

## Antwalk Software Engineering Foundations

### Abstract class, Interface, Inner classes, Exceptions

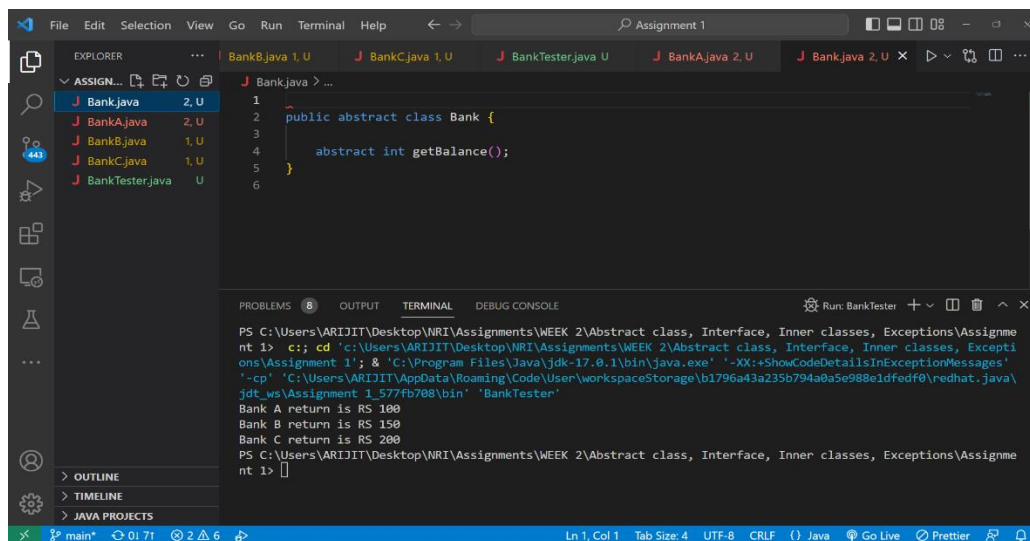
#### Assignment 1

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



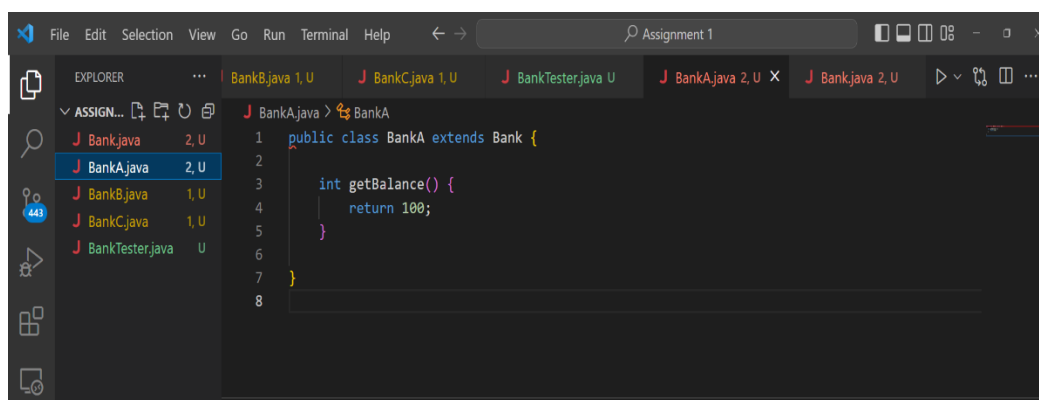
```
File Edit Selection View Go Run Terminal Help
Assignment 1

EXPLORER
  ASSIGN...
    Bank.java 2, U
    BankA.java 2, U
    BankB.java 1, U
    BankC.java 1, U
    BankTester.java U

Bank.java > ...
1
2 public abstract class Bank {
3
4     abstract int getBalance();
5
6 }

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
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

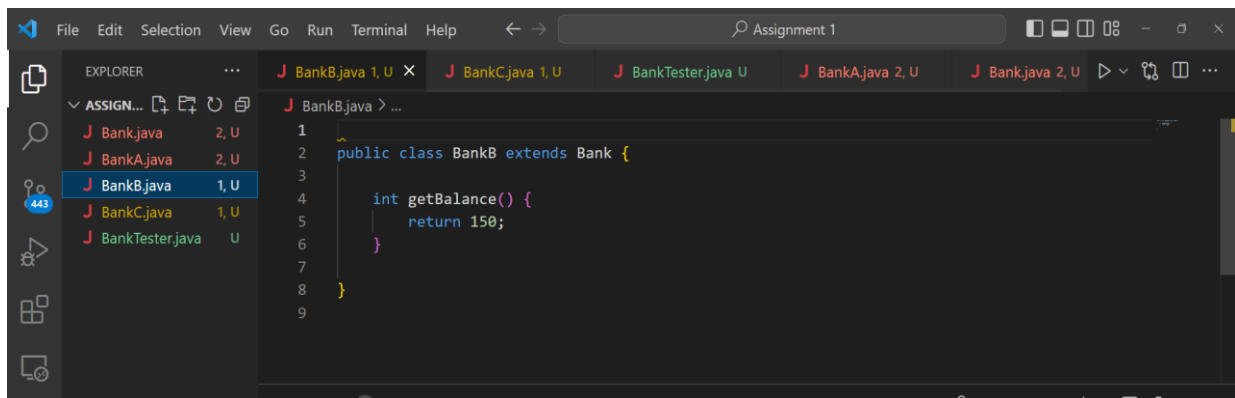


