```python
# General error handling
try:
 result = 10/0
except Exception as e:
 print(f"An error occured {e}")

##Zero division error

## Value error

try:
 num = int("Hello")
except ValueError as e:
 print(f"Value error: {e}")

## TYPE ERROR

try:
 result = 10+ "Hello"
except TypeError as e:
  print(f"Type error: {e}")

## Indec Error

# lst =[1,2,3]
# print(lst[5])

## Key Error

try:
 dictionary = {"a":1 , "b": 2}
 print(dictionary["c"])
except KeyError as e:
 print("Key doesnot exist")


## FILE NOT FOUND ERROR
try:
 with open("file