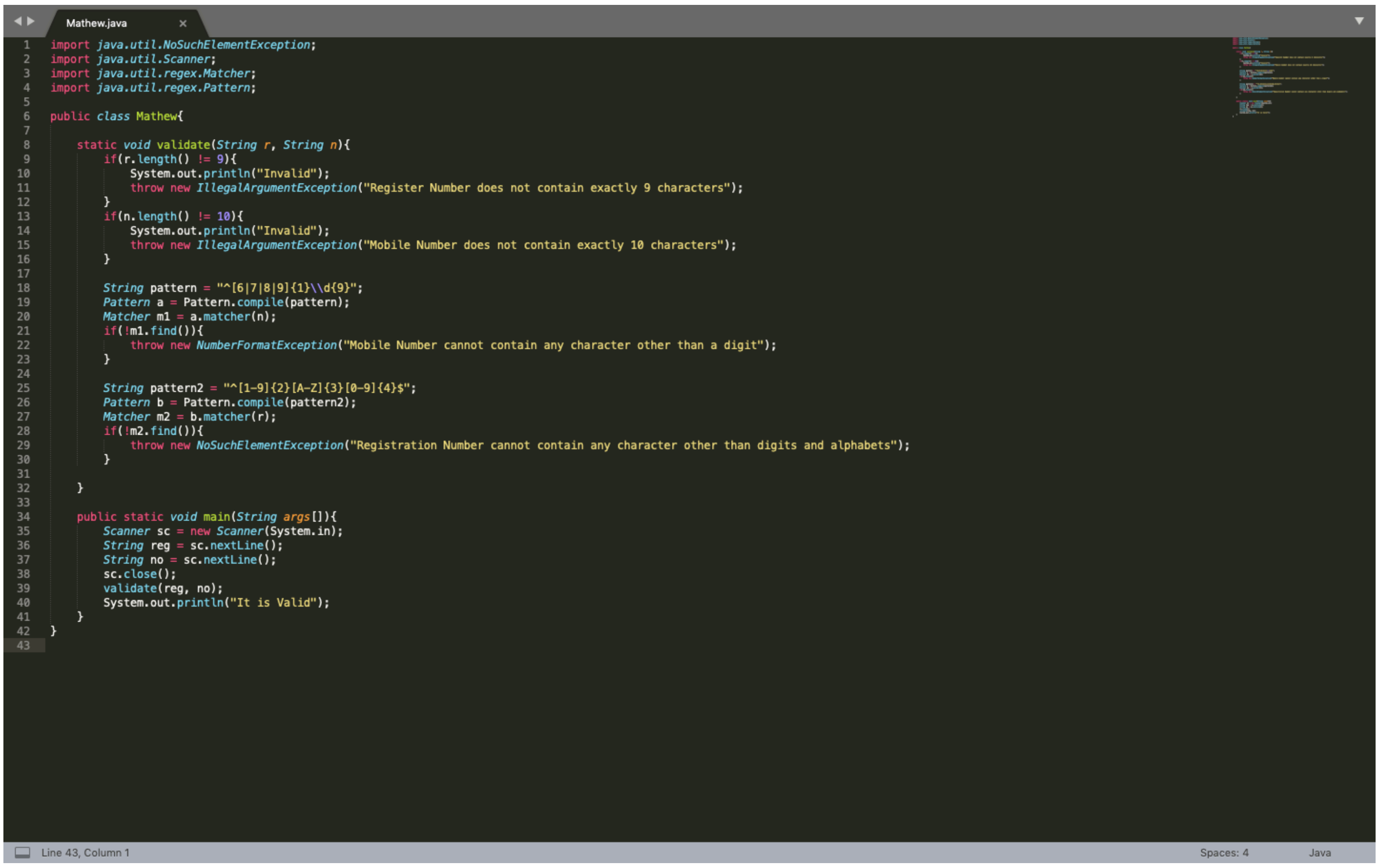
Eg., Read the Register Number and Mobile Number of a student. If the Register Number does not contain exactly 9 characters or if the Mobile Number does not contain exactly 10 characters, throw an IllegalArgumentException. If the Mobile Number contains any character other than a digit, raise a NumberFormatException. If the Register Number contains any character other than digits and alphabets, throw a NoSuchElementException. If they are valid, print the message 'valid' else 'invalid'

SOURCE CODE:

```
import java.util.NoSuchElementException;

import java.util.Scanner;

import java.util.regex.Matcher;

import java.util.regex.Pattern;



public class Mathew{



  static void validate(String r, String n){

    if(r.length() != 9){

      System.out.println("Invalid");

      throw new IllegalArgumentException("Register Number does not contain exactly 9
characters");

    }

    if(n.length() != 10){

      System.out.println("Invalid");
```

```
            throw new IllegalArgumentException("Mobile Number does not contain exactly 10
characters");

        }


        String pattern = "^[6|7|8|9]{1}\\d{9}";

        Pattern a = Pattern.compile(pattern);

        Matcher m1 = a.matcher(n);

        if(!m1.find()){

