Antwalk Software Engineering Foundations

Abstract class, Interface, Inner classes, Exceptions

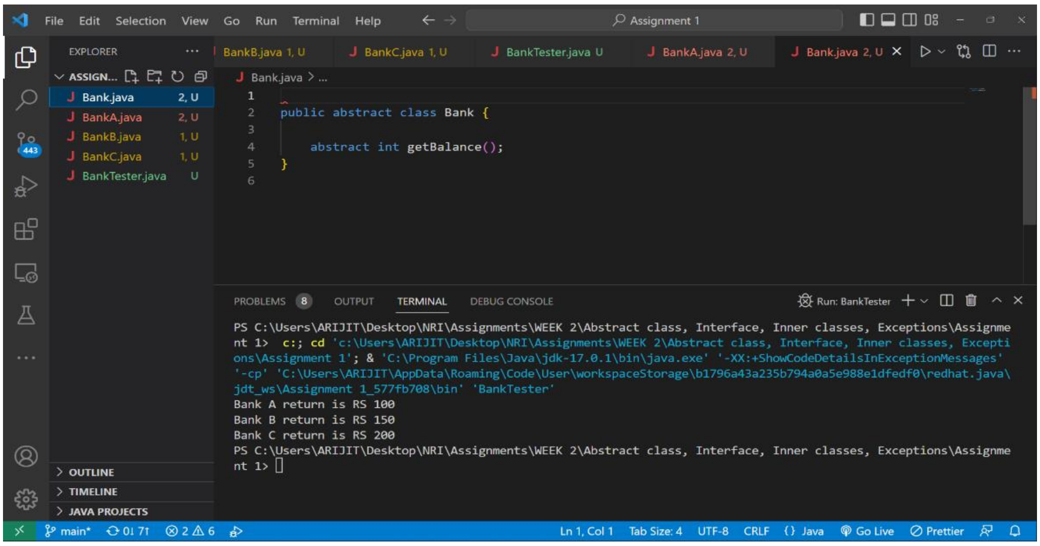
Assignment 1

Name : Arijit Saha , College : Future Institute of Engineering and Management , Employee Id : T23010330 , Mail Id: arijits@trainee.nrifintech.com

Q1. Abstract Class:

Create an abstract class 'Bank' with an abstract method 'getBalance'. $100, $150 and $200 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes.

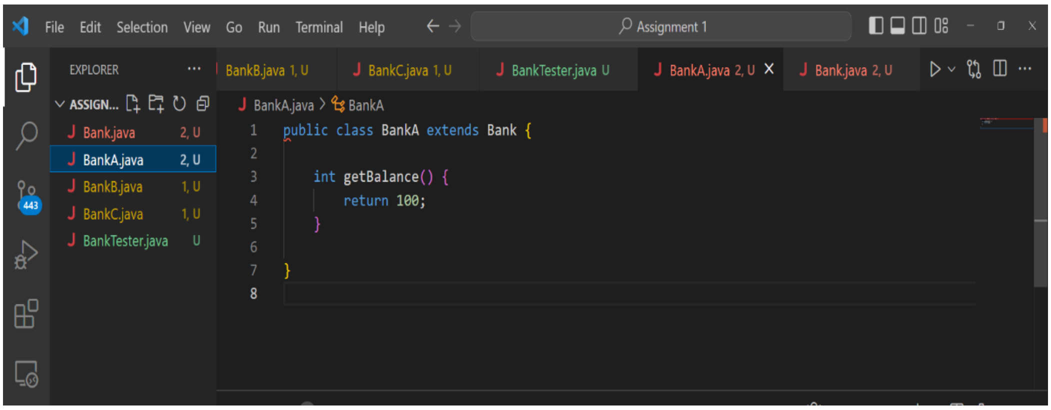
Bank.java



```java

public abstract class Bank {

    abstract int getBalance();
}

```

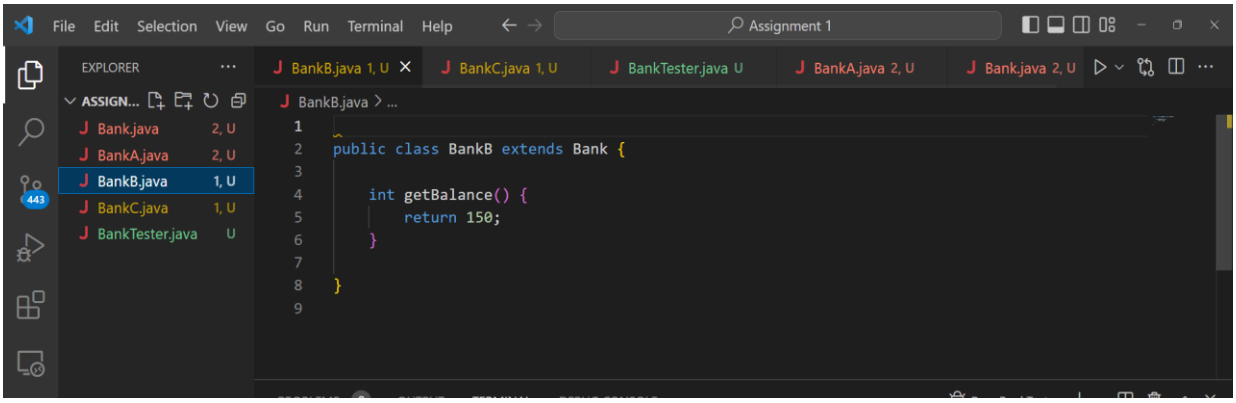
```
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Exceptions\Assignme
nt 1>  c:; cd 'c:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Excepti
ons\Assignment 1'; & 'C:\Program Files\Java\jdk-17.0.1\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages'
'-cp' 'C:\Users\ARIJIT\AppData\Roaming\Code\User\workspaceStorage\b1796a43a235b794a0a5e988e1dfedf0\redhat.java\
jdt_ws\Assignment 1_577fb708\bin' 'BankTester'
Bank A return is RS 100
Bank B return is RS 150
Bank C return is RS 200
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Exceptions\Assignme
nt 1> 
```

