Tuhin Mukherjee

CSE, class roll: 37

3rd year

University roll: 17600119047

PCA2, 2021-22

Assignment

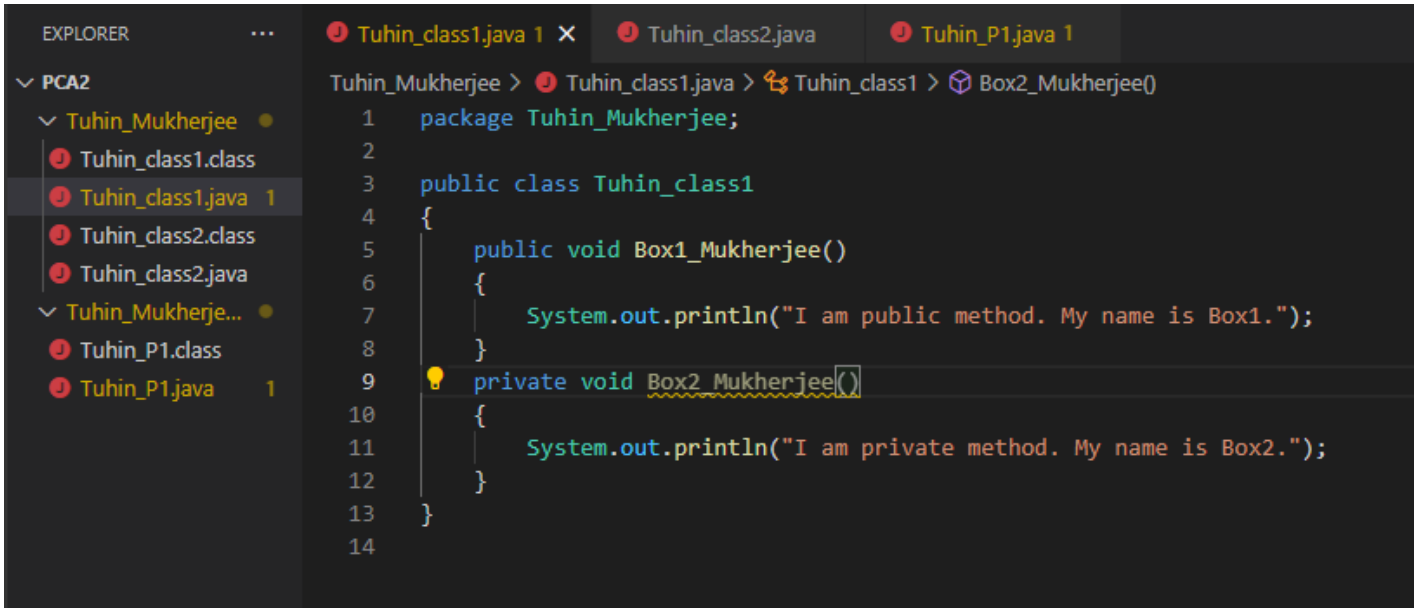
Subject: OOP

1. Create two classes under a package. In one class, define members with public and private access specifiers. In the other class, define members with protected and default access specifiers. Import the classes in another package, and do necessary implementations to show the use of those two classes and their members.
Package name should be in the form 'YournameYoursurname'.

INPUT:

(Screenshot)

Creating class Tuhin_class1 class within Tuhin_Mukherjee package containing two methods, one is public method and another is private method.



