Lab 31

Taking User Inputs in Java

Task: Demonstrate exception handling using sample codes:

In Java, you can take user inputs from the console using the `Scanner` class, which is a part of the `java.util` package. Here's a step-by-step guide on how to take user inputs in Java:

1. Import the `Scanner` class:

First, you need to import the `Scanner` class at the beginning of your Java file:

```java

import java.util.Scanner;

```

2. Create a `Scanner` object:

Next, create a `Scanner` object to read input from the console. You can do this in your `main` method or any other method where you want to take user input:

```java

Scanner scanner = new Scanner(System.in);

```

3. Prompt the user for input:

Display a message to prompt the user for input. This message should provide information about what kind of input you are expecting:

```java

System.out.print("Enter your name: ");

```

4. Read user input:

Use the `Scanner` object's methods to read user input. The most commonly used methods are `nextLine()` for reading strings, `nextInt()` for reading integers, and `nextDouble()` for reading floating-point numbers. Here's an example of reading a string:

```java

String name = scanner.nextLine();

```

You can similarly use `nextInt()` and `nextDouble()` for reading numbers.

5. Close the `Scanner` object (optional):

It's a good practice to close the `Scanner` object when you're done with it to free up system resources. You can do this using the `close()` method:

```java

scanner.close();

```

6. Use the user input:

You can now use the user's input in your program as needed. Here's a complete example that takes the user's name and greets them:

import java.util.Scanner;

public class UserInputDemo {

public static void main(String[] args) {

// Create a Scanner object

Scanner scanner = new Scanner(System.in);

// Prompt the user for input

System.out.print("Enter your name: ");

// Read user input (a string)

String name = scanner.nextLine();

// Close the Scanner object

scanner.close();

// Use the user's input

System.out.println("Hello, " + name + "!");

}

}

When you run this program, it will wait for the user to enter their name, and then it will greet them by printing "Hello, [name]!" where `[name]` is the name the user entered.

Remember to handle exceptions that may occur when reading user input, such as `InputMismatchException` if the input is not of the expected type. Also, be cautious when closing the `Scanner` object; if you close it prematurely, you won't be able to read further input.