BankA.java



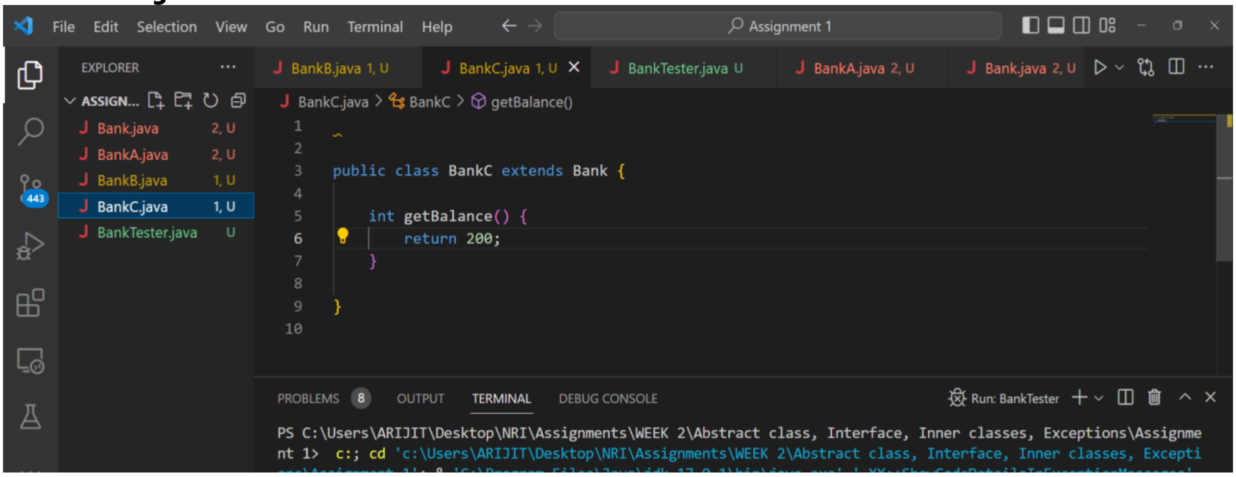
```java
public class BankA extends Bank {

    int getBalance() {
        return 100;
    }

}

```

BankB.java



```java

public class BankB extends Bank {

    int getBalance() {
        return 150;
    }

}

```

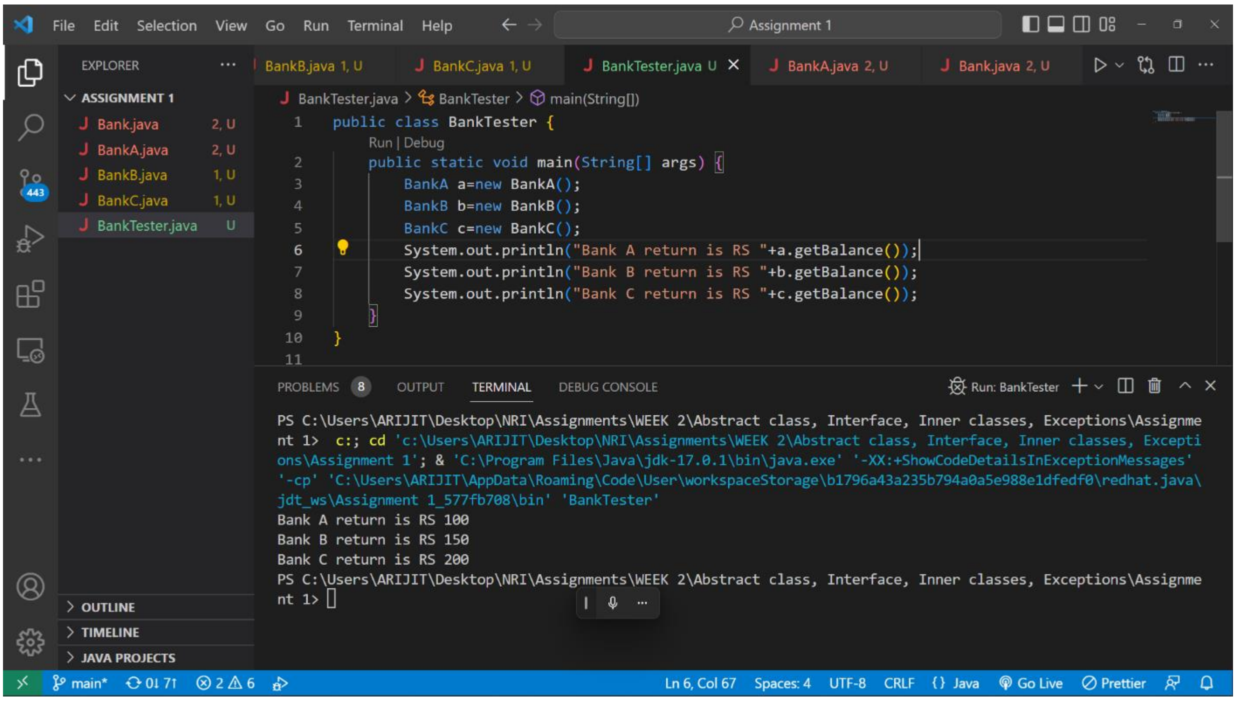
BankC.java



```java


public class BankC extends Bank {

    int getBalance() {
        return 200;
    }

}