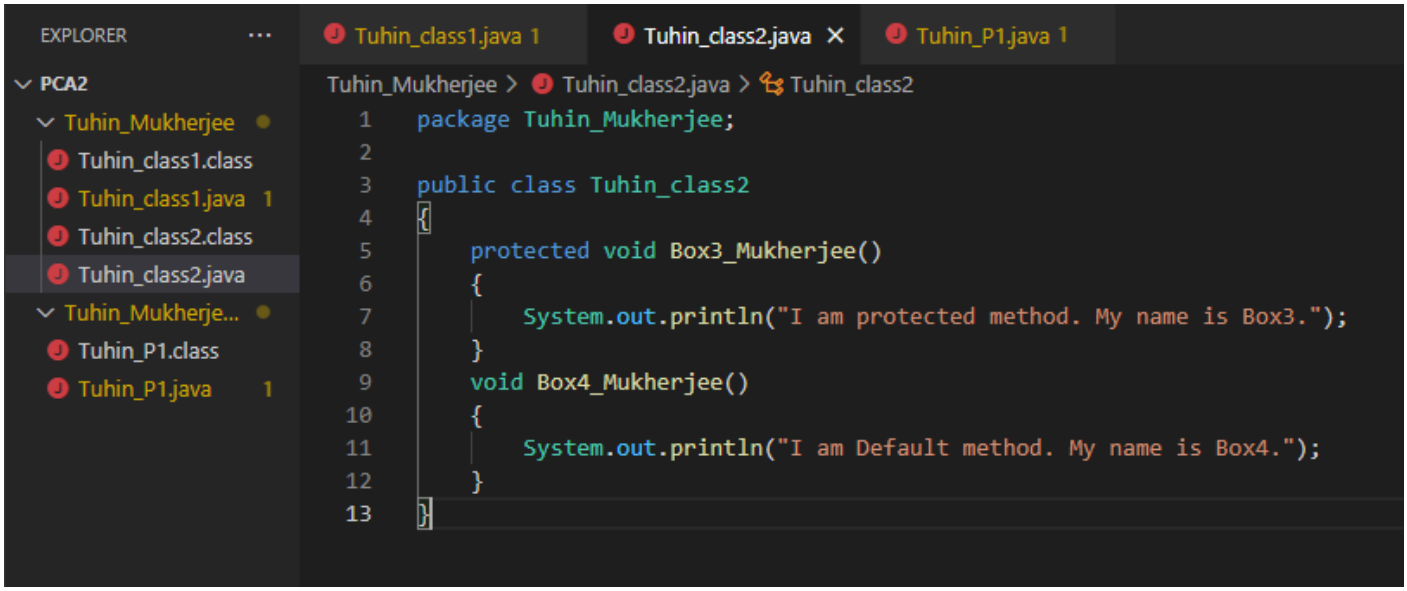
Code:

```
package Tuhin_Mukherjee;

public class Tuhin_class1
{
    public void Box1_Mukherjee()
    {
        System.out.println("I am public method. My name is Box1.");
    }
    private void Box2_Mukherjee()
    {
        System.out.println("I am private method. My name is Box2.");
    }
}
```

(Screenshot)

Creating another class namely Tuhin_class2 class within Tuhin_Mukherjee package containing two methods, one is protected method and another is default method.



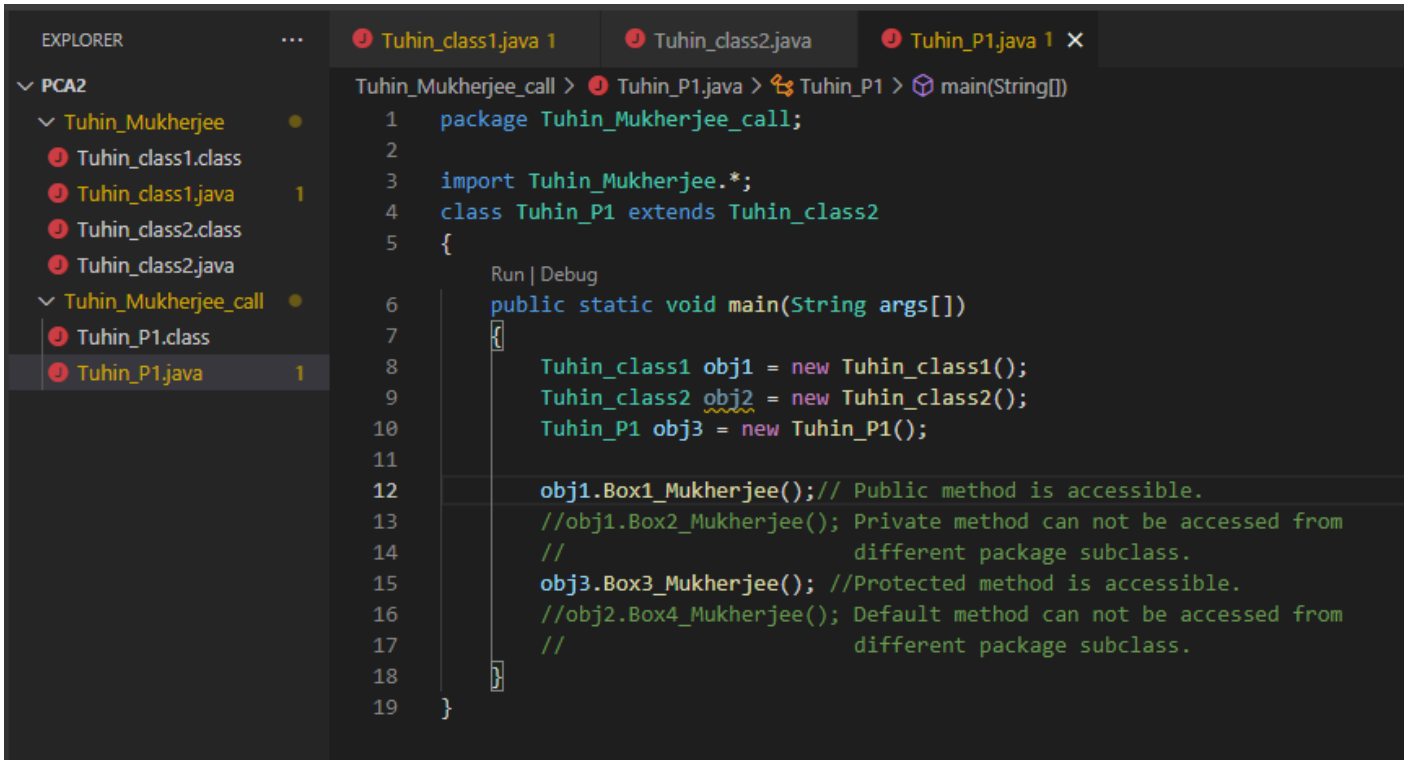
Code:

```
package Tuhin_Mukherjee;

public class Tuhin_class2
{
    protected void Box3_Mukherjee()
    {
        System.out.println("I am protected method. My name is Box3.");
    }
    void Box4_Mukherjee()
    {
        System.out.println("I am Default method. My name is Box4.");
    }
}
```