            throw new NumberFormatException("Mobile Number cannot contain any character other
than a digit");

        }


        String pattern2 = "^[1-9]{2}[A-Z]{3}[0-9]{4}$";

        Pattern b = Pattern.compile(pattern2);

        Matcher m2 = b.matcher(r);

        if(!m2.find()){

            throw new NoSuchElementException("Registration Number cannot contain any character
other than digits and alphabets");

        }



    }


    public static void main(String args[]){

        Scanner sc = new Scanner(System.in);

        String reg = sc.nextLine();

        String no = sc.nextLine();
        sc.close();

        validate(reg, no);

        System.out.println("It is Valid");

    }

}
```

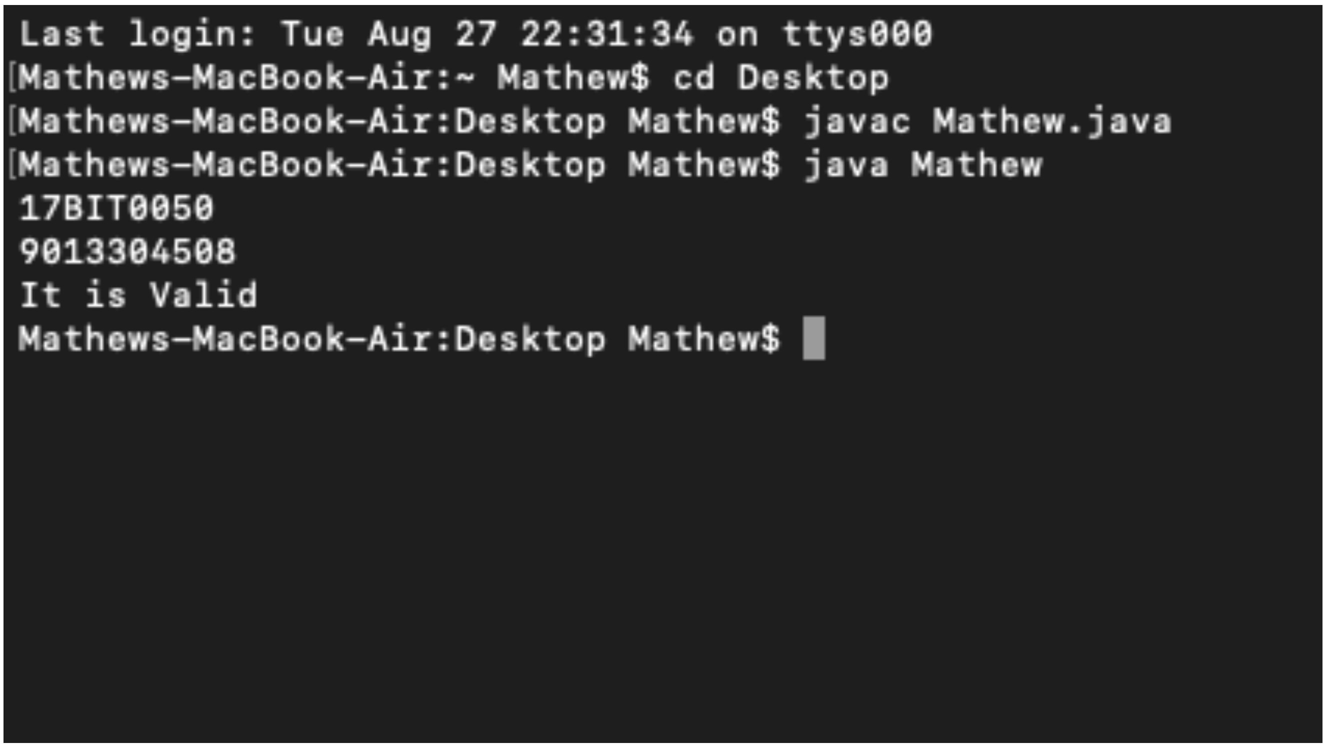
EXECUTION:



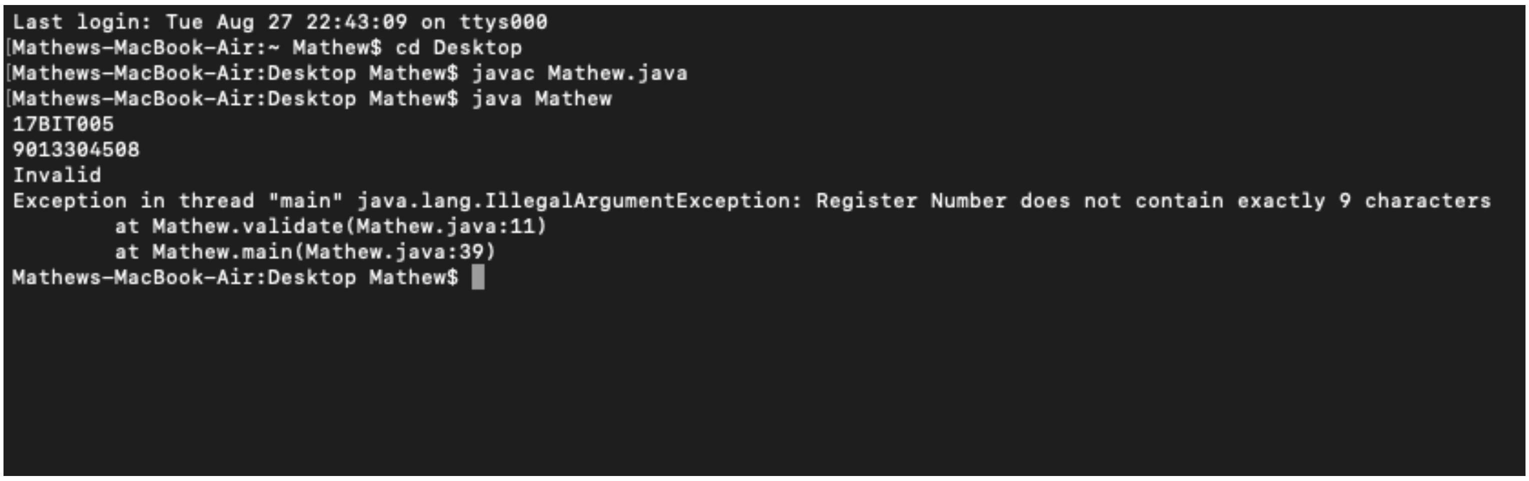
Mathew.java

```
import java.util.NoSuchElementException;
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Mathew{

    static void validate(String r, String n){
        if(r.length() != 9){
            System.out.println("Invalid");
            throw new IllegalArgumentException("Register Number does not contain exactly 9 characters");
        }
        if(n.length() != 10){
            System.out.println("Invalid");
            throw new IllegalArgumentException("Mobile Number does not contain exactly 10 characters");