```

```
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Exceptions\Assignme
nt 1>  c:; cd 'c:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Excepti
```

BankTester.java



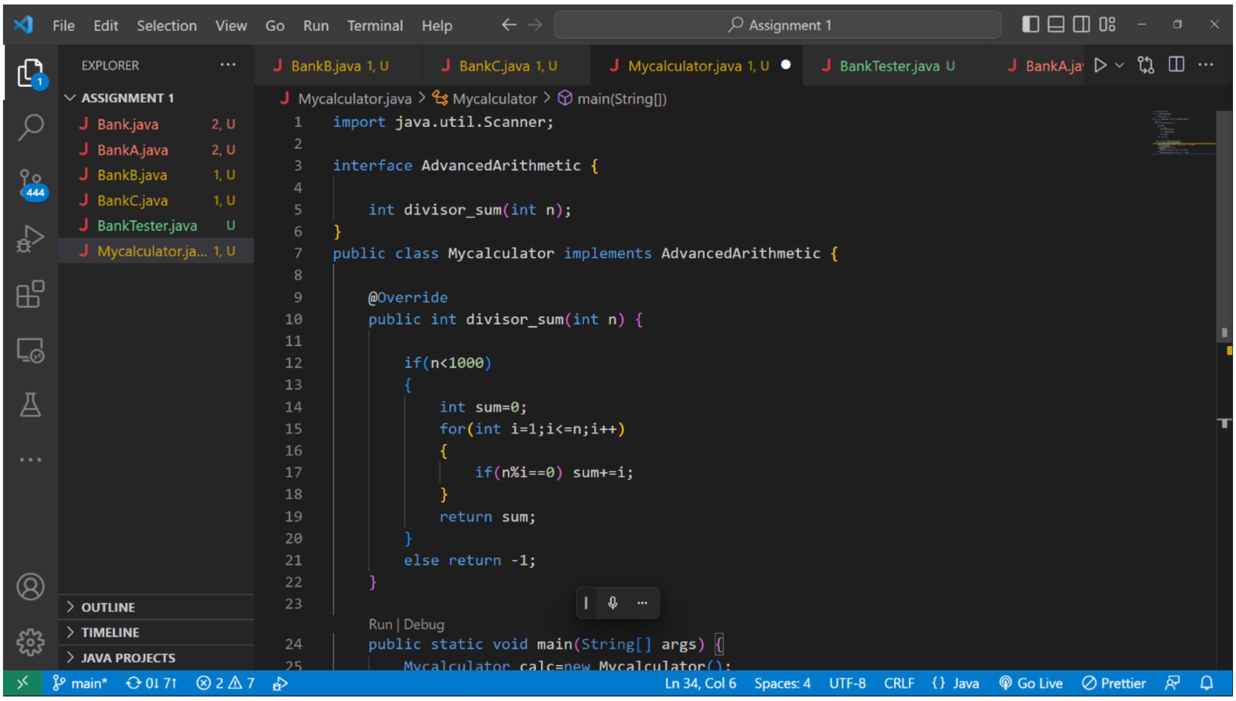
```java
public class BankTester {
    public static void main(String[] args) {
        BankA a=new BankA();
        BankB b=new BankB();
        BankC c=new BankC();
        System.out.println("Bank A return is RS "+a.getBalance());
        System.out.println("Bank B return is RS "+b.getBalance());
        System.out.println("Bank C return is RS "+c.getBalance());
    }
}

```

```
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Exceptions\Assignme
nt 1>  c:; cd 'c:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Excepti
ons\Assignment 1'; & 'C:\Program Files\Java\jdk-17.0.1\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages'
'-cp' 'C:\Users\ARIJIT\AppData\Roaming\Code\User\workspaceStorage\b1796a43a235b794a0a5e988e1dfedf0\redhat.java\
jdt_ws\Assignment 1_577fb708\bin' 'BankTester'
Bank A return is RS 100
Bank B return is RS 150
Bank C return is RS 200
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Exceptions\Assignme
nt 1> 
```

