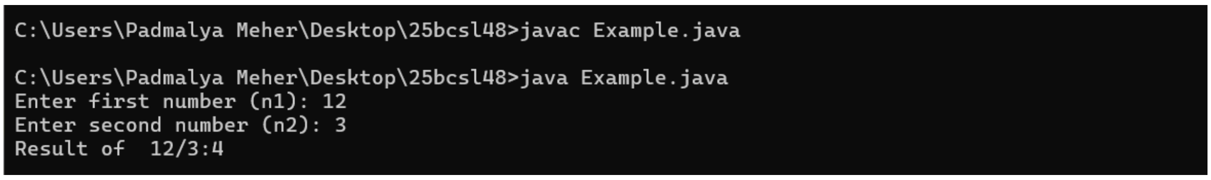
ASSIGNMENT - 7

1.Write an exception handling java program to read two numbers n1, n2 and calculate and print the result of n1/n2. If n2 is Zero (0) then it will be handled by exception handler and again ask the value of n2. In the exception handler the program should display appropriate message to the user.

```
import java.util.Scanner;
public class Example {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int n1, n2;
      System.out.print("Enter first number (n1): ");
      n1 = sc.nextInt();
         try {
            System.out.print("Enter second number (n2): ");
            n2 = sc.nextInt();
            int result = n1 / n2;
            System.out.println("Result of" + "  " + n1 + "/" + n2 + ":" + result);
          }
         catch (ArithmeticException e) {
            System.out.println("Error: Not divisible by 0.Please enter a non-zero value of n2");
         }
      sc.close();
   }
}
```

OUTPUT :



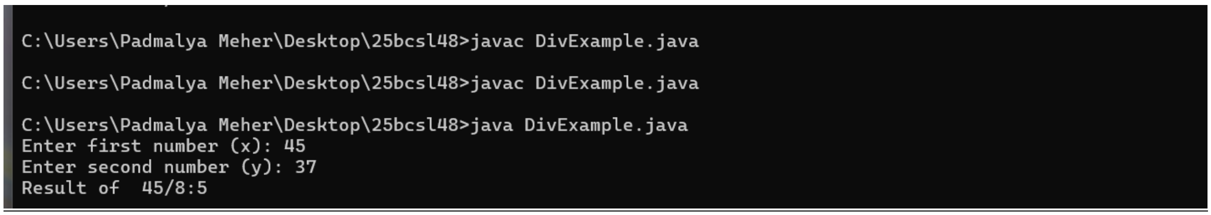
```
C:\Users\Padmalya Meher\Desktop\25bcsl48>javac Example.java

C:\Users\Padmalya Meher\Desktop\25bcsl48>java Example.java
Enter first number (n1): 12
Enter second number (n2): 3
Result of  12/3:4
```

2.Write a java program to read two numbers x and y and calculate x/(x–y). The program should check the value of x–y. Before dividing with x, it should throw an exception if x–y is zero. In the exception handler the program should display appropriate message to the user.

```
import java.util.Scanner;
public class DivExample {
   public static void main(String[] args) {
      int x , y;
      Scanner sc = new Scanner(System.in);
```

```
        System.out.print("Enter first number (x): ");
        x = sc.nextInt();
          try {
            System.out.print("Enter second number (y): ");
            y = sc.nextInt();
            int res = x-y;
            int result = x / res;
            System.out.println("Result of" + "  " + x + "/" + (x-y) + ":" + result);
          }
          catch (ArithmeticException e) {
            System.out.println("Error: Not divisible by 0.the value of (x-y) should not be zero");
          }
        sc.close();
      }
    }
```

OUTPUT :



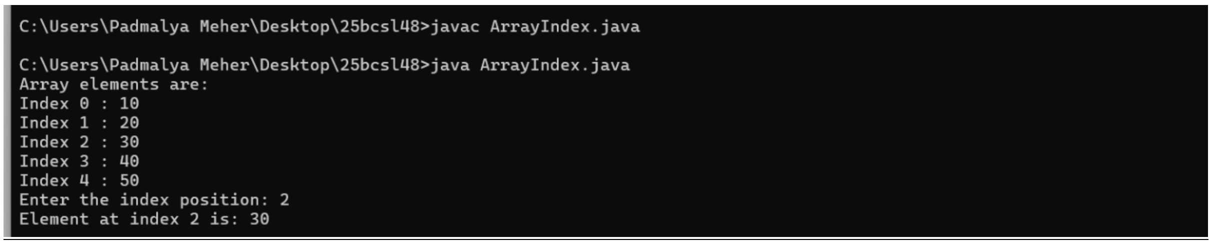
```
C:\Users\Padmalya Meher\Desktop\25bcsl48>javac DivExample.java

C:\Users\Padmalya Meher\Desktop\25bcsl48>javac DivExample.java

C:\Users\Padmalya Meher\Desktop\25bcsl48>java DivExample.java
Enter first number (x): 45
Enter second number (y): 37
Result of  45/8:5
```