```
File Edit Selection View Go Run Terminal Help
Assignment 1

EXPLORER
  ASSIGN...
    Bank.java 2, U
    BankA.java 2, U
    BankB.java 1, U
    BankC.java 1, U
    BankTester.java U

BankA.java > BankA
1 public class BankA extends Bank {
2
3     int getBalance() {
4         return 100;
5     }
6
7 }
8
```

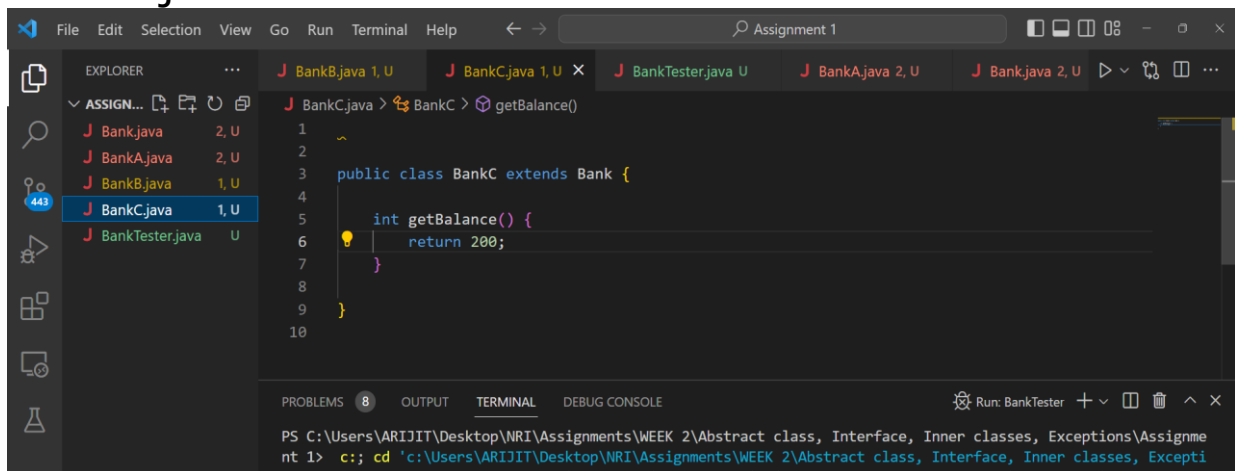
## BankB.java



The screenshot shows an IDE window titled "Assignment 1" with several tabs: BankB.java 1, U, BankC.java 1, U, BankTester.java U, BankA.java 2, U, and Bank.java 2, U. The Explorer panel on the left shows a project structure with files Bank.java, BankA.java, BankB.java, BankC.java, and BankTester.java. BankB.java is selected. The main editor displays the following code:

