

---

# Lecture 11: Exception Handling in Object-Oriented Programming

## 1. Introduction

Exception handling is essential for building robust programs. It allows the program to respond gracefully to unexpected errors during execution.

## 2. Basics of Exception Handling in Python

Python uses `try`, `except`, `else`, and `finally` blocks to handle exceptions.

```
try:
    num = int(input("Enter a number: "))
    result = 10 / num
except ZeroDivisionError:
    print("Cannot divide by zero!")
except ValueError:
    print("Invalid input! Please enter a valid number.")
else:
    print(f"Result is {result}")
finally:
    print("Execution complete.")
```

## 3. Custom Exceptions

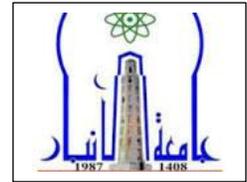
You can create your own exceptions by inheriting from the built-in `Exception` class.

```
class NegativeNumberError(Exception):
    pass

def check_positive(num):
    if num < 0:
        raise NegativeNumberError("Negative numbers are not allowed!")

try:
    check_positive(-5)
```

**Name:** Hamsa M Ahmed  
**Subject:** OOP Practical  
**Department:** Computer Network System



```
except NegativeNumberError as e:  
    print(e)
```

## 4. Exception Handling in Classes

Integrate exception handling within class methods to manage errors gracefully.

```
class BankAccount:  
    def __init__(self, balance=0):  
        self.balance = balance  
  
    def withdraw(self, amount):  
        try:  
            if amount > self.balance:  
                raise ValueError("Insufficient funds")  
            self.balance -= amount  
        except ValueError as e:  
            print(e)
```

## 5. Exercises

1. Modify the `BankAccount` class to handle invalid deposit and withdrawal amounts using exceptions.
2. Create a custom exception `InvalidAgeError` and use it in a `Person` class to validate age input.

## 6. Summary

Proper exception handling helps maintain program stability and improves user experience by handling errors without abrupt crashes.

## 7. References

1. Python Docs - Errors and Exceptions
2. Real Python - Exception Handling