3,Write an exception handling java program to print the index position of an existing integer array. The index value will be entered by user. It will be handled by exception handler if index position is greaterthen the size of array. In the exception handler the program should display appropriate message to the user.

```
import java.util.Scanner;
public class ArrayIndex {
  public static void main(String[] args)
    int[] numbers = {10, 20, 30, 40, 50};
    Scanner scanner = new Scanner(System.in);
    System.out.println("Array elements are:");
    for (int i = 0; i < numbers.length; i++) {
      System.out.println("Index " + i + " : " + numbers[i]);
    }
    System.out.print("Enter the index position: ");
```

```
        int index = scanner.nextInt();
        try {
            System.out.println("index position of the element is:" +  + " is: " + numbers[index]);
        }
        catch (ArrayIndexOutOfBoundsException e) {
        System.out.println("Error: Invalid index! Please enter an index between 0 and " + (numbers.length - 1));
        }
        scanner.close();
    }
}
```

OUTPUT :

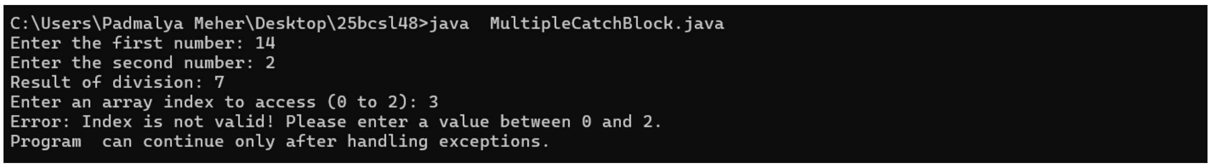


```
C:\Users\Padmalya Meher\Desktop\25bcsl48>javac ArrayIndex.java

C:\Users\Padmalya Meher\Desktop\25bcsl48>java ArrayIndex.java
Array elements are:
Index 0 : 10
Index 1 : 20
Index 2 : 30
Index 3 : 40
Index 4 : 50
Enter the index position: 2
Element at index 2 is: 30
```

4.Write a program to illustrate the use of multiple catch blocks associated with a single try block.

```
import java.util.Scanner;
public class MultipleCatchBlock {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Enter the first number: ");
            int num1 = sc.nextInt();
            System.out.print("Enter the second number: ");
            int num2 = sc.nextInt();
            int result = num1 / num2;
            System.out.println("Result of division: " + result);
            int[] arr = {10, 20, 30};
            System.out.print("Enter an array index to access (0 to 2): ");
            int index = sc.nextInt();
            System.out.println("Element at index " + index + ": " + arr[index]);
           }
            catch (ArithmeticException e) {
            System.out.println("Error: Not divisible by 0!");}
            catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Error: Index is not valid! Please enter a value between 0 and 2."); }
           catch (Exception e) {
                System.out.println("Error: An unexpected error occurred - " + e.getMessage());
     }
        System.out.println("Program  can continue only after handling exceptions.");
        sc.close();
    }
}
```

OUTPUT :



```
C:\Users\Padmalya Meher\Desktop\25bcsl48>java  MultipleCatchBlock.java
Enter the first number: 14
Enter the second number: 2
Result of division: 7
Enter an array index to access (0 to 2): 3
Error: Index is not valid! Please enter a value between 0 and 2.
Program  can continue only after handling exceptions.
```

5.Write a class called Account with the following properties and methods:

Properties: String name, int acc_no, double balance

Methods: void deposit (double amt)

void withdraw (double amt)