```
1 public class BankB extends Bank {
2
3
4     int getBalance() {
5         return 150;
6     }
7
8 }
9
```

## BankC.java



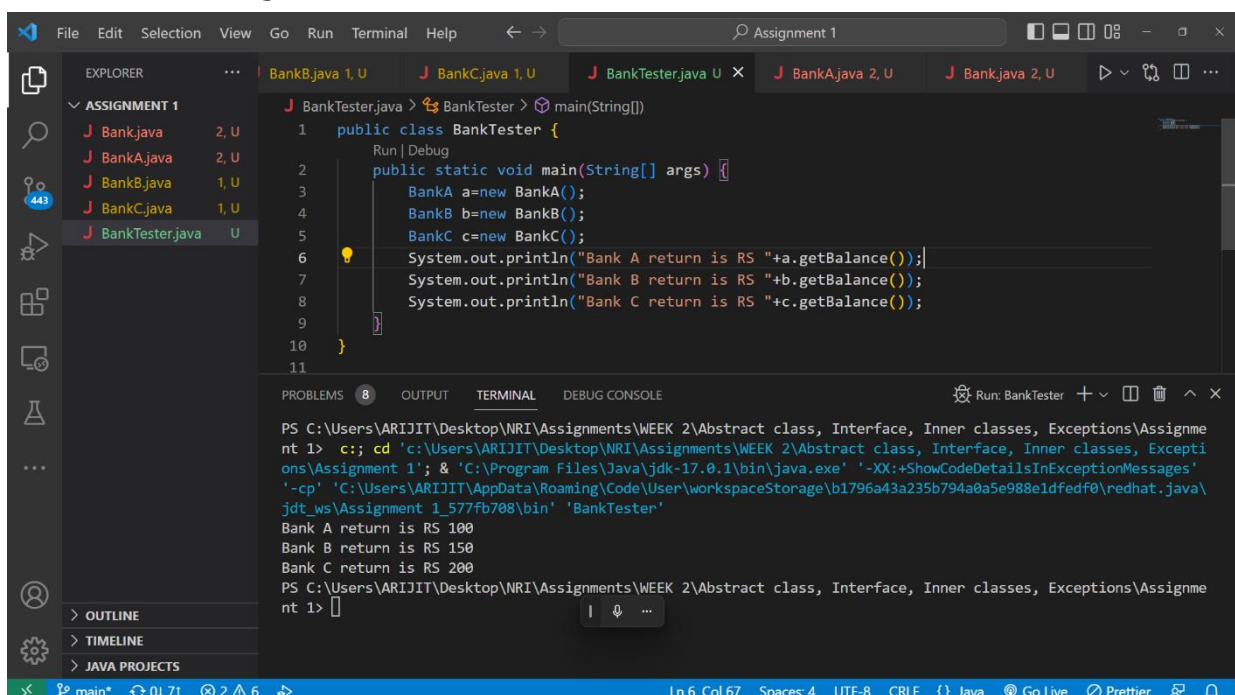
The screenshot shows the same IDE window with BankC.java selected. The Explorer panel shows BankC.java is selected. The main editor displays the following code:

```
1 public class BankC extends Bank {
2
3
4     int getBalance() {
5         return 200;
6     }
7
8 }
9
10
```

The bottom panel shows the terminal output:

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

## BankTester.java



The screenshot shows the same IDE window with BankTester.java selected. The Explorer panel shows BankTester.java is selected. The main editor displays the following code:

```
1 public class BankTester {
2
3     public static void main(String[] args) {
4         BankA a=new BankA();
5         BankB b=new BankB();
6         BankC c=new BankC();
7         System.out.println("Bank A return is RS "+a.getBalance());
8         System.out.println("Bank B return is RS "+b.getBalance());
9         System.out.println("Bank C return is RS "+c.getBalance());
10    }
11 }
```

The bottom panel shows the terminal output:

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

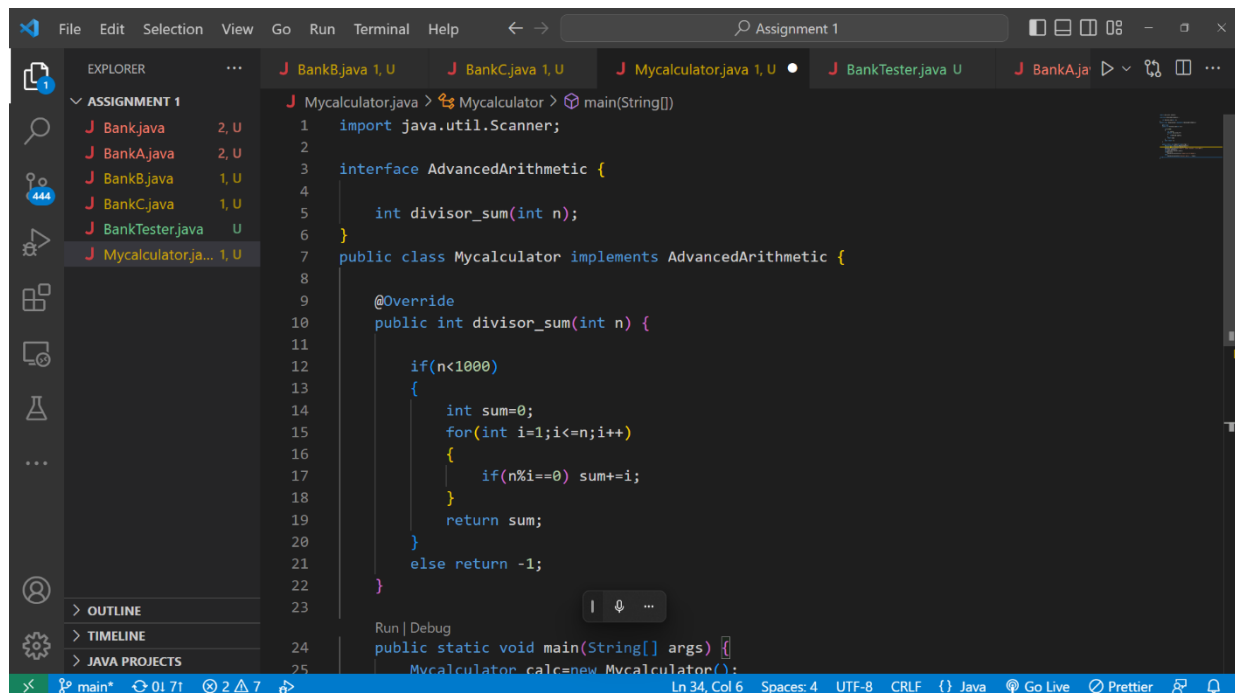
```
Bank A return is RS 100
Bank B return is RS 150
Bank C return is RS 200
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Exceptions\Assignment 1>
```

## 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.

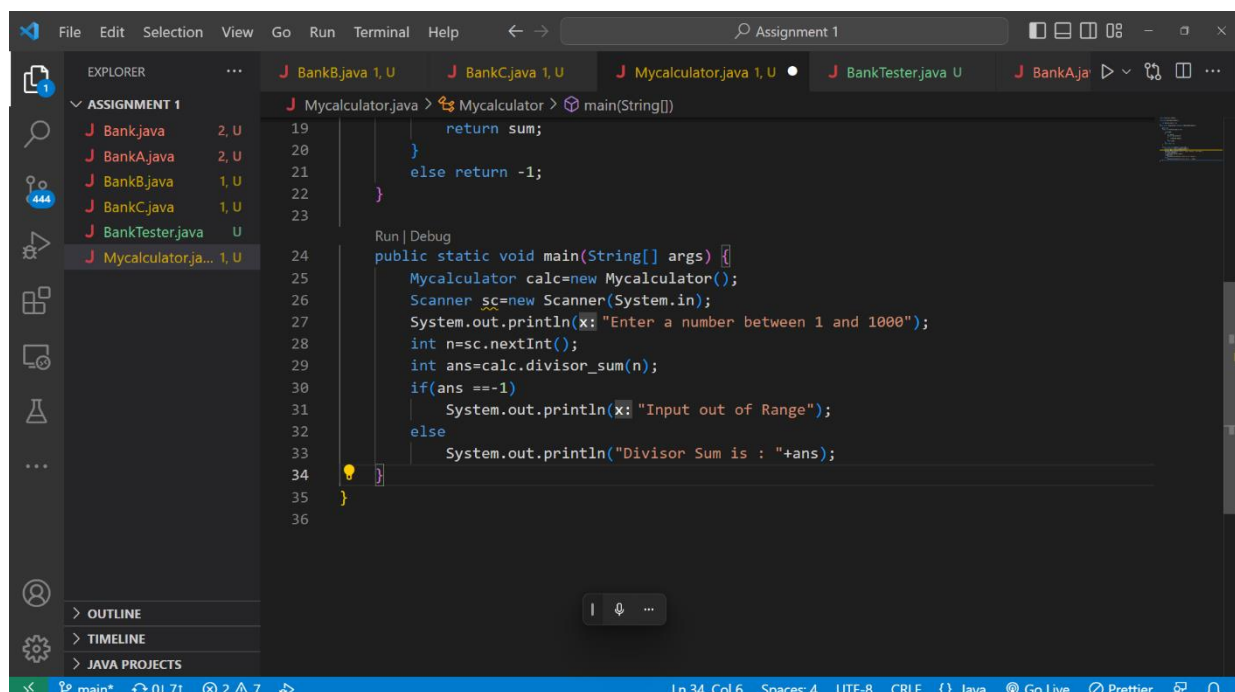
## Mycalculator.java containing the AdvancedArithmetic Interface