(Screenshot)

Creating Tuhin_P1 as the main class within Tuhin_Mukherjee_call package containing all the aforesaid methods.



In-case of Public method:

Code:

```
package Tuhin_Mukherjee_call;


import Tuhin_Mukherjee.*;
class Tuhin_P1 extends Tuhin_class2
{
    public static void main(String args[])
    {
        Tuhin_class1 obj1 = new Tuhin_class1();
        Tuhin_class2 obj2 = new Tuhin_class2();
        Tuhin_P1 obj3 = new Tuhin_P1();

        obj1.Box1_Mukherjee();// Public method is accessible anywhere.
        //obj1.Box2_Mukherjee(); Private method can not be accessed from
        //                       different package subclass.
        //obj3.Box3_Mukherjee(); //Protected method is accessible.
        //obj2.Box4_Mukherjee(); Default method can not be accessed from
        //                       different package subclass.
    }
}
```

OUTPUT:

```
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2> java Tuhin_Mukherjee_call.Tuhin_P1
I am public method. My name is Box1.
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2>
```

In-Case of Private method:

Code:

```
package Tuhin_Mukherjee_call;

import Tuhin_Mukherjee.*;
class Tuhin_P1 extends Tuhin_class2
{
    public static void main(String args[])
    {
        Tuhin_class1 obj1 = new Tuhin_class1();
        Tuhin_class2 obj2 = new Tuhin_class2();
        Tuhin_P1 obj3 = new Tuhin_P1();

        //obj1.Box1_Mukherjee();// Public method is accessible anywhere.
        obj1.Box2_Mukherjee(); //  Private method can not be accessed from
        //                        different package subclass.
        //obj3.Box3_Mukherjee(); //Protected method is accessible.
        //obj2.Box4_Mukherjee(); Default method can not be accessed from
        //                      different package subclass.
    }
}
```

Output:

```
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2> javac Tuhin_Mukherjee_call\Tuhin_P1.java
Tuhin_Mukherjee_call\Tuhin_P1.java:13: error: Box2_Mukherjee() has private access in Tuhin_class1
        obj1.Box2_Mukherjee(); //  Private method can not be accessed from
            ^
1 error
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2>
```

In-Case of Protected Method:

Code:

```
package Tuhin_Mukherjee_call;

import Tuhin_Mukherjee.*;
class Tuhin_P1 extends Tuhin_class2
{
    public static void main(String args[])
    {
        Tuhin_class1 obj1 = new Tuhin_class1();
        Tuhin_class2 obj2 = new Tuhin_class2();
        Tuhin_P1 obj3 = new Tuhin_P1();

        //obj1.Box1_Mukherjee();//   Public method is accessible anywhere.
        //obj1.Box2_Mukherjee(); //  Private method can not be accessed from
        //                          different package subclass.
        obj3.Box3_Mukherjee(); //    Protected method can be accessed through subclass.
        //obj2.Box4_Mukherjee();     Default method can not be accessed from
        //                          different package subclass.
    }
}
```

Output:

```
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2> javac Tuhin_Mukherjee_call\Tuhin_P1.java
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2> java Tuhin_Mukherjee_call.Tuhin_P1
I am protected method. My name is Box3.
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2>
```

In-Case of Private Method:

Code:

```
package Tuhin_Mukherjee_call;

import Tuhin_Mukherjee.*;
class Tuhin_P1 extends Tuhin_class2
{
    public static void main(String args[])
    {
        Tuhin_class1 obj1 = new Tuhin_class1();
        Tuhin_class2 obj2 = new Tuhin_class2();
        Tuhin_P1 obj3 = new Tuhin_P1();

        //obj1.Box1_Mukherjee();//   Public method is accessible anywhere.
        //obj1.Box2_Mukherjee(); //  Private method can not be accessed from
        //                          different package subclass.
        //obj3.Box3_Mukherjee(); //  Protected method can be accessed through subclass.
        obj2.Box4_Mukherjee();     //Default method can not be accessed from
        //                          different package subclass.
    }
}
```

Output:

```
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2> javac Tuhin_Mukherjee_call\Tuhin_P1.java
Tuhin_Mukherjee_call\Tuhin_P1.java:16: error: Box4_Mukherjee() is not public in Tuhin_class2; cannot be accessed from outside package
        obj2.Box4_Mukherjee();     //Default method can not be accessed from
            ^
1 error
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2>
```