        }

        String pattern = "^[6|7|8|9]{1}\\d{9}";
        Pattern a = Pattern.compile(pattern);
        Matcher m1 = a.matcher(n);
        if(!m1.find()){
            throw new NumberFormatException("Mobile Number cannot contain any character other than a digit");
        }

        String pattern2 = "^[1-9]{2}[A-Z]{3}[0-9]{4}$";
        Pattern b = Pattern.compile(pattern2);
        Matcher m2 = b.matcher(r);
        if(!m2.find()){
            throw new NoSuchElementException("Registration Number cannot contain any character other than digits and alphabets");
        }

    }

    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        String reg = sc.nextLine();
        String no = sc.nextLine();
        sc.close();
        validate(reg, no);
        System.out.println("It is Valid");
    }
}
```

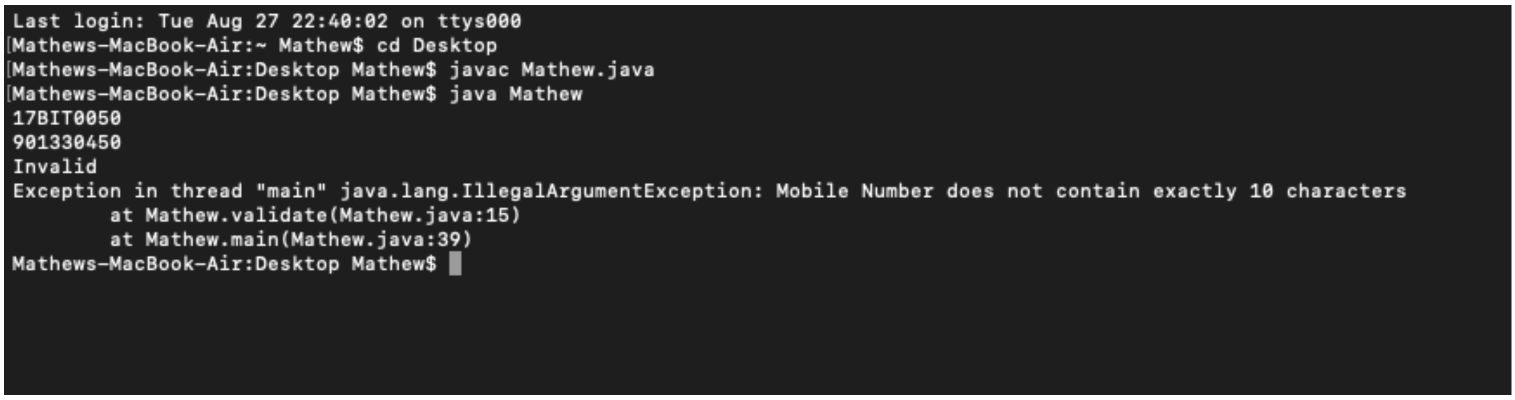
:RESULTS:



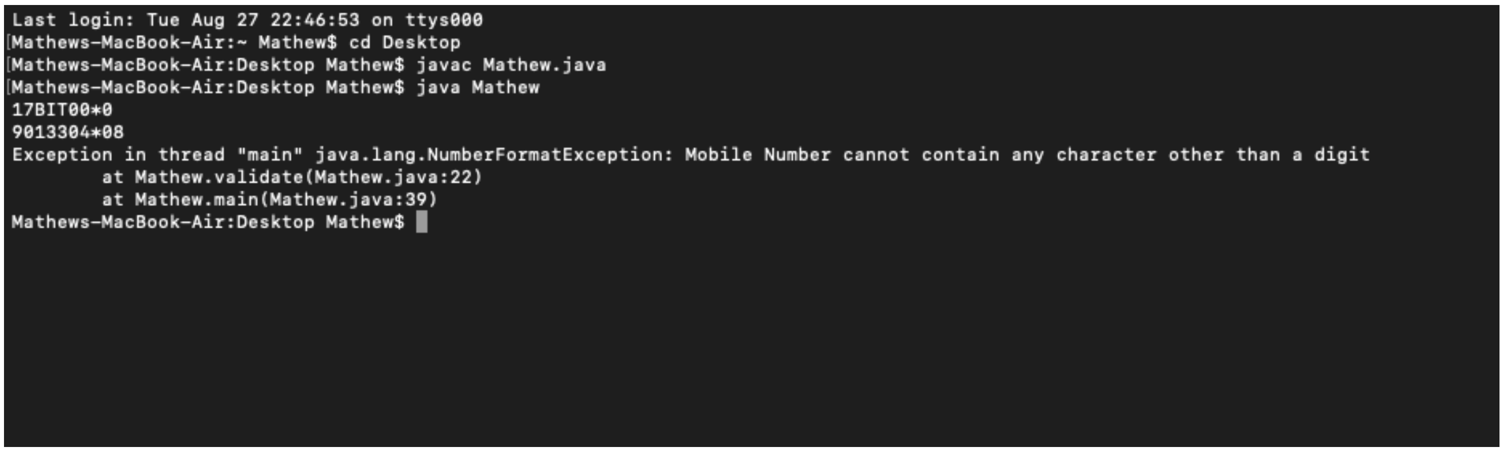
```
Last login: Tue Aug 27 22:31:34 on ttys000
[Mathews-MacBook-Air:~ Mathew$ cd Desktop
[Mathews-MacBook-Air:Desktop Mathew$ javac Mathew.java
[Mathews-MacBook-Air:Desktop Mathew$ java Mathew
17BIT0050
9013304508
It is Valid
Mathews-MacBook-Air:Desktop Mathew$
```



```
Last login: Tue Aug 27 22:43:09 on ttys000
[Mathews-MacBook-Air:~ Mathew$ cd Desktop