```
File Edit Selection View Go Run Terminal Help
Assignment 1

EXPLORER
ASSIGNMENT 1
  Bank.java 2, U
  BankA.java 2, U
  BankB.java 1, U
  BankC.java 1, U
  BankTester.java U
  Mycalculator.java 1, U

Mycalculator.java > Mycalculator > main(String[])
1  import java.util.Scanner;
2
3  interface AdvancedArithmetic {
4
5      int divisor_sum(int n);
6  }
7  public class Mycalculator implements AdvancedArithmetic {
8
9      @Override
10     public int divisor_sum(int n) {
11
12         if(n<1000)
13         {
14             int sum=0;
15             for(int i=1;i<=n;i++)
16             {
17                 if(n%i==0) sum+=i;
18             }
19             return sum;
20         }
21         else return -1;
22     }
23
24     Run | Debug
25     public static void main(String[] args) {
26         Mycalculator calc=new Mycalculator();
```

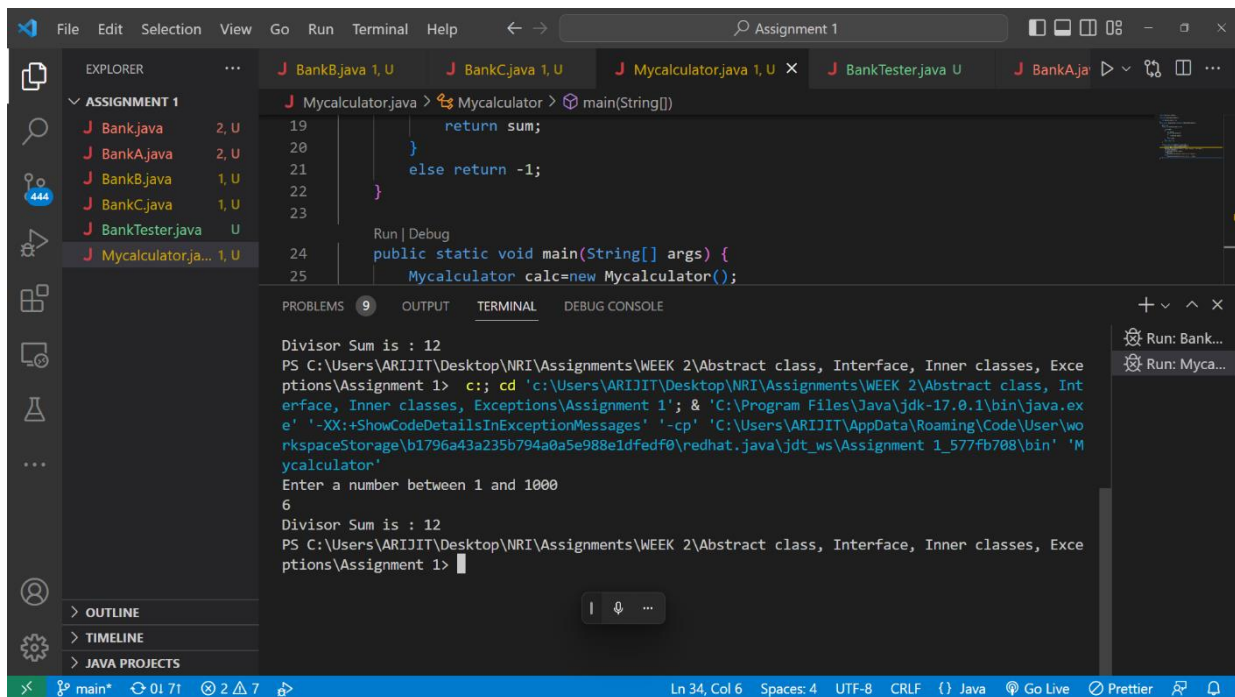


