

Object Oriented Programming (Using Python)

UNIT- II

Polymorphism and Inheritance: cont'd.

- Exceptions – Example Program using Inheritance

Prof. R. MADANA MOHANA

Professor, Artificial Intelligence & Data Science

<http://rmadanamohana.com/>

Exceptions

Example:

```
exceptions with inheritance.py - C:/Users/rmmna/AppData/Local/Programs/Python/Python310/exceptions with inheritance.py (3.10.0)
File Edit Format Run Options Window Help
class BankAccount:
    def __init__(self, account_number, balance):
        self.account_number = account_number
        self.balance = balance

    def withdraw(self, amount):
        if amount > self.balance:
            raise ValueError("Insufficient funds")
        self.balance -= amount

class SavingsAccount(BankAccount):
    def __init__(self, account_number, balance, interest_rate):
        super().__init__(account_number, balance)
        self.interest_rate = interest_rate

    def add_interest(self):
        interest = self.balance * self.interest_rate
        self.balance += interest
```

Exceptions

Example: cont'd.

```
try:
    account = SavingsAccount("123456789", 1000, 0.05)
    account.withdraw(1500)
except ValueError as e:
    print(e)
```

```
>>>
= RESTART: C:/Users/rmmna/AppData/Local/Programs/Python/Python310/exceptions wit
h inheritance.py
Insufficient funds
>>>
```

Exceptions

Example: cont'd.

In this example, we define a **BankAccount** class and a subclass **SavingsAccount**.

The **SavingsAccount** class has an additional method **add_interest()** that adds interest to the balance of the account.

We also define an **exception** to be **raised** if the user attempts to withdraw more money than is available in the account.

We use a **try-except** block to **catch** the **exception** and print out the **error message**.

This **example** demonstrates how **exceptions** can be used to **handle errors** in an **inheritance hierarchy**.