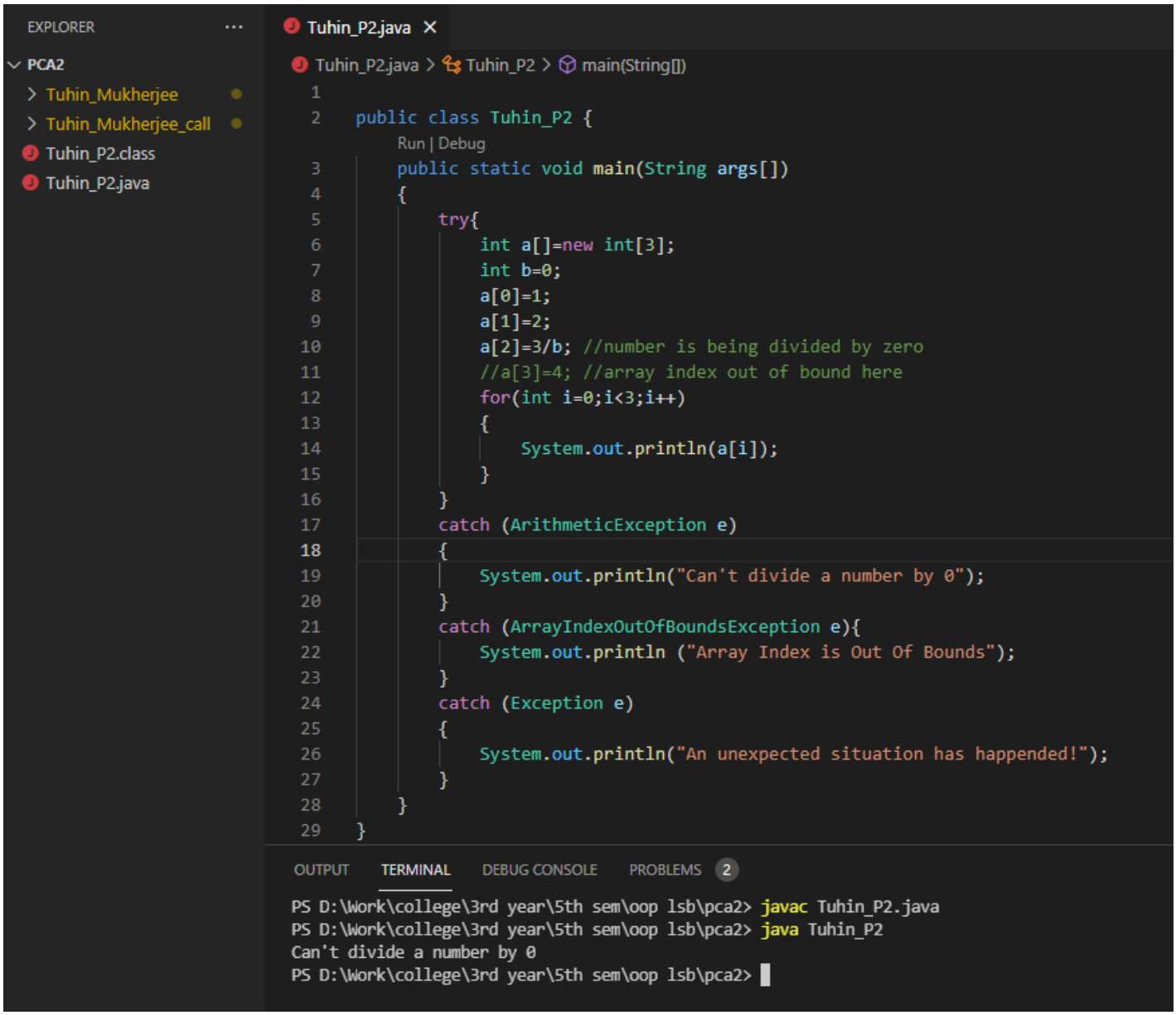
	default	private	protected	public
Same Class	Yes	Yes	Yes	Yes
Same package subclass	Yes	No	Yes	Yes
Same package non-subclass	Yes	No	Yes	Yes
Different package subclass	No	No	Yes	Yes
Different package non-subclass	No	No	No	Yes

2. Write a program to create and handle ArithmeticException and ArrayIndexOutOfBoundsException using a single *try* block. Show also the use of 'Exception' class in handling exceptions.

