WEEK 12

Question 1: Write a program that calls a method that throws an exception of type ArithmeticException in a for loop at an undesirable situation (such as divide by zero or taking square root of negative number). Catch the exception and display appropriate message. (Example of Unchecked Exception).

Code: *Main.java*

```
package shunya;
import java.util.Scanner;

public class Main {
   //Method for division
   public static int divideNumbers(int dividend, int divisor) {
      return dividend / divisor;
   }
   //Main method
   public static void main(String[] args) {

      Scanner scanner = new Scanner(System.in);
      System.out.print("How many division operations would you like to perform? ");
      int n = scanner.nextInt();

      for (int i = 1; i <= n; i++) {

         System.out.print("Enter dividend (numerator): ");
         int dividend = scanner.nextInt();
         System.out.print("Enter divisor (denominator): ");
         int divisor = scanner.nextInt();

         try {
            int result = divideNumbers(dividend, divisor);
            System.out.println("Result: " + dividend + " / " + divisor + " = " + result);

         } catch (ArithmeticException e) {

            System.out.println("Division by zero is not allowed!");
            System.out.println("Please enter a non-zero divisor.");
         }
      }
      scanner.close();
   }
}
```

Output:

```
"C:\Program Files\Java\OpenJDK\jdk-25\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=59580" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni Material\LAB\sem 3\Week 12\Question1\target\classes" shunya.Main
How many division operations would you like to perform? 5
Enter dividend (numerator): 865309
Enter divisor (denominator): 3
Result: 865309 / 3 = 288436
Enter dividend (numerator): 128
Enter divisor (denominator): 0
Division by zero is not allowed!
Please enter a non-zero divisor.
Enter dividend (numerator): 67
Enter divisor (denominator): 84
Result: 67 / 84 = 0
Enter dividend (numerator): 0
Enter divisor (denominator): 95
Result: 0 / 95 = 0
Enter dividend (numerator): 0
Enter divisor (denominator): 0
Division by zero is not allowed!
Please enter a non-zero divisor.

Process finished with exit code 0
```

Question 2: Write a program of your choice where a Checked Exception occurs at third function but handled at the first calling function. Use both ways of managing Checked Exception i.e. using try-catch block and throws keyword.

Code: *Main.java*

```
package com.question2;
import java.io.IOException;
public class Main {
    static void thirdFunction() throws IOException {
        System.out.println("Third function: Throwing IOException...");
        throw new IOException("IOException in third function!");
    }
    static void secondFunction() throws IOException {
        System.out.println("Second function: Calling third function...");
        thirdFunction();
    }
    static void firstFunction() throws IOException {
        System.out.println("First function: Calling second function...");
        secondFunction();
    }
    public static void main(String[] args) {
        try {
            firstFunction();
        } catch (IOException e) {
            System.out.println("Main caught the exception: " + e.getMessage());
        }
        System.out.println("Program ends.");
    }
}
```

Output:

```
"C:\Program Files\Java\OpenJDK\jdk-25\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=57888" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni Material\LAB\sem 3\Week 12\Question2\target\classes" com.question2.Main

First function: Calling second function...

Second function: Calling third function...

Third function: Throwing IOException...

Main caught the exception: IOException in third function!

Program ends.


Process finished with exit code 0
```

Question 3: You are developing an online banking system where users can transfer money between accounts. If a user tries to withdraw more money than is available in their account, an InsufficientFundsException should be thrown.

Code: *InsufficientFundsException.java*

```
package com.bank;
public class InsufficientFundsException extends Exception {
   public InsufficientFundsException() { super("Insufficient Funds!"); }
}
```

Account.java

```
package com.bank;
public class Account {
   double balance;
   public Account(double balance){ this.balance = balance; }
   public void withdraw(int amount) throws InsufficientFundsException {
      if(amount > balance) throw new InsufficientFundsException();
      else balance -= amount;
   }
}
```

Main.java

```
package com.bank;
import java.util.Scanner;
public class Main {
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);
      System.out.print("Enter opening balance: ");
      int openingBalance = input.nextInt();
      System.out.print("Please enter the amount you want to withdraw: ");
      int amount = input.nextInt();
      Account acc = new Account(openingBalance);
      try {
         acc.withdraw(amount);
      } catch (InsufficientFundsException e) {
         System.out.println("Exception Occurred: " + e.getMessage());
      }
   }
}
```

Output:

```
"C:\Program Files\Java\OpenJDK\jdk-25\bin\java.exe" "-javaagent:C:\Program
Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=56602" -Dfile.encoding=UTF-8 -
Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni
Material\LAB\sem 3\Week 12\Question3\target\classes" com.bank.Main

Enter opening balance: 1100

Please enter the amount you want to withdraw: 2000

Exception Occurred: Insufficient Funds!

Process finished with exit code 0
```

Question 4: Create a user-defined exception InvalidAgeException when the age of a person is below 18 years. Use this exception at appropriate place.

Code: *InvalidAgeException.java*

```java
package com.election.committee;
public class InvalidAgeException extends Exception {
    public InvalidAgeException(String msg) {
        super(msg);
    }
}
```

Person.java

```java
package com.election.committee;
public class Person {

    private String name;
    private int age;

    public Person(String name, int age) {
        this.name = name; this.age = age;
    }
    public void checkAge() throws InvalidAgeException {
        if (age < 18) {
            throw new InvalidAgeException("Age must be 18 or above. Current age: "
+ age);
        }
    }
    public String getName() { return name; }
}
```

Code:

Main.java

```
package com.election.committee;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.print("Please enter your name: ");
        String name = sc.nextLine();
        System.out.print("Please enter your age: ");
        int age = sc.nextInt();

        Person p1 = new Person(name, age);

        try {
            p1.checkAge();
        } catch (InvalidAgeException e) {
            System.out.println("Exception caught for " +  p1.getName() + ".");
            System.out.println(e.getMessage());
        }
    }
}
```

Output:

```
"C:\Program Files\Java\OpenJDK\jdk-25\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=56340" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath "D:\Uni Material\LAB\sem 3\Week 12\Question4\target\classes" com.election.committee.Main

Please enter your name: Chandu

Please enter your age: 11

Exception caught for Chandu.

Age must be 18 or above. Current age: 11


Process finished with exit code 0
```