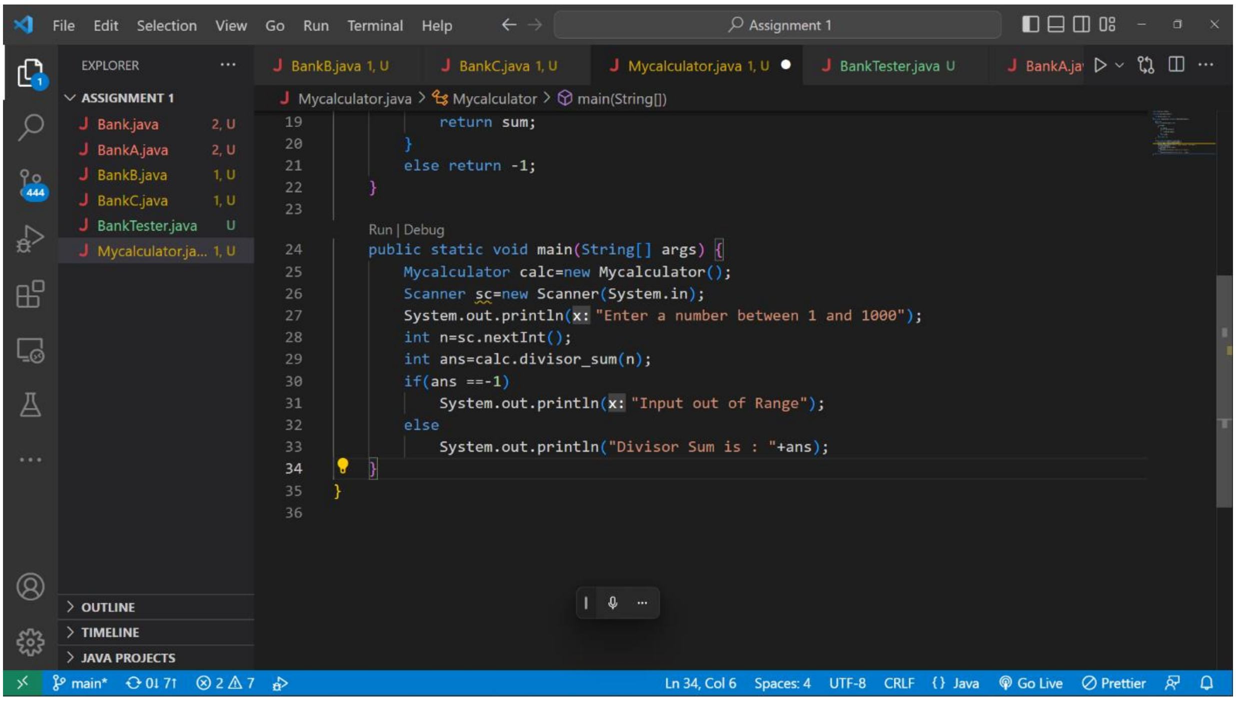
Q2. Interface :

Given an interface AdvancedArithmetic which contains a method signature int divisor_sum(int n). You need to write a class called MyCalculator which implements the interface.
divisor_sum function just takes an integer as input and return the sum of all its divisors. For example divisors of 6 are 1, 2, 3 and 6, so divisor_sum should return 12. The value of n will be at most 1000.

Mycalculator.java containing the AdvancedArithmetic Interface

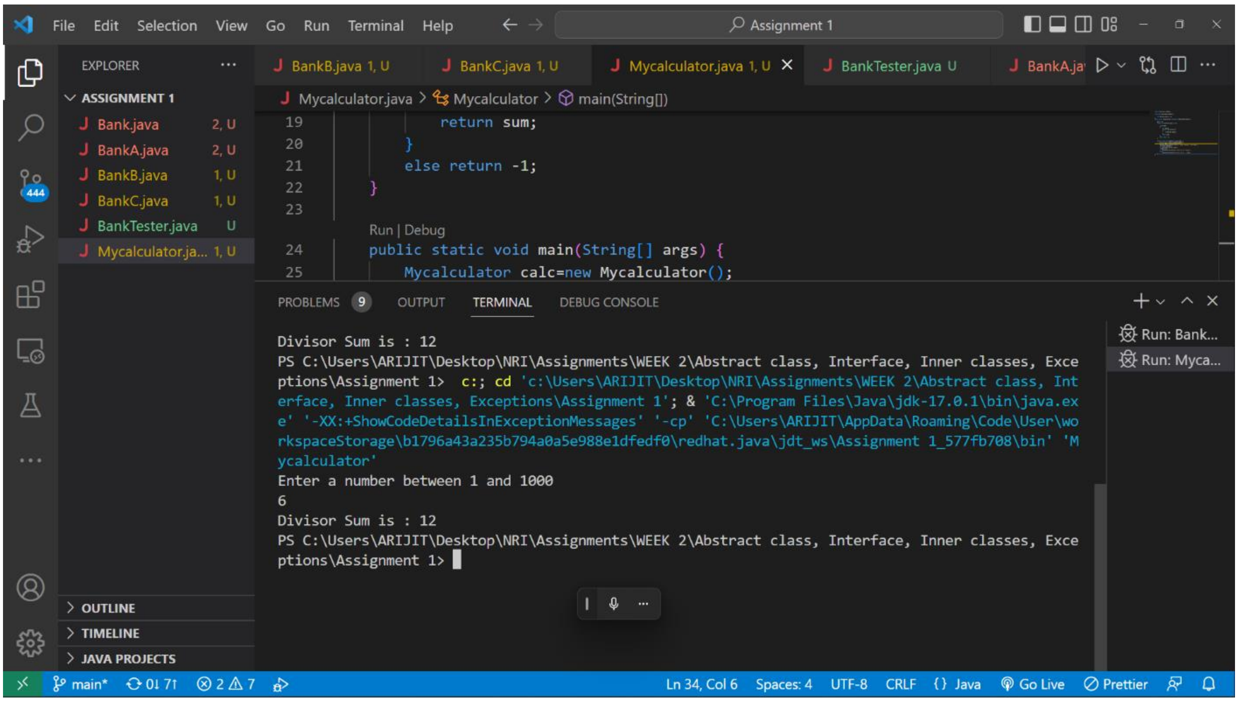


```java
import java.util.Scanner;

interface AdvancedArithmetic {

    int divisor_sum(int n);
}
public class Mycalculator implements AdvancedArithmetic {

    @Override
    public int divisor_sum(int n) {

        if(n<1000)
        {
            int sum=0;
            for(int i=1;i<=n;i++)
            {
                if(n%i==0) sum+=i;
            }
            return sum;
        }
        else return -1;
    }

    public static void main(String[] args) {
        Mycalculator calc=new Mycalculator();
```



```java
            return sum;
        }
        else return -1;
    }

    public static void main(String[] args) {
        Mycalculator calc=new Mycalculator();
        Scanner sc=new Scanner(System.in);
        System.out.println(x: "Enter a number between 1 and 1000");
        int n=sc.nextInt();
        int ans=calc.divisor_sum(n);
        if(ans ==-1)
            System.out.println(x: "Input out of Range");
        else
            System.out.println("Divisor Sum is : "+ans);
    }
}