INPUT:

In-Case of ArithmeticException:

(Screenshot)



Code:

```
public class Tuhin_P2 {
    public static void main(String args[])
    {
        try{
            int a[]=new int[3];
            int b=0;
            a[0]=1;
            a[1]=2;
            a[2]=3/b; //number is being divided by zero
            //a[3]=4; //array index out of bound here
            for(int i=0;i<3;i++)
            {
```

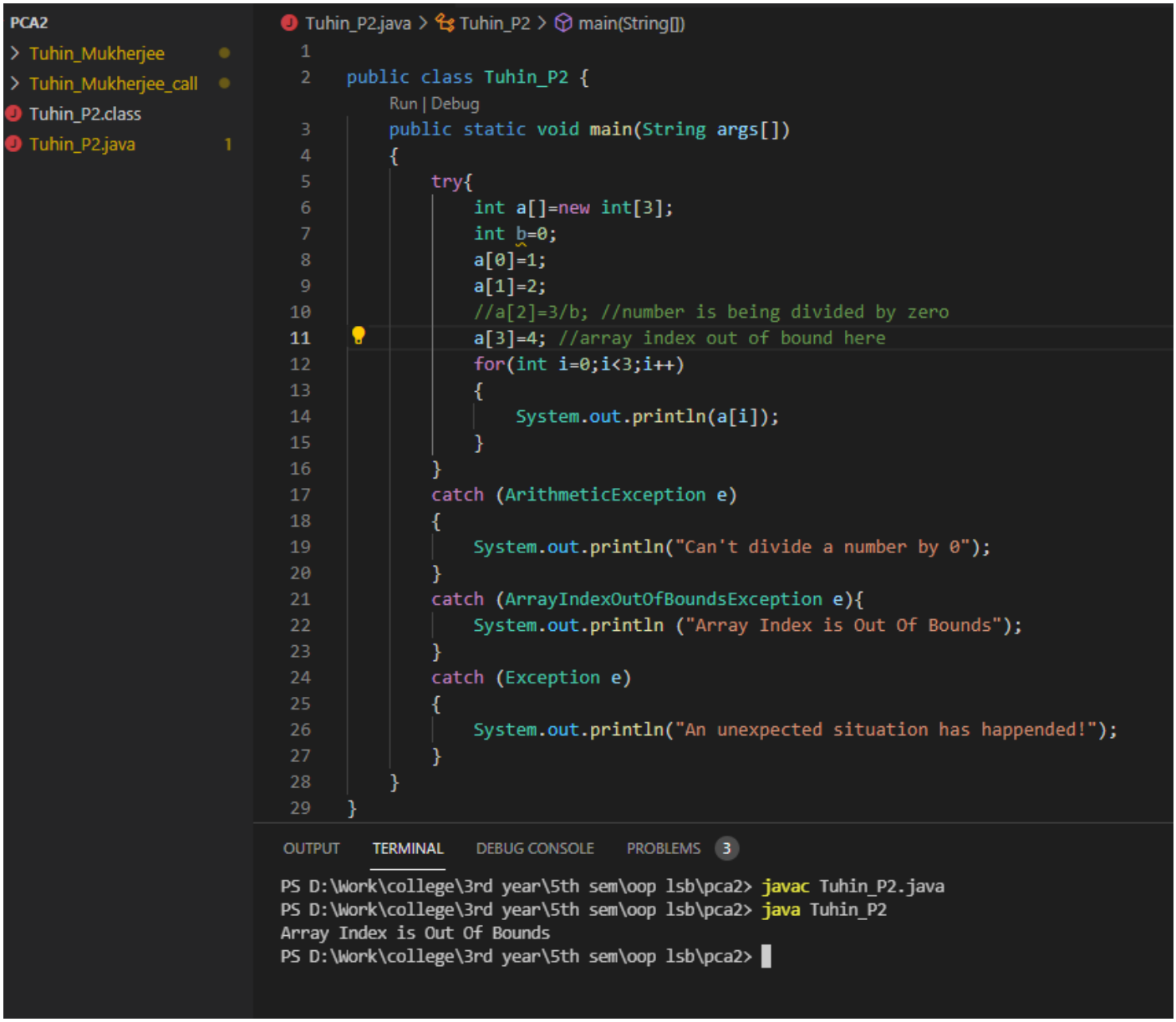
```
                System.out.println(a[i]);
            }
        }
        catch (ArithmeticException e)
        {
            System.out.println("Can't divide a number by 0");
        }
        catch (ArrayIndexOutOfBoundsException e){
            System.out.println ("Array Index is Out Of Bounds");
        }
        catch (Exception e)
        {
            System.out.println("An unexpected situation has happended!");
        }
    }
}
```

Output:

```
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2> javac Tuhin_P2.java
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2> java Tuhin_P2
Can't divide a number by 0
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2>
```

(Screenshot)

In-Case of ArrayIndexOutOfBoundsException:



Code:

```
public class Tuhin_P2 {
    public static void main(String args[])
    {
        try{
            int a[]=new int[3];
            int b=0;
            a[0]=1;
            a[1]=2;
            //a[2]=3/b; //number is being divided by zero
            a[3]=4; //array index out of bound here
            for(int i=0;i<3;i++)
            {
                System.out.println(a[i]);
            }
        }
```

```
        catch (ArithmeticException e)
        {
            System.out.println("Can't divide a number by 0");
        }
        catch (ArrayIndexOutOfBoundsException e){
            System.out.println ("Array Index is Out Of Bounds");
        }
        catch (Exception e)
        {
            System.out.println("An unexpected situation has happended!");
        }
    }
}
```

