

Exception handling

Assignment-1

```
package ExampleProgram;

import java.util.Scanner;

public class SimpleCalculator {

    public static void main(String[] args) {

        // Create a scanner object to read user input
        Scanner scanner = new Scanner(System.in);

        try {
            // Prompt user to enter the first number
            System.out.print("Enter the first number: ");
            double num1 = scanner.nextDouble();

            // Prompt user to enter the second number
            System.out.print("Enter the second number: ");
            double num2 = scanner.nextDouble();

            // Prompt user to enter the operator
            System.out.print("Enter the operator (+, -, *, /): ");
            String operator = scanner.next();

            double result = 0; // Initialize result variable

            // Perform calculation based on the operator
            switch (operator) {
                case "+":
                    result = num1 + num2;
                    break;
                case "-":
                    result = num1 - num2;
                    break;
                case "*":
                    result = num1 * num2;
                    break;
                case "/":
                    // Handle division by zero
```

```

if (num2 == 0) {
System.out.println("Error: Division by zero is not allowed.");
return; // Exit the program after showing the error
} else {
result = num1 / num2;
}
break;
default:
System.out.println("Error: Invalid operator.");
return; // Exit the program after showing the error
}

// Display the result of the calculation
System.out.println("The result of " + num1 + " " + operator + " " + num2 + " is: " +
result);

} catch (Exception e) {
// Handle any input mismatch or other exceptions
System.out.println("Error: Invalid input. Please enter valid numbers.");
} finally {
// Close the scanner
scanner.close();

}

}
}

```

Output:

```

Enter the first number: 23
Enter the second number: 21
Enter the operator (+, -, *, /): *
The result of 23.0 * 21.0 is: 483.0

```

```

Enter the first number: 47
Enter the second number: 65
Enter the operator (+, -, *, /): +
The result of 47.0 + 65.0 is: 112.0

```

```

Enter the first number: 87
Enter the second number: 32

```

Enter the operator (+, -, *, /): -
The result of 87.0 - 32.0 is: 55.0

Enter the first number: 98
Enter the second number: 2
Enter the operator (+, -, *, /): /
The result of 98.0 / 2.0 is: 49.0

Assignment-2

```
package ExampleProgram;

import java.util.Scanner;

//BankAccount class that represents a simple bank account
class BankAccount
{
    private double balance; // The balance of the account

    // Constructor to initialize the balance
    public BankAccount(double initialBalance)
    {

        if (initialBalance >= 0)
        {

            this.balance = initialBalance;
        }
        else
        {
            this.balance = 0; // If the initial balance is negative, set it to 0
        }
    }

    // Method to deposit money into the account
    public void deposit(double amount)
    {

        if (amount > 0)
        {

            balance += amount;
            System.out.println("Deposited: Rs" + amount);
        }
    }
}
```

```

else
{
System.out.println("Error: Deposit amount must be greater than 0.");
}
}

// Method to withdraw money from the account
public void withdraw(double amount) throws InsufficientFundsException
{

if (amount <= 0)
{
System.out.println("Error: Withdrawal amount must be greater than 0.");
}
else if (amount > balance)
{
// If the withdrawal amount exceeds the balance, throw an exception
throw new InsufficientFundsException("Insufficient funds for this withdrawal.");
}
else
{
balance -= amount;
System.out.println("Withdrew: Rs" + amount);
}
}

// Method to get the current balance
public double getBalance()
{
return balance;
}
}

//Custom exception class for insufficient funds
class InsufficientFundsException extends Exception
{
public InsufficientFundsException(String message)
{
super(message);
}
}

public class SimpleBankingApp
{

public static void main(String[] args)

```

```

{
Scanner scanner = new Scanner(System.in);

// Create a bank account with an initial balance of Rs 10000
BankAccount account = new BankAccount(10000.00);

while (true)
{
System.out.println("\n--- Simple Banking Application ---");
System.out.println("Current balance: Rs: " + account.getBalance());
System.out.println("Choose an operation:");
System.out.println("1. Deposit");
System.out.println("2. Withdraw");
System.out.println("3. Exit");

int choice = scanner.nextInt();

switch (choice)
{
case 1:
// Deposit operation
System.out.print("Enter deposit amount Rs: ");
double depositAmount = scanner.nextDouble();
account.deposit(depositAmount);
break;

case 2:

// Withdraw operation
System.out.print("Enter withdrawal amount Rs: ");
double withdrawAmount = scanner.nextDouble();
try {
account.withdraw(withdrawAmount);
}
catch (InsufficientFundsException e)
{
System.out.println("Error: " + e.getMessage());
}
break;

case 3:
// Exit the program
System.out.println("Thank you for using the Simple Banking Application. Goodbye!");
scanner.close();
return;
}
}

```

```
default:
System.out.println("Invalid choice. Please try again.");
}
}

}

}
```

Output:

```
--- Simple Banking Application ---
Current balance: Rs: 10000.0
Choose an operation:
1. Deposit
2. Withdraw
3. Exit
1
Enter deposit amount Rs: 6000
Deposited: Rs6000.0

--- Simple Banking Application ---
Current balance: Rs: 16000.0
Choose an operation:
1. Deposit
2. Withdraw
3. Exit
2
Enter withdrawal amount Rs: 4000
Withdrew: Rs4000.0

--- Simple Banking Application ---
Current balance: Rs: 12000.0
Choose an operation:
1. Deposit
2. Withdraw
3. Exit
3
Thank you for using the Simple Banking Application. Goodbye!

--- Simple Banking Application ---
Current balance: Rs: 10000.0
Choose an operation:
1. Deposit
```

2. Withdraw

3. Exit

2

Enter withdrawal amount Rs: 30000

Error: Insufficient funds for this withdrawal.