Exception Handling

Section A: Basic Try-Except

1. Divide two numbers, handle ZeroDivisionError:

```
try:
    a = int(input("Enter numerator: "))
    b = int(input("Enter denominator: "))
    result = a / b
    print("Result:", result)
except ZeroDivisionError:
    print("Error: Cannot divide by zero.")
```

2. Convert string to integer, handle ValueError:

```
try:
    s = input("Enter a number: ")
    num = int(s)
    print("Integer:", num)
except ValueError:
    print("Error: Invalid input, not a number.")
```

3. Add two numbers, handle invalid types:

```
try:
    a = int(input("Enter first number: "))
    b = int(input("Enter second number: "))
    print("Sum:", a + b)
except ValueError:
    print("Error: Please enter valid numbers.")
```

4. Access list index, handle IndexError:

```
try:
    lst = [10, 20, 30]
    index = int(input("Enter index: "))
    print("Element:", lst[index])
except IndexError:
    print("Error: Index out of range.")
```

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

5. Square of a number with try-except-else:

```
try:
    num = int(input("Enter a number: "))
except ValueError:
    print("Invalid number.")
else:
    print("Square:", num ** 2)
```

6. Open and read file with FileNotFoundError:

```
try:
    f = open("data.txt", "r")
except FileNotFoundError:
    print("File not found.")
else:
    print(f.read())
    f.close()
```

7. Convert to binary, handle invalid input:

```
try:
    num = int(input("Enter an integer: "))
except ValueError:
    print("Invalid input.")
```

```
else:
    print("Binary:", bin(num))
```

Section C: Try-Finally (5 marks each)

8. Ensure file closes with try-finally:

```
try:
    f = open("sample.txt", "w")
    f.write("Hello, World!")
finally:
    f.close()
    print("File closed.")
```

9. Simulate login with try and logging in finally:

```
try:
    user = input("Username: ")
    pwd = input("Password: ")
    if user == "admin" and pwd == "123":
        print("Login successful.")
    else:
        print("Login failed.")
finally:
    print("Login attempt logged.")
```

10. Division with error handling and cleanup:

```
try:
    a = int(input("Enter numerator: "))
    b = int(input("Enter denominator: "))
    print("Result:", a / b)
except (ZeroDivisionError, ValueError):
    print("Error occurred.")
```

```
finally:
    print("End of operation.")
```

Section D: Combined Exception Handling (6 marks each)

11. Multiple exception handling with finally:

```
try:
    a = int(input("Enter numerator: "))
    b = int(input("Enter denominator: "))
    print("Result:", a / b)
except ZeroDivisionError:
    print("Cannot divide by zero.")
except ValueError:
    print("Invalid number entered.")
finally:
    print("Execution complete.")
```

12. Simulate bank withdrawal with all clauses:

```
try:
    balance = 5000
    withdraw = int(input("Enter amount to withdraw: "))
    if withdraw > balance:
        print("Insufficient balance.")
    else:
        balance -= withdraw
except ValueError:
    print("Invalid amount entered.")
else:
    print("Withdrawal successful. Remaining balance:", balance)
finally:
    print("Transaction processed.")
```