Assignment 10

**Question 1:** Write a JAVA program to demonstrate the usage of throw keyword

**Code:**

class test

{

public static void main(String[] args)

{

try{

throw new ArrayIndexOutOfBoundsException();

}

catch(Exception e)

{

System.out.println("Exception caught");

}

}

}

**Output:**

****

**Question 2:** Write a JAVA program to demonstrate throws

**Code:**

import java.io.\*;

class test

{

public static void main(String[] args)throws Exception

{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int n;

try

{

n = Integer.parseInt(br.readLine());

}

catch(Exception e)

{

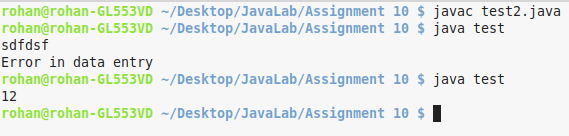
System.out.println("Error in data entry");

}

}

}

**Output:**



**Question 3:** Write a JAVA program to show the use of finally keyword

**Code:**

import java.io.\*;

class test

{

public static void main(String[] args)throws Exception

{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int n;

try

{

n = Integer.parseInt(br.readLine());

}

catch(Exception e)

{

System.out.println("Error in data entry");

}

finally

{

System.out.println("Data entry finished!!!");

}

}

}

**Output:**



**Question 4:** Write a program to accept a number and print the factorial of that number. If a negative number is entered, NegativeNumberException is thrown

**Code:**

import java.util.Scanner;

class NegativeNumber extends Exception

{

public NegativeNumber()

{

super("You have entered a negative number!!!");

}

}

class test

{

public static void main(String[] args)throws Exception

{

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

try

{

if(n < 0)

throw new NegativeNumber();

int ans = 1;

for(int i = 2; i <= n; i++)

ans \*= i;

System.out.println(ans);

}

catch(NegativeNumber ex)

{

System.out.println("NegativeNumber exception caught");

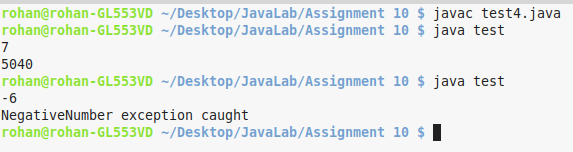
ex.getMessage();

}

}

}

**Output:**



**Question 5:** Write a JAVA program to implement a class Account which has some account balance in it and has functions for withdrawing and depositing funds. If the account balance falls below 1000, throw an exception

**Code:**

class InsufficientBalanceException extends Exception

{

InsufficientBalanceException()

{

super("Cannot have balance less than 1000");

}

}

class Account

{

double balance;

Account()

{

balance = 1000;

}

void deposit(double amt)

{

balance += amt;

System.out.println("Your current balance after deposit is Rs "+balance);

}

void withdraw(double amt)throws Exception

{

if(balance - amt < 1000)

throw new InsufficientBalanceException();

balance -= amt;

System.out.println("Your current balance after withdrawal is Rs "+balance);

}

}

class test

{

public static void main(String[] args) throws Exception

{

Account obj = new Account();

try

{

obj.deposit(500);

obj.withdraw(400);

obj.withdraw(400);

}

catch(InsufficientBalanceException exp)

{

System.out.println("Insufficient balance exception caught!!!");

System.out.println(exp.getMessage());

}

}

}

**Output:**