```
File Edit Selection View Go Run Terminal Help
Assignment 1

EXPLORER
ASSIGNMENT 1
  Bank.java 2, U
  BankA.java 2, U
  BankB.java 1, U
  BankC.java 1, U
  BankTester.java U
  Mycalculator.java 1, U

Mycalculator.java > Mycalculator > main(String[])
19     return sum;
20     }
21     else return -1;
22 }
23
24 Run | Debug
25 public static void main(String[] args) {
26     Mycalculator calc=new Mycalculator();
27     Scanner sc=new Scanner(System.in);
28     System.out.println(x: "Enter a number between 1 and 1000");
29     int n=sc.nextInt();
30     int ans=calc.divisor_sum(n);
31     if(ans ==-1)
32         System.out.println(x: "Input out of Range");
33     else
34         System.out.println("Divisor Sum is : "+ans);
35 }
36
```

## OUTPUT:

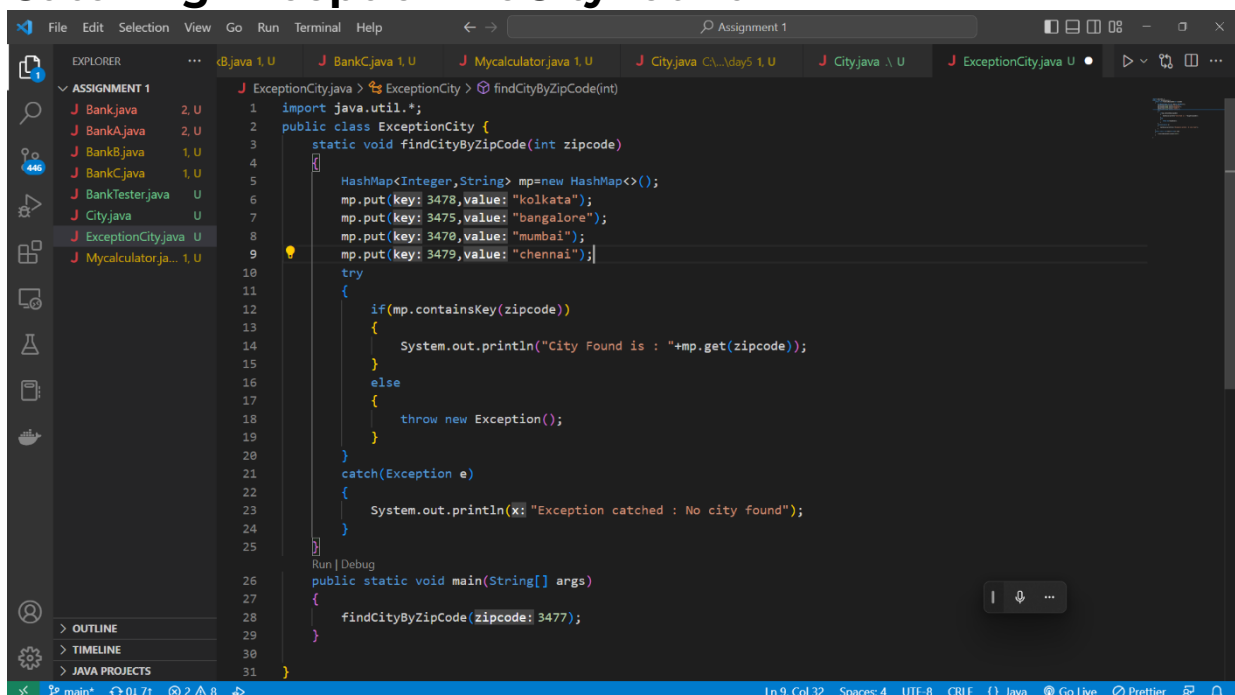


