

## Examples of try except

```
try:
    numerator = int(input("Enter the numerator: "))
    denominator = int(input("Enter the denominator: "))
    result = numerator / denominator
except ZeroDivisionError:
    print("Error: Cannot divide by zero!")
else:
    print("Result:", result)
finally:
    print("This block always executes.")
```

## Oral Questions:

1. What is the purpose of the try statement in Python?
2. Explain the role of the except block in exception handling.
3. When is the else block executed in a try-except construct?
4. Describe the function of the finally block in Python.
5. How do you handle specific types of exceptions in Python?
6. Can you have multiple except blocks in a single try statement? If so, how?
7. What happens if an exception occurs in the else block?
8. Is it mandatory to have an except block along with a try block?
9. Can you give an example of resource cleanup using the finally block?
10. Explain the sequence of execution in a try-except-else-finally construct.

### Coding Questions:

11. What is the purpose of the try statement in Python?
12. Explain the role of the except block in exception handling.
13. When is the else block executed in a try-except construct?
14. Describe the function of the finally block in Python.
15. How do you handle specific types of exceptions in Python?
16. Can you have multiple except blocks in a single try statement? If so, how?
17. What happens if an exception occurs in the else block?
18. Is it mandatory to have an except block along with a try block?
19. Can you give an example of resource cleanup using the finally block?
20. Explain the sequence of execution in a try-except-else-finally construct.