

Practical 7: Introduction to Exception Handling

Name: Sutariya Savankumar

Roll no: MA065

1. Create a try block that is likely to generate divide by zero and `StringIndexOutOfBoundsException` exceptions then incorporate necessary catch blocks to catch and handle them appropriately and also implement finally block.

Code

```
import java.util.Scanner;
public class ExceptionHandlingExample {
    public static void main(String[] args) {
        int x,y;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter two numbers: ");
        x = sc.nextInt();
        y = sc.nextInt();
        try {
            // Divide by zero exception
            int z = x / y;

            // String index out of bounds exception
            String str = "Hello";
            char c = str.charAt(10);
        } catch (ArithmeticException e) {
            System.out.println("Error: " + e.getMessage());
        } catch (StringIndexOutOfBoundsException e) {
            System.out.println("Error: " + e.getMessage());
        } finally {
            System.out.println("Finally block executed.");
        }
    }
}
```

Output

```

● PS C:\Drive\Study\MCA\DDU\SEM_2\OOPJ\Practicals\Labs Program\Lab 7> java ExceptionHandlingExample
Enter two numbers:
5
0
Error: / by zero
Finally block executed.
● PS C:\Drive\Study\MCA\DDU\SEM_2\OOPJ\Practicals\Labs Program\Lab 7> java ExceptionHandlingExample
⊗ Enter two numbers:
⊗ 5
● 7
Error: String index out of range: 10
Finally block executed.

```

2. Write a Java program to create a class BankDemo which invokes the deposit() and withdraw() methods of the CheckingAccount class where withdraw() throws an InsufficientFundsException exception if the amount to be withdrawn is more than the available balance. (Use throws keyword)

Code

```

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

class CheckingAccount {
    private double balance;
    private int accountNumber;

    public CheckingAccount(int accountNumber) {
        this.accountNumber = accountNumber;
    }

    public void deposit(double amount) {
        balance += amount;
    }

    public void withdraw(double amount) throws InsufficientFundsException {
        if (amount > balance) {
            throw new InsufficientFundsException("Insufficient funds");
        }
    }
}

```

```

        }
        balance -= amount;
    }

    public double getBalance() {
        return balance;
    }

    public int getAccountNumber() {
        return accountNumber;
    }
}

class BankDemo {
    public static void main(String[] args) {
        CheckingAccount account = new CheckingAccount(12345);

        account.deposit(1000);
        System.out.println("Account balance: $" + account.getBalance());

        try {
            account.withdraw(1500);
            System.out.println("Withdrawal successful. New balance: $" +
account.getBalance());
        } catch (InsufficientFundsException e) {
            System.out.println("Withdrawal failed: " + e.getMessage());
        }
    }
}

```

Output

- PS C:\Drive\Study\MCA\DDU\SEM_2\OOPJ\Practicals\Labs Program\Lab 7> javac .\BankDemo.java
- PS C:\Drive\Study\MCA\DDU\SEM_2\OOPJ\Practicals\Labs Program\Lab 7> java BankDemo
- Account balance: \$1000.0
- Withdrawal failed: Insufficient funds

3. Define an exception called “NoMatchException” that is thrown when a string is not equal to “Object Oriented Programing with JAVA”. Write a program that uses this exception. (Use throw keyword)

Code

```
import java.util.Scanner;
class NoMatchException extends Exception {
    public NoMatchException(String message) {
        super(message);
    }
}

class p3 {
    public static void main(String[] args) {
        String str;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a string: ");
        str = sc.nextLine();
        try {
            if (!str.equals("Object Oriented Programming with JAVA")) {
                throw new NoMatchException("String does not match");
            }
        } catch (NoMatchException e) {
            System.out.println("Caught exception: " + e.getMessage());
        }
    }
}
```

Output

- PS C:\Drive\Study\MCA\DDU\SEM_2\OOPJ\Practicals\Labs Program\Lab 7> javac .\p3.java
- PS C:\Drive\Study\MCA\DDU\SEM_2\OOPJ\Practicals\Labs Program\Lab 7> java p3
- Enter a string:
- Object Oriented Programming with JAVA
- PS C:\Drive\Study\MCA\DDU\SEM_2\OOPJ\Practicals\Labs Program\Lab 7> java p3
- Enter a string:
- ⊗ Hello this is Java
- Caught exception: String does not match