```

OUTPUT:



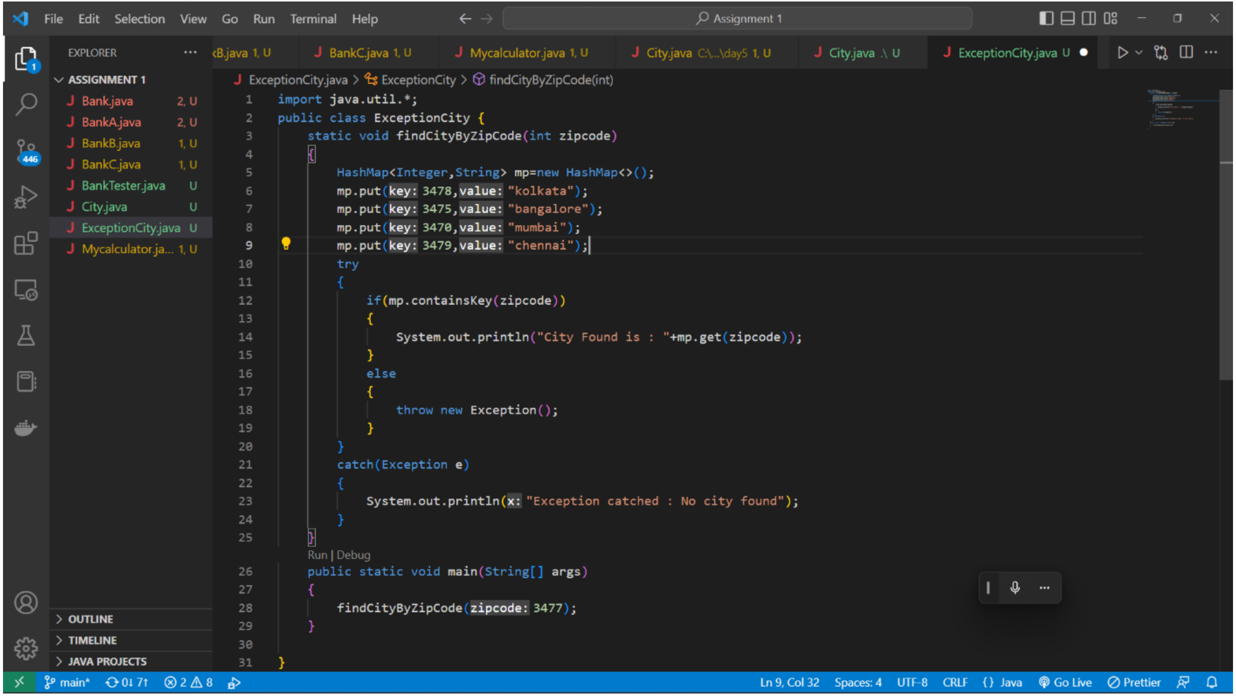
```
            return sum;
        }
        else return -1;
    }

    Run | Debug
    public static void main(String[] args) {
        Mycalculator calc=new Mycalculator();
```

```
Divisor Sum is : 12
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Exce
ptions\Assignment 1>  c:; cd 'c:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Int
erface, Inner classes, Exceptions\Assignment 1'; & 'C:\Program Files\Java\jdk-17.0.1\bin\java.ex
e' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ARIJIT\AppData\Roaming\Code\User\wo
rkspaceStorage\b1796a43a235b794a0a5e988e1dfedf0\redhat.java\jdt_ws\Assignment 1_577fb708\bin' 'M
ycalculator'
Enter a number between 1 and 1000
6
Divisor Sum is : 12
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Exce
ptions\Assignment 1> 
```