Output:

```
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2> javac Tuhin_P2.java
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2> java Tuhin_P2
Array Index is Out Of Bounds
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2>
```

3. **Numerology is an ancient study that draws meaning from different numbers, number combinations, letters, and symbols in your life. This art can help us tap into the underlying patterns of the universe and reveal new truths about who we are and what relationship we have.**

The word "FLAMES" has length 6, is used to determine the relationship between two names, where F stands for Friend, L for Love, A for Affection, M for Marriage, E for Enemy, and S for Serious.

a) **Accept two names as input and then remove the letters which are common in two. Now count the length of each name. Combine these two lengths. Perform Mod operation between two lengths. (combined length and 'FLAMES' length) .Search the position of the value obtained after Mod operation in FLAMES. .Finally, draw the relationship among these two names based on position and above-stated criterion.**

b) Perform the bitwise XOR operation on those two names after removal of common letters. If you get all one (1) or zero (0) then you find a perfect match

c) Perform necessary exception handling to see that there is no presence of 'punctuations, special characters, or numbers '

d) Output/Result: Screenshot of each part/sectio(a,b,and c) is required..

Example 1: Ranbir and Deepika

After removal of common letters i.e. 'a' and 'i' the number of remaining letters is 9 [Rnbr + Deepk]

Remaining value after Mod operation is 9%6 =3. Now the position of 3 in "FLAMES" is 'M' i.e. Marriage. [0 based index]

Example 2: Ajay and Kajal

After removal of common letters i.e. 'A', 'a','j', the number of remaining letters is 3.

Mod value 3%6=3. Now the position of 3 in "FLAMES" is 'M' i.e. Marriage. [0 based index]

Full Code of the problem:

```
import java.util.*;
import java.util.regex.Pattern;

class StringChecking extends Exception
{
    StringChecking(String message)
    {
        super(message);
    }
}

public class Tuhin_p3 {

    /**Part C Solution */

    /**Dealing with Exceptions in the User Entered String*/
    void Checking_String_Mukherjee(String name1, String name2)
    {
        Pattern Exception_Found = Pattern.compile("[^a-zA-Z ]+?");
        try
        {
            if (Exception_Found.matcher(name1).find() ||
Exception_Found.matcher(name2).find()) {
                throw new StringChecking("Found EXCEPTION!\nPerforming necessary exception
handling to see that there is no presence of 'punctuations, special characters, or
numbers.\n By removing Exceptions.");
            }
        }
        catch (StringChecking error)
```

```
        {
            System.out.println(error);
        }
     }

     /** Part C Solution Ending here.....................................................
*/


     /** Part B Solution */

    /**Mehtod to convert string to binary
     * First converting each character to ASCII value
     * then converting ASCII values to Binary */
    static String strToBinary_Mukherjee (String s )
    {
        String S ="";
        for (int i =0; i <s .length(); i ++)
        {
            int val  = Integer.valueOf(s .charAt(i ));
            String bin  = "";
            while (val  > 0)
            {
                if (val  % 2 == 1)
                {
                    bin  += '1';
                }
                else
                    bin  += '0';
                val  /= 2;
            }
            bin  = reverse_Mukherjee (bin );
            S =S +bin ; // S contains the converted binary string
        }
        return S ;
    }
    static String reverse_Mukherjee (String input )
    {
        char[] a =input.toCharArray();
        int l , r =0;
        r  = a .length - 1;
        for (l  = 0; l <r ; l ++, r --)
        {
            char temp  = a [l ];
            a [l ] = a [r ];
            a [r ] = temp ;
        }
        return String.valueOf(a );
    }
    /**Method to insert padding if both the strings are not equal after
     * converting to their respective binary string*/
    static String addZerosPadding_Mukherjee (String str , int n )
    {
        for (int i =0; i <n ; i ++)
        {
            str ="0"+str ;
```

```
        }
        return str ;
    }
    /**Method to perform Bitwise XOR operation */
    static String getXOR_Mukherjee (String a , String b )
    {
        int aLength  = a .length();
        int bLength  = b .length();
        //checking if padding of Binary Strings is reqired
        if (aLength  > bLength )
        {
            a  = addZerosPadding_Mukherjee (b , aLength  - bLength );
        }
        else if (bLength  > aLength )
        {
            a  = addZerosPadding_Mukherjee (a , bLength  - aLength );
        }
        //performing Bitwise XOR here.
        int len  = Math.max(aLength , bLength );
        String res  = "";

        for (int i =0; i <len ; i ++)
        {
            res  += a .charAt(i ) ^ b .charAt(i );
        }
        return res ;
    }

    /** Part B Solution ending
here.......................................................... */

    //Main Preogram body
    public static void main(String args[])
    {

        /** Part A Solution */

        String word  = "FLAMES";

        /**Taking inputs of two names*/
        Scanner scan  = new Scanner(System.in);
        System.out.println("Enter first name : ");
        String name1  = scan .nextLine();
        System.out.println("Enter Second name : ");
        String name2  = scan .nextLine();

        Tuhin_p3 obj = new Tuhin_p3();
        obj.Checking_String_Mukherjee(name1, name2);

        /**Performing necessary exception handling to see that there is no presence of
        'punctuations, special characters, or numbers '*/
        name1 =name1 .replaceAll("[^a-zA-Z ]", "");
        name2 =name2 .replaceAll("[^a-zA-Z ]", "");

        /**Removal of common letters from both the words*/
        String common ="";
```

```
        for(int i =0; i <name1 .length();i ++)
        {
            for(int j =0; j <name2 .length();j ++)
            {
                if(name1 .charAt(i ) == name2 .charAt(j ))
                {
                    common  += name1 .charAt(i );
                }
            }
        }
        for(int i =0; i <common .length();i ++)
        {
            String ch  = common .charAt(i )+"";
            name1  = name1 .replace(ch , "");
            name2  = name2 .replace(ch , "");
        }

        /**Finding the correct position*/
        int name1_length =name1 .length();
        int name2_length =name2 .length();
        int word_length =word .length();
        int position =(name1_length +name2_length )%word_length ;
        char ch =word .charAt(position );

        /**searching word FLAMES for correct match*/
        if(ch =='F')
        {
            System.out.println("Friend");
        }
        else if(ch =='L')
        {
            System.out.println("Love");
        }
        else if(ch =='A')
        {
            System.out.println("Affection");
        }
        else if(ch =='M')
        {
            System.out.println("Marriage");
        }
        else if(ch =='E')
        {
            System.out.println("Enemy");
        }
        else if(ch =='S')
        {
            System.out.println("Serious");
        }

        /** Part A Solution Ending here..................................... */

        String S1 =strToBinary_Mukherjee (name1 );
        String S2 =strToBinary_Mukherjee (name2 );
        String X_O_R =getXOR_Mukherjee (S1 , S2 );
        int flag =0;
```

```
        for(int i =0; i <X_O_R .length()-1; i ++)
        {
            if(X_O_R .charAt(i ) == X_O_R .charAt(i +1))
            {
                flag  = 0;
            }
            else
            {
                flag  = 1;
            }
        }
        if(flag ==0)
        {
            System.out.println("Perfect match");
        }
        else
        {
            System.out.println("Not a perfect match");
        }
    }
}
```

