# Exception Handling – Question Paper

## Section A: Basic Try-Except (2 marks each)

1. Write a program to divide two numbers entered by the user. Handle ZeroDivisionError using try-except.
2. Write a program to convert a string to an integer. Handle ValueError if the input is not a valid number.
3. Accept two numbers from the user and perform addition. Use try-except to handle invalid input types.
4. Write a program to read an element from a list using an index entered by the user. Handle IndexError.

## Section B: Try-Except-Else (4 marks each)

1. Create a program that accepts a number from the user and prints its square. Use try-except-else to handle ValueError and ensure successful computation is shown only if there's no error.
2. Write a program to open a file and read contents. Use try-except-else to handle FileNotFoundError.
3. Write a Python program to convert a number to its binary format. Use try-except-else to handle any invalid input.

## Section C: Try-Finally (5 marks each)

1. Write a program that opens a file and ensures it gets closed, whether or not an exception occurs. Use try-finally.
2. Simulate a login process where the user input is handled in a try block and a log message is printed in finally regardless of success or failure.
3. Write a program that divides two numbers, catching errors with try-except, and printing a clean-up message using finally.

## Section D: Combined Exception Handling (6 marks each)

1. Create a program that handles multiple exceptions: ZeroDivisionError, ValueError, and always prints "Execution complete" using finally.
2. Write a program to simulate bank withdrawal. Use try-except-else-finally to handle incorrect amount input, and always print a message whether the transaction succeeded or failed.

**Answers:**

**Section A: Basic Try-Except**

**1. Divide two numbers (handle ZeroDivisionError):**

1. a = int(input("Enter num1"))
2. b = int(input("Enter num2"))
3. try:
4. res = a/b
5. print(res)
6. except ZeroDivisionError:
7. print("Error, cannot be divisible by zero")

2. Convert string to integer (handle ValueError):

str= input("Enter the num: ")

try:

    num = int(str)

    print(num)

except ValueError:

    print("Invalid input, not a number")

3. Addition of two numbers (handle invalid types):

a = int(input("Enter num1"))

b = int(input("Enter num2"))

try:

    res = a+b

    print(res)

except ValueError:

    print("Invalid input, enter valid numbers")

4. Access list element by index (handle IndexError):

list = ["apple","banana","mango"]

try:

    index = int(input("Enter the index : "))

    print(list[index])

except IndexError:

    print("Invalid index (out of range), enter valid index")

Section B: Try-Except-Else

5. Square of number (handle ValueError):

try:

    num = int(input("enter the number"))

except ValueError:

    print("Error: not a valid number")

else:

    print("Square of num:",num\*num)

6. Open and read file (handle FileNotFoundError):

try:

    f = open("text.txt","r")

except FileNotFoundError:

    print("File not found")

else:

    print(f.read())

    f.close()

7. Convert number to binary (handle invalid input):

try:

    num = int(input("Enter the number"))

except ValueError:

    print("invalid input")

else:

    print("Binary value:", bin(num))

Section C: Try-Finally

8. Ensure file is closed

try:

    f = open("text.txt","r")

    print(f.read())

finally:

    f.close()

    print("File closed")

9. Simulate login with final log message:

try:

    username = input("Username: ")

    password = input("Password: ")

    if(username=="admin" and password =="1234"):

        print("Login successful")

    else:

        print("Invalid credentials.")

finally:

    print("Login attempt logged")

10. Divide with cleanup message:

try:

    a = int(input("enter num1: "))

    b = int(input("enter num2: "))

    res = a/b

    print(res)

except ZeroDivisionError:

    print("number cannot be divided by zero")

finally:

    print("clean up done")

Section D: Combined Exception Handling

11. Handle multiple exceptions + finally:

try:

    a = int(input("enter num1: "))

    b = int(input("enter num2: "))

    res = a/b

    print(res)

except ZeroDivisionError:

    print("number cannot be divided by zero")

except ValueError:

    print("Invalid input")

finally:

    print("clean up done")

12. Simulate bank withdrawal:

balance = 50000

try:

    amount = float(input("Enter the amount to be withdrawed"))

    if amount <= 0:

        raise ValueError("Amount must be positive.")

    if amount > balance:

        print("Insufficient balance.")

    else:

        balance -= amount

except ValueError as v:

    print("Error, ", v)

else:

    print("Withdrawal successful, balance = ", balance)

finally:

    print("Transaction done")