Q3. Exception handling :

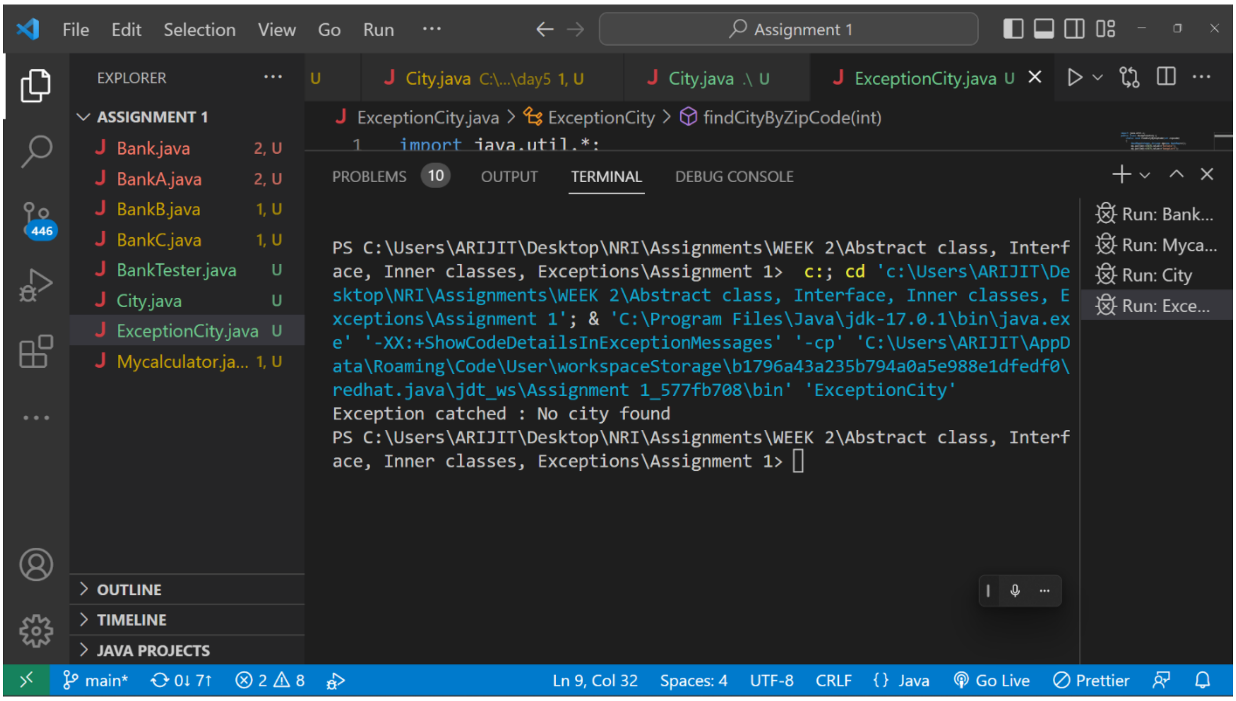
Create a City class that contains the list of the city with its zip code.
It contains findCityByZipCode(int zipCode) method. If it doesn't find city name, then it'll catch the exception "City not found".
Create a user defined Exception for the same.
add printStackTrace method for the exception details.

Catching Exception NoCityFound



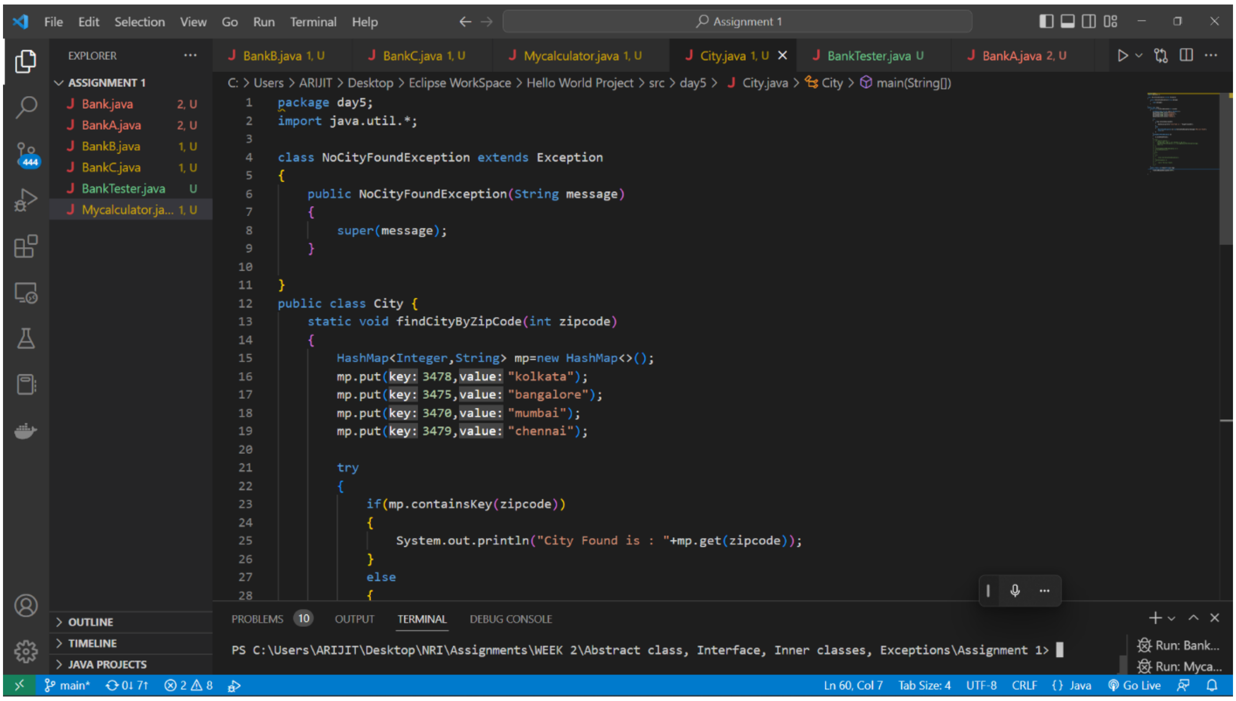
```
import java.util.*;
public class ExceptionCity {
    static void findCityByZipCode(int zipcode)
    {
        HashMap<Integer,String> mp=new HashMap<>();
        mp.put(key: 3478,value: "kolkata");
        mp.put(key: 3475,value: "bangalore");
        mp.put(key: 3470,value: "mumbai");
        mp.put(key: 3479,value: "chennai");
        try
        {
            if(mp.containsKey(zipcode))
            {
                System.out.println("City Found is : "+mp.get(zipcode));
            }
            else
            {
                throw new Exception();
            }
        }
        catch(Exception e)
        {
            System.out.println(x: "Exception catched : No city found");
        }
    }
    Run | Debug
    public static void main(String[] args)
    {
        findCityByZipCode(zipcode: 3477);
    }

}
```