```
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Exceptions\Assignment 1> c:; cd 'c:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Exceptions\Assignment 1'; & 'C:\Program Files\Java\jdk-17.0.1\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ARIJIT\AppData\Roaming\Code\User\workspaceStorage\b1796a43a235b794a0a5e988e1dfef0\redhat.java\jdt_ws\Assignment_1_577fb708\bin' 'Mycalculator'
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, Exceptions\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

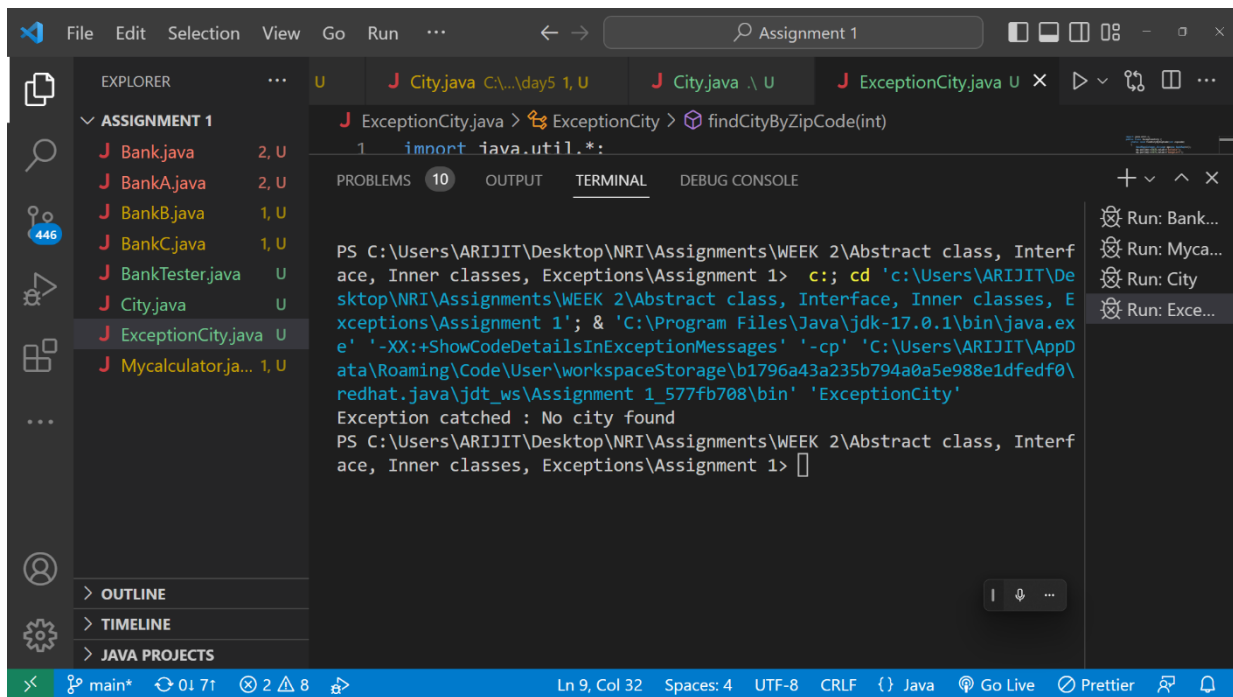


```
import java.util.*;

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 caught : No city found");
        }
    }

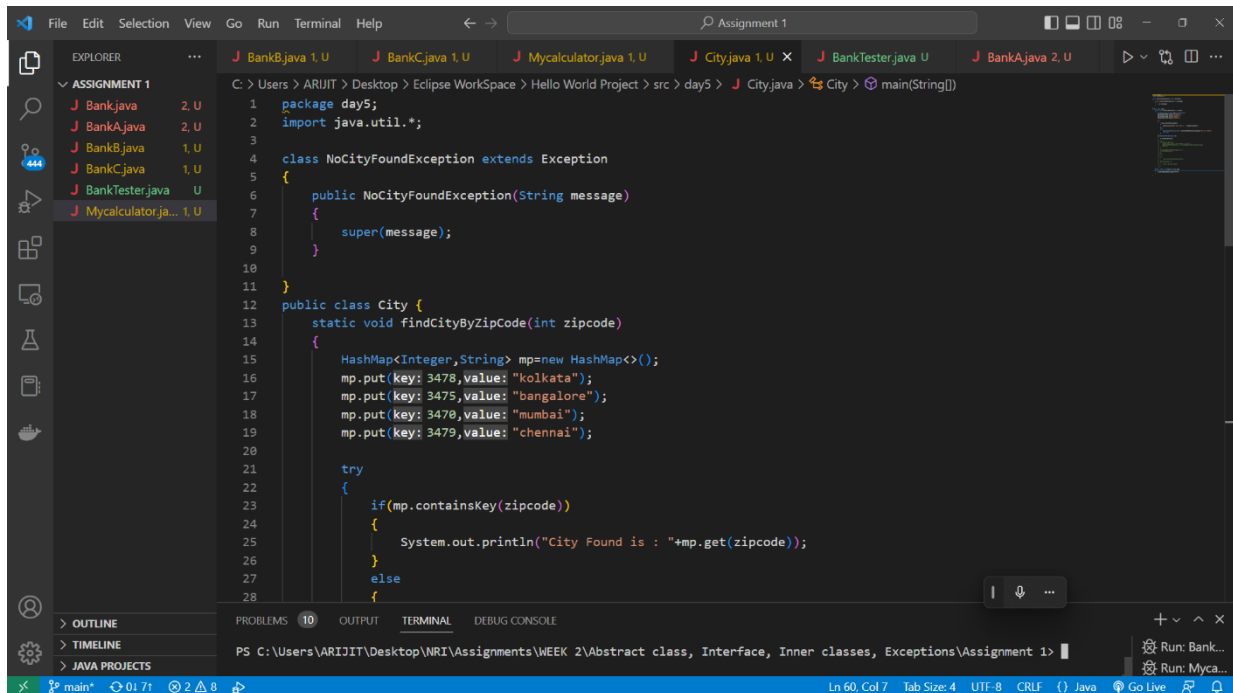
    public static void main(String[] args) {
        findCityByZipCode(zipcode: 3477);
    }
}
```

