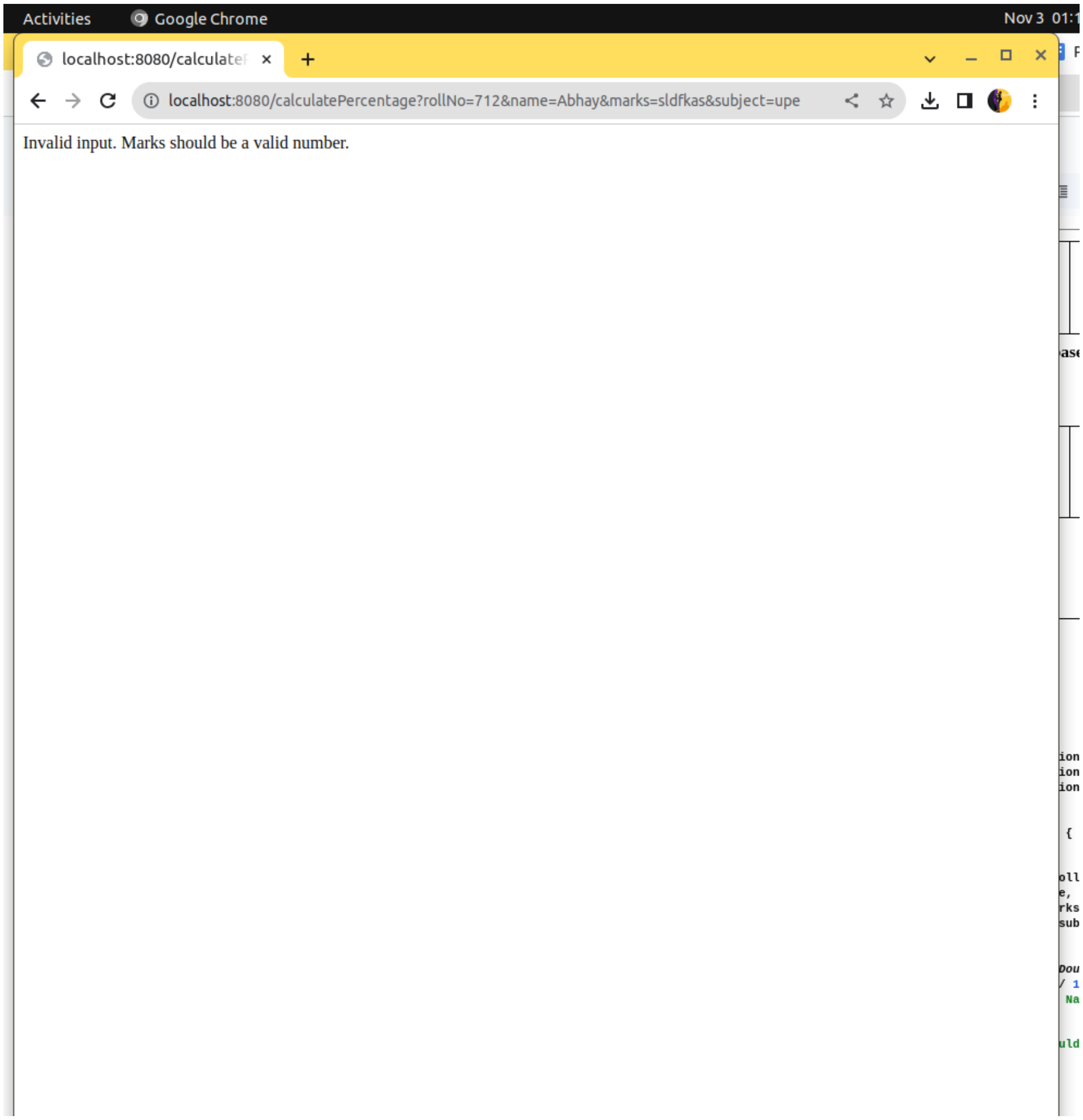
Name of student: Abhay Omprakash Prajapati		
Roll no: 41	**Tutorial No: 5**	
Title of LAB Assignment: Exception Handling with Spring boot		
DOP: 25-09-2023	**DOS:02-10-2023**	
CO Mapped: Co1,Co2	**PO Mapped:** PO3 ,PO6	**Signature:**