Output:



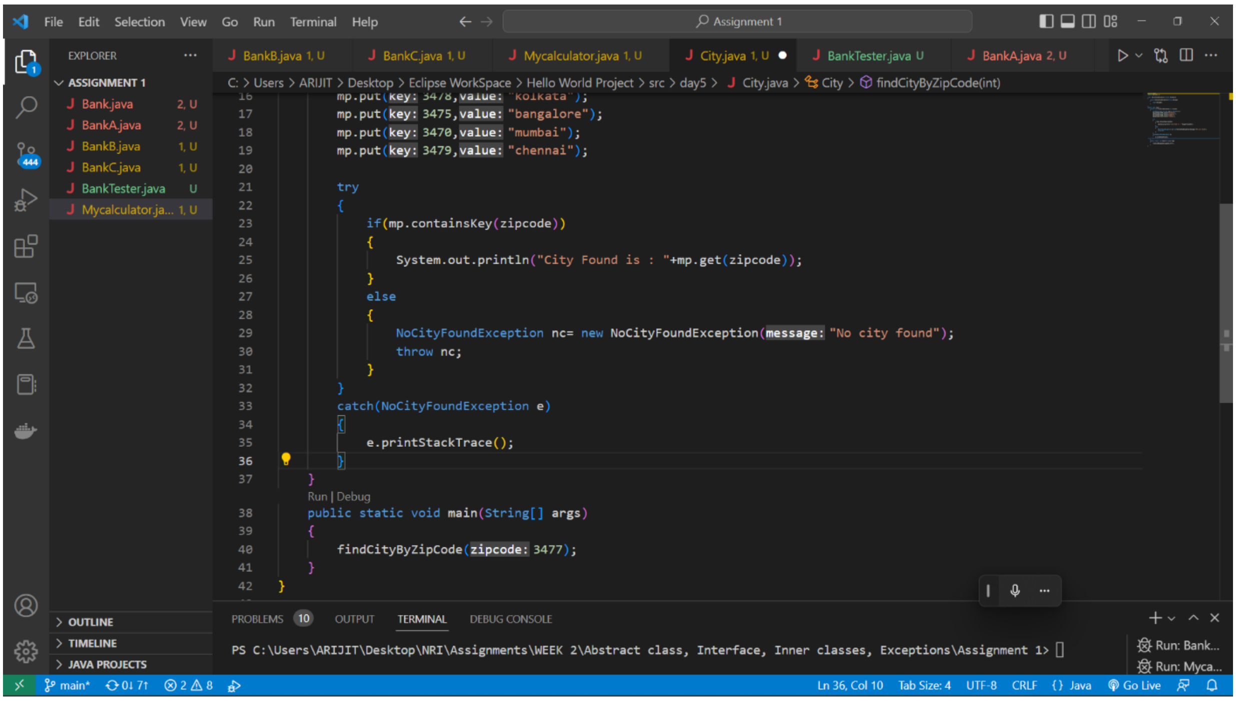
```
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interf
ace, Inner classes, Exceptions\Assignment 1>  c:; cd 'c:\Users\ARIJIT\De
sktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, E
xceptions\Assignment 1'; & 'C:\Program Files\Java\jdk-17.0.1\bin\java.ex
e' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ARIJIT\AppD
ata\Roaming\Code\User\workspaceStorage\b1796a43a235b794a0a5e988e1dfedf0\
redhat.java\jdt_ws\Assignment 1_577fb708\bin' 'ExceptionCity'
Exception catched : No city found
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interf
ace, Inner classes, Exceptions\Assignment 1>
```