## Output:



```
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Exceptions\Assignment 1> c:; cd 'c:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Exceptions\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' 'ExceptionCity'
Exception caught : No city found
PS C:\Users\ARIJIT\Desktop\NRI\Assignments\WEEK 2\Abstract class, Interface, Inner classes, Exceptions\Assignment 1>
```

## Userdefined Exception with printStackTrace method



```
C: > Users > ARIJIT > Desktop > Eclipse Workspace > Hello World Project > src > day5 > J City.java > City > main(String[])
1 package day5;
2 import java.util.*;
3
4 class NoCityFoundException extends Exception
5 {
6     public NoCityFoundException(String message)
7     {
8         super(message);
9     }
10 }
11
12 public class City {
13     static void findCityByZipCode(int zipcode)
14     {
15         HashMap<Integer,String> mp=new HashMap<>();
16         mp.put(key: 3478,value: "kolkata");
17         mp.put(key: 3475,value: "bangalore");
18         mp.put(key: 3470,value: "mumbai");
19         mp.put(key: 3479,value: "chennai");
20
21         try
22         {
23             if(mp.containsKey(zipcode))
24             {
25                 System.out.println("City Found is : "+mp.get(zipcode));
26             }
27             else
28             {
29                 throw new NoCityFoundException("City not found");
30             }
31         }
32     }
33 }
```

The screenshot shows the Eclipse IDE with the following details:

- Explorer:** A project named "ASSIGNMENT 1" containing files: Bank.java (2, U), BankA.java (2, U), BankB.java (1, U), BankC.java (1, U), BankTester.java (U), and Mycalculator.java (1, U).
- Editor:** The file `City.java` is open, showing the `findCityByZipCode(int)` method. The code uses a `Map` to store city names by zip code and throws a `NoCityFoundException` if the city is not found. The `main` method calls `findCityByZipCode(3477)`.
- Terminal:** The command prompt shows the execution of the program, resulting in the output: `City Found is : bangalore`.

## Output :

The screenshot shows the Eclipse IDE with the following details:

- Explorer:** The same project structure as the first screenshot.
- Editor:** The file `City.java` is open, showing the `main` method. The code calls `findCityByZipCode(3477)`.
- Terminal:** The command prompt shows the execution of the program, resulting in the output: `City Found is : bangalore`.