This is the end of the complete code of the Whole problem statement but as per the given question I have taken screenshot of the subparts containing the individual solutions of the asked question and also provided the necessary outputs.

***All the screenshots taken from the above code.**

(a) Screenshot

```
112     //Main Preogram body
        Run | Debug
113     public static void main(String args[])
114     {
115
116         /** Part A Solution */
117
118         String word  = "FLAMES";
119
120         /**Taking inputs of two names*/
121         Scanner scan  = new Scanner(System.in);
122         System.out.println("Enter first name : ");
123         String name1  = scan .nextLine();
124         System.out.println("Enter Second name : ");
125         String name2  = scan .nextLine();
126
127         Tuhin_p3 obj = new Tuhin_p3();
128         obj.Checking_String_Mukherjee(name1, name2);
129
130         /**Performing necessary exception handling to see that there is no presence of
131         'punctuations, special characters, or numbers '*/
132         name1 =name1 .replaceAll("[^a-zA-Z ]", "");
133         name2 =name2 .replaceAll("[^a-zA-Z ]", "");
134
135         /**Removal of common letters from both the words*/
136         String common ="";
137         for(int i =0; i <name1 .length();i ++)
138         {
139             for(int j =0; j <name2 .length();j ++)
140             {
141                 if(name1 .charAt(i ) == name2 .charAt(j ))
142                 {
143                     common  += name1 .charAt(i );
144                 }
145             }
146         }
147         for(int i =0; i <common .length();i ++)
148         {
149             String ch  = common .charAt(i )+"";
150             name1  = name1 .replace(ch , "");
151             name2  = name2 .replace(ch , "");
152         }
```

```
153
154           /**Finding the correct position*/
155           int name1_length =name1 .length();
156           int name2_length =name2 .length();
157           int word_length =word .length();
158           int position =(name1_length +name2_length )%word_length ;
159           char ch =word .charAt(position );
160
```

```
161         /**searching word FLAMES for correct match*/
162         if(ch =='F')
163         {
164             System.out.println("Friend");
165         }
166         else if(ch =='L')
167         {
168             System.out.println("Love");
169         }
170         else if(ch =='A')
171         {
172             System.out.println("Affection");
173         }
174         else if(ch =='M')
175         {
176             System.out.println("Marriage");
177         }
178         else if(ch =='E')
179         {
180             System.out.println("Enemy");
181         }
182         else if(ch =='S')
183         {
184             System.out.println("Serious");
185         }
186 
187         /** Part A Solution Ending here..................................... */
```

Output:

```
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2> javac Tuhin_p3.java
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2> java Tuhin_p3
Enter first name :
Ranbir
Enter Second name :
Deepika
Marriage
```

(b) Screenshot

```
35      /** Part B Solution */
36
37      /**Mehtod to convert string to binary
38       * First converting each character to ASCII value
39       * then converting ASCII values to Binary */
40      static String strToBinary_Mukherjee (String s )
41      {
42          String S ="";
43          for (int i =0; i <s .length(); i ++)
44          {
45              int val  = Integer.valueOf(s .charAt(i ));
46              String bin  = "";
47              while (val  > 0)
48              {
49                  if (val  % 2 == 1)
50                  {
51                      bin  += '1';
52                  }
53                  else
54                      bin  += '0';
55                  val  /= 2;
56              }
57              bin  = reverse_Mukherjee (bin );
58              S =S +bin ; // S contains the converted binary string
59          }
60          return S ;
61      }
62      static String reverse_Mukherjee (String input )
63      {
64          char[] a =input.toCharArray();
65          int l , r =0;
66          r  = a .length - 1;
67          for (l  = 0; l <r ; l ++, r --)
68          {
69              char temp  = a [l ];
70              a [l ] = a [r ];
71              a [r ] = temp ;
72          }
73          return String.valueOf(a );
74      }
```

```
75      /**Method to insert padding if both the strings are not equal after
76       * converting to their respective binary string*/
77      static String addZerosPadding_Mukherjee (String str , int n )
78      {
79          for (int i =0; i <n ; i ++)
80          {
81              str ="0"+str ;
82          }
83          return str ;
84      }
```

```java
    /**Method to perform Bitwise XOR operation */
    static String getXOR_Mukherjee (String a , String b )
    {
        int aLength  = a .length();
        int bLength  = b .length();
        //checking if padding of Binary Strings is reqired
        if (aLength  > bLength )
        {
            a  = addZerosPadding_Mukherjee (b , aLength  - bLength );
        }
        else if (bLength  > aLength )
        {
            a  = addZerosPadding_Mukherjee (a , bLength  - aLength );
        }
        //performing Bitwise XOR here.
        int len  = Math.max(aLength , bLength );
        String res  = "";

        for (int i =0; i <len ; i ++)
        {
            res  += a .charAt(i ) ^ b .charAt(i );
        }
        return res ;
    }

    /** Part B Solution ending here..... */
```

```java
            String S1 =strToBinary_Mukherjee (name1 );
            String S2 =strToBinary_Mukherjee (name2 );
            String X_O_R =getXOR_Mukherjee (S1 , S2 );
            int flag =0;
            for(int i =0; i <X_O_R .length()-1; i ++)
            {
                if(X_O_R .charAt(i ) == X_O_R .charAt(i +1))
                {
                    flag  = 0;
                }
                else
                {
                    flag  = 1;
                }
            }
            if(flag ==0)
            {
                System.out.println("Perfect match");
            }
            else
            {
                System.out.println("Not a perfect match");
            }
        }
    }
```

Output:

```
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2> javac Tuhin_p3.java
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2> java Tuhin_p3
Enter first name :
Ranbir
Enter Second name :
Deepika
Marriage
Not a perfect match
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2>
```

(c) Screenshot

```
14    /**Part C Solution */
15
16    /**Dealing with Exceptions in the User Entered String*/
17    void Checking_String_Mukherjee(String name1, String name2)
18    {
19        Pattern Exception_Found = Pattern.compile("[^a-zA-Z ]+?");
20        try
21        {
22            if (Exception_Found.matcher(name1).find() || Exception_Found.matcher(name2).find()) {
23                throw new StringChecking("Found EXCEPTION!\nPerforming necessary exception handling to see that there is no presence of 'punctuations, special characters, or numbers.\n By removir
24            }
25        }
26        catch (StringChecking error)
27        {
28            System.out.println(error);
29        }
30    }
31
32    /** Part C Solution Ending here..... */
```

```
130             /**Performing necessary exception handling to see that there is no presence of
131             'punctuations, special characters, or numbers '*/
132             name1 =name1 .replaceAll("[^a-zA-Z ]", "");
133             name2 =name2 .replaceAll("[^a-zA-Z ]", "");
134
```

Output:

```
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2> javac Tuhin_p3.java
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2> java Tuhin_p3
Enter first name :
Ran@bir
Enter Second name :
Deepika
StringChecking: Found EXCEPTION!
Performing necessary exception handling to see that there is no presence of 'punctuations, special characters, or numbers.
 By removing Exceptions.
Marriage
Not a perfect match
PS D:\Work\college\3rd year\5th sem\oop lsb\pca2>
```