Userdefined Exception with printStackTrace method



```java
package day5;
import java.util.*;

class NoCityFoundException extends Exception
{
    public NoCityFoundException(String message)
    {
        super(message);
    }

}
public class City {
    static void findCityByZipCode(int zipcode)
    {
        HashMap<Integer,String> mp=new HashMap<>();
        mp.put(key: 3478,value: "kolkata");
        mp.put(key: 3475,value: "bangalore");
        mp.put(key: 3470,value: "mumbai");
        mp.put(key: 3479,value: "chennai");

        try
        {
            if(mp.containsKey(zipcode))
            {
                System.out.println("City Found is : "+mp.get(zipcode));
            }
            else
            {
```

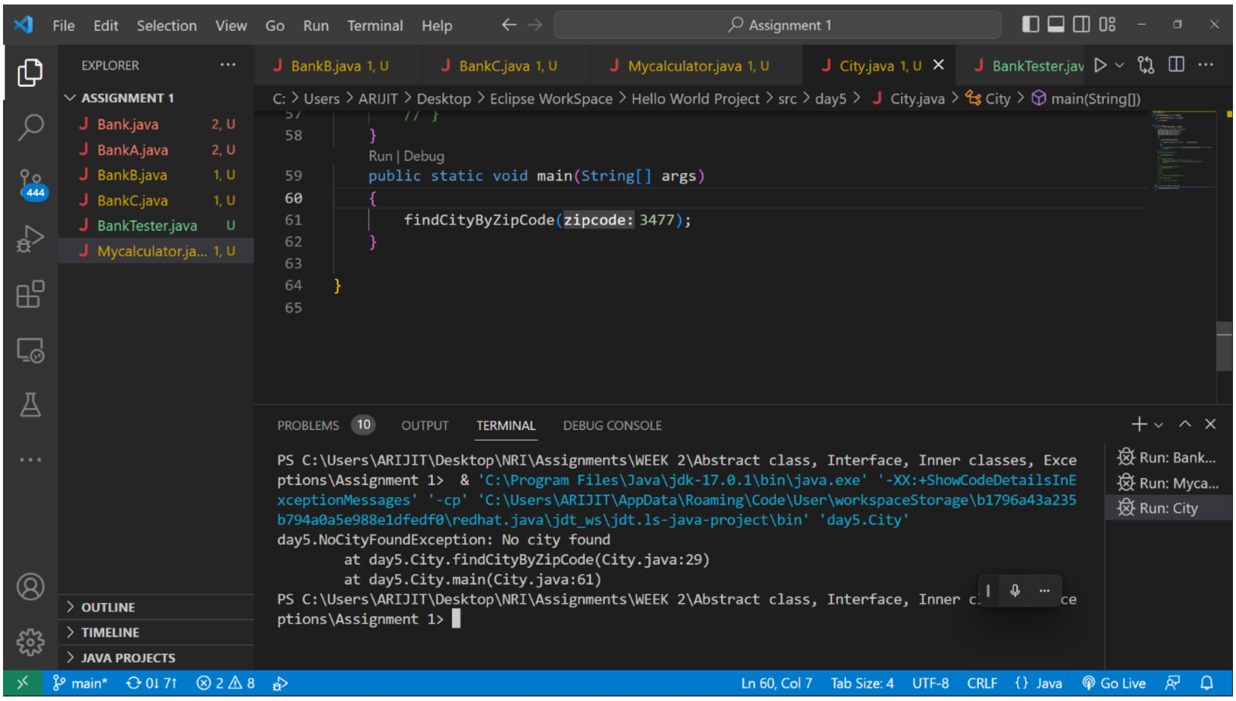
```
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Exceptions\Assignment 1>
```



```
        mp.put(key: 3478,value: "kolkata");
        mp.put(key: 3475,value: "bangalore");
        mp.put(key: 3470,value: "mumbai");
        mp.put(key: 3479,value: "chennai");

        try
        {
            if(mp.containsKey(zipcode))
            {
                System.out.println("City Found is : "+mp.get(zipcode));
            }
            else
            {
                NoCityFoundException nc= new NoCityFoundException(message: "No city found");
                throw nc;
            }
        }
        catch(NoCityFoundException e)
        {
            e.printStackTrace();
        }
    }
    Run | Debug
    public static void main(String[] args)
    {
        findCityByZipCode(zipcode: 3477);
    }
}
```

```
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Exceptions\Assignment 1> 
```

Output :



```
        // }
    }
    Run | Debug
    public static void main(String[] args)
    {
        findCityByZipCode(zipcode: 3477);
    }

}

```

```
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Exce
ptions\Assignment 1>  & 'C:\Program Files\Java\jdk-17.0.1\bin\java.exe' '-XX:+ShowCodeDetailsInE
xceptionMessages' '-cp' 'C:\Users\ARIJIT\AppData\Roaming\Code\User\workspaceStorage\b1796a43a235
b794a0a5e988e1dfedf0\redhat.java\jdt_ws\jdt.ls-java-project\bin' 'day5.City'
day5.NoCityFoundException: No city found
        at day5.City.findCityByZipCode(City.java:29)
        at day5.City.main(City.java:61)
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner c            ce
ptions\Assignment 1> 
```