[Mathews-MacBook-Air:Desktop Mathew$ javac Mathew.java
[Mathews-MacBook-Air:Desktop Mathew$ java Mathew
17BIT005
9013304508
Invalid
Exception in thread "main" java.lang.IllegalArgumentException: Register Number does not contain exactly 9 characters
        at Mathew.validate(Mathew.java:11)
        at Mathew.main(Mathew.java:39)
Mathews-MacBook-Air:Desktop Mathew$
```



```
Last login: Tue Aug 27 22:40:02 on ttys000
Mathews-MacBook-Air:~ Mathew$ cd Desktop
Mathews-MacBook-Air:Desktop Mathew$ javac Mathew.java
Mathews-MacBook-Air:Desktop Mathew$ java Mathew
17BIT0050
901330450
Invalid
Exception in thread "main" java.lang.IllegalArgumentException: Mobile Number does not contain exactly 10 characters
        at Mathew.validate(Mathew.java:15)
        at Mathew.main(Mathew.java:39)
Mathews-MacBook-Air:Desktop Mathew$ 
```



```
Last login: Tue Aug 27 22:46:53 on ttys000
Mathews-MacBook-Air:~ Mathew$ cd Desktop
Mathews-MacBook-Air:Desktop Mathew$ javac Mathew.java
Mathews-MacBook-Air:Desktop Mathew$ java Mathew
17BIT00*0
9013304*08
Exception in thread "main" java.lang.NumberFormatException: Mobile Number cannot contain any character other than a digit
        at Mathew.validate(Mathew.java:22)
        at Mathew.main(Mathew.java:39)
Mathews-MacBook-Air:Desktop Mathew$ 
```