TASK 10

1.Source code

Custom Exception (Checked Exception)

InsufficientBalanceException.java

```
/*
 * Custom checked exception
 */
public class InsufficientBalanceException extends Exception {

   public InsufficientBalanceException(String message) {
       super(message);
   }
}
```

This is a **checked exception** because it extends Exception.

Main Program – Exception Handling Demo

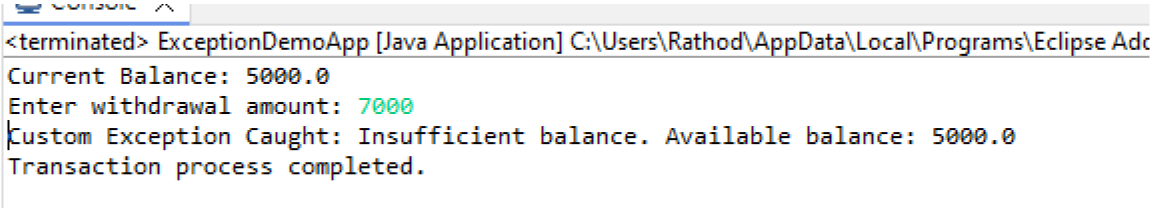
ExceptionDemoApp.java

```
import java.util.Scanner;

/*
 * Task 10: Exception Handling & Custom Exceptions
 */
public class ExceptionDemoApp {

   // Method that throws a custom checked exception
```

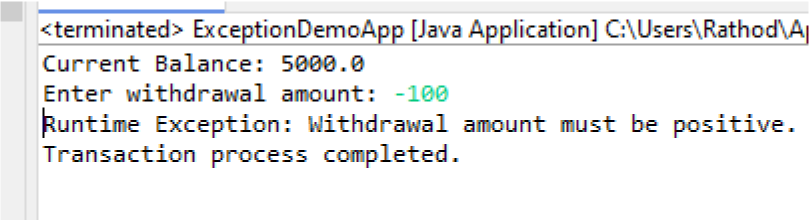
```
static void withdraw(double balance, double amount)
        throws InsufficientBalanceException {

    if (amount <= 0) {
        throw new IllegalArgumentException("Withdrawal amount must be positive.");
    }

    if (amount > balance) {
        throw new InsufficientBalanceException(
                "Insufficient balance. Available balance: " + balance);
    }

    System.out.println("Withdrawal successful. Remaining balance: " + (balance - amount));
}

public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);
    double balance = 5000;

    try {
        System.out.println("Current Balance: " + balance);

        System.out.print("Enter withdrawal amount: ");
        double amount = sc.nextDouble();
```

```
            // Calling method that may throw exception
            withdraw(balance, amount);

        }
        catch (InsufficientBalanceException e) {
            // Handling custom checked exception
            System.out.println("Custom Exception Caught: " + e.getMessage());
        }
        catch (IllegalArgumentException e) {
            // Handling unchecked exception
            System.out.println("Runtime Exception: " + e.getMessage());
        }
        catch (Exception e) {
            // Generic exception handling
            System.out.println("Unexpected Error: " + e.getMessage());
        }
        finally {
            // Cleanup block
            System.out.println("Transaction process completed.");
            sc.close();
        }
    }
}
```

2.Output



```
<terminated> ExceptionDemoApp [Java Application] C:\Users\Rathod\AppData\Local\Programs\Eclipse Ad
Current Balance: 5000.0
Enter withdrawal amount: 7000
Custom Exception Caught: Insufficient balance. Available balance: 5000.0
Transaction process completed.
```



```
<terminated> ExceptionDemoApp [Java Application] C:\Users\Rathod\A
Current Balance: 5000.0
Enter withdrawal amount: -100
Runtime Exception: Withdrawal amount must be positive.
Transaction process completed.
```