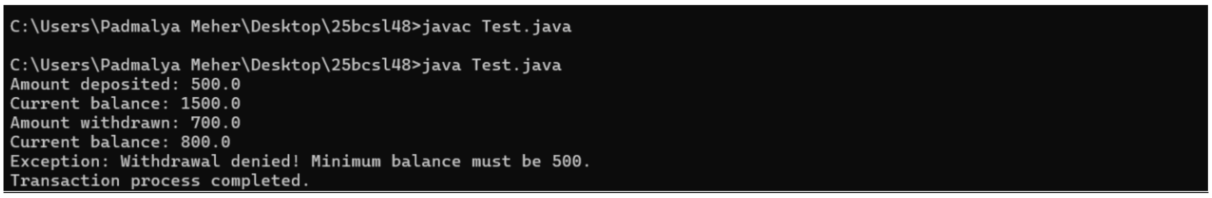
Assume that an account needs to have a minimum balance of 500. If an attempt is made to withdraw, which results in balance going below 500, throw a user defined exception called MinimumBalanceException. Use throw and throws wherever necessary.

```
class MinimumBalanceException extends Exception {
  public MinimumBalanceException(String message) {
    super(message);
  }
}
class Account {
  String name;
  int acc_no;
  double balance;
  public Account(String name, int acc_no, double balance) {
    this.name = name;
    this.acc_no = acc_no;
```

```
        this.balance = balance;
    }
    void deposit(double amt) {
        balance += amt;
        System.out.println("Amount deposited: " + amt);
        System.out.println("Current balance: " + balance);
    }
    void withdraw(double amt) throws MinimumBalanceException {
        if (balance - amt < 500) {
            throw new MinimumBalanceException( "Withdrawal denied! Minimum balance must be 500.”);
        } else {
            balance -= amt;
            System.out.println("Amount withdrawn: " + amt);
            System.out.println("Current balance: " + balance);
        } }
}
public class Test {
    public static void main(String[] args) {
        Account acc = new Account("John Doe", 1235, 1000.0);
        try {
            acc.deposit(500); // Normal deposit
            acc.withdraw(700); // Valid withdrawal
            acc.withdraw(900); // This should trigger MinimumBalanceException
        }
        catch (MinimumBalanceException e) {
            System.out.println("Exception: " + e.getMessage());
        }
        System.out.println("Transaction process completed.");
    }
}
```

OUTPUT :

ASSIGNMENT - 7


```
C:\Users\Padmalya Meher\Desktop\25bcsl48>javac Test.java

C:\Users\Padmalya Meher\Desktop\25bcsl48>java Test.java
Amount deposited: 500.0
Current balance: 1500.0
Amount withdrawn: 700.0
Current balance: 800.0
Exception: Withdrawal denied! Minimum balance must be 500.
Transaction process completed.
```

6.Write a class called Account with the following properties and methods:

Properties: String name, int acc_no, double balance

Methods: void deposit (double amt)

void withdraw (double amt),

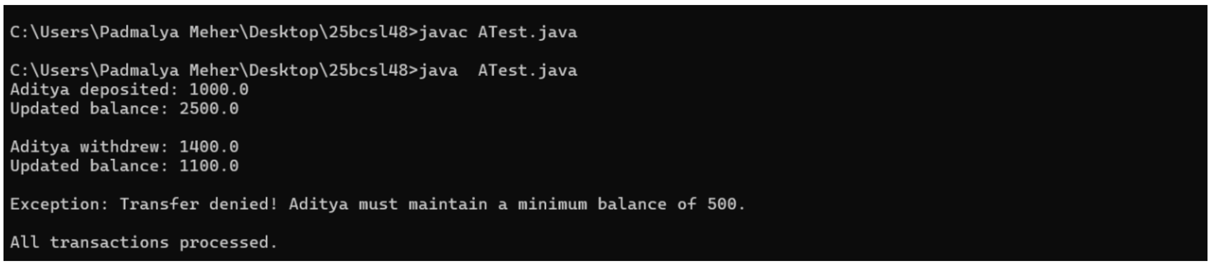
void transfer (Account acc1, Account acc2, double amt)

Assume that an account needs to have a minimum balance of 500. If an attempt is made to withdraw or transfer, which results in balance going below 500, throw a user defined exception called MinimumBalanceException. Use throw and throws wherever necessary

```
class MinimumBalanceException extends Exception {
  public MinimumBalanceException(String message) {
    super(message);
  }
}
class Account {
  String name;
  int acc_no;
  double balance;
  public Account(String name, int acc_no, double balance) {
    this.name = name;
    this.acc_no = acc_no;
    this.balance = balance;
  }
  void deposit(double amt) {
    balance += amt;
    System.out.println(name + " deposited: " + amt);
    System.out.println("Updated balance: " + balance + "\n");
  }
  void withdraw(double amt) throws MinimumBalanceException {
    if (balance - amt < 500) {
      throw new MinimumBalanceException("Withdrawal denied for " + name +
        "! Minimum balance must be 500." );
    }
else {
      balance -= amt;
      System.out.println(name + " withdrew: " + amt);
      System.out.println("Updated balance: " + balance + "\n");
    } }
  void transfer(Account acc1, Account acc2, double amt) throws MinimumBalanceException {
    if (acc1.balance - amt < 500) {
      throw new MinimumBalanceException("Transfer denied! " + acc1.name +
        " must maintain a minimum balance of 500." );
    } else {
      acc1.balance -= amt;
      acc2.balance += amt;
      System.out.println("Transfered " + amt + " from " + acc1.name + " to " + acc2.name);
      System.out.println(acc1.name + "'s updated balance: " + acc1.balance);
      System.out.println(acc2.name + "'s updated balance: " + acc2.balance + "\n");
    }
}
}
public class ATest {
  public static void main(String[] args) {
    Account acc1 = new Account("Aditya", 122, 1500.0);
    Account acc2 = new Account("Biren", 125, 900.0);
    try {
      acc1.deposit(1000);
      acc1.withdraw(1400);
      acc1.transfer(acc1, acc2, 1000);
      acc1.withdraw(1000);
      acc2.transfer(acc2, acc1, 500);
    }
    catch (MinimumBalanceException e) {
      System.out.println("Exception: " + e.getMessage() + "\n");
    }
```

```
        System.out.println("All transactions processed.");
    }
}
```

OUTPUT :



```
C:\Users\Padmalya Meher\Desktop\25bcsl48>javac ATest.java

C:\Users\Padmalya Meher\Desktop\25bcsl48>java  ATest.java
Aditya deposited: 1000.0
Updated balance: 2500.0

Aditya withdrew: 1400.0
Updated balance: 1100.0

Exception: Transfer denied! Aditya must maintain a minimum balance of 500.

All transactions processed.
```

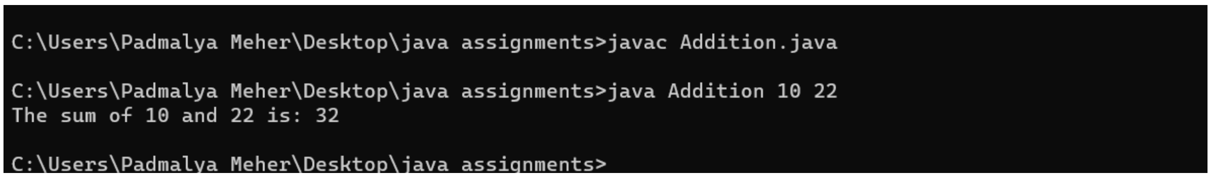
7.Write a program that prompts the user to enter a length in feet and inches and outputs the equivalent length in centimetres. If the user enters a negative number or a non-digit number, throw and handle an appropriate exception and prompt the user to enter another set of numbers.

```
import java.util.InputMismatchException;
import java.util.Scanner;
class NegativeNumberException extends Exception {
    public NegativeNumberException(String message) {
        super(message);
    }}
public class Driver {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        boolean validInput = false; // controls the loop
        while (!validInput) {
            try {
                System.out.print("Enter length in feet: ");
                double feet = sc.nextDouble();
                System.out.print("Enter length in inches: ");
                double inches = sc.nextDouble();
                if (feet < 0 || inches < 0) {
                    throw new NegativeNumberException("Error: Feet and inches cannot be negative!");
                }
                double totalInches = (feet * 12) + inches;
                double centimetres = totalInches * 2.54;
```

ASSIGNMENT - 7

```
            System.out.printf("Equivalent length: %.2f centimetres%n", centimetres);
            validInput = true;   }
          catch (InputMismatchException e) {
            System.out.println("Error: Please enter numeric values only!");
            sc.nextLine();
          }
          catch (NegativeNumberException e) {
            System.out.println(e.getMessage());
          }
if (!validInput) {
System.out.println("Please try again.\n");
          } }
        sc.close();
    }
}
```

OUTPUT :



```
C:\Users\Padmalya Meher\Desktop\java assignments>javac Addition.java

C:\Users\Padmalya Meher\Desktop\java assignments>java Addition 10 22
The sum of 10 and 22 is: 32

C:\Users\Padmalya Meher\Desktop\java assignments>
```

Name – Padmalaya Meher

Roll.no – 01

Date of Experiment – 10/10/25