Example 1: NumberFormatException Handling

```
package com.example.a4_2.controllers;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class PercentageCalculatorController {
   @GetMapping("/calculatePercentage")
   public String calculatePercentage(
           @RequestParam("rollNo") String rollNo,
           @RequestParam("name") String name,
           @RequestParam("marks") String marks,
           @RequestParam("subject") String subject
   ) {
       try {
           double marksValue = Double.parseDouble(marks);
           double percentage = (marksValue / 100) * 100;
           return "Roll No: " + rollNo + ", Name: " + name + ", Percentage: "
+ percentage + "%";
       } catch (NumberFormatException e) {
           return "Invalid input. Marks should be a valid number.";
       }
   }
}
```

Output:


Activities
Google Chrome
Nov 3 01:1
localhost:8080/calculateF
localhost:8080/calculatePercentage?rollNo=712&name=Abhay&marks=sldfkas&subject=upe

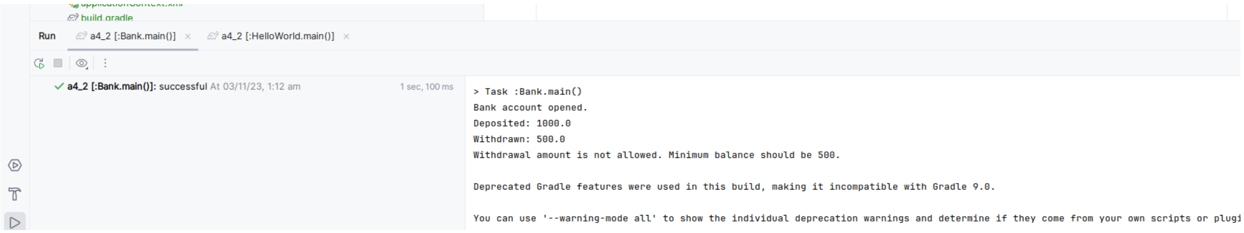
Invalid input. Marks should be a valid number.

Example 2: Custom Exception Handling

```
//PayoutOutOfBoundsException.java
package com.example.a4_2.exceptions;

public class PayoutOutOfBoundsException extends Exception {
   public PayoutOutOfBoundsException(String message) {
       super(message);
   }
}

//Bank.java
package com.example.a4_2.service;

import com.example.a4_2.exceptions.PayoutOutOfBoundsException;

public class Bank {
   private double balance = 0;

   public void open() {
       System.out.println("Bank account opened.");
   }

   public void deposit(double amount) {
       balance += amount;
       System.out.println("Deposited: " + amount);
   }

   public void withdraw(double amount) throws PayoutOutOfBoundsException {
       if (balance - amount < 500) {
           throw new PayoutOutOfBoundsException("Withdrawal amount is not
allowed. Minimum balance should be 500.");
       }
       balance -= amount;
       System.out.println("Withdrawn: " + amount);
   }
}


//Main.java
public static void main(String[] args) {
   Bank bank = new Bank();
   bank.open();
   bank.deposit(1000);
   try {
       bank.withdraw(500);
       bank.withdraw(500);
       bank.withdraw(500);
   } catch (PayoutOutOfBoundsException e) {
```

```
            System.out.println(e.getMessage());
        }
    }
```

Output:



✓ a4_2 [:Bank.main()]: successful At 03/11/23, 1:12 am 1 sec, 100 ms

```
> Task :Bank.main()
Bank account opened.
Deposited: 1000.0
Withdrawn: 500.0
Withdrawal amount is not allowed. Minimum balance should be 500.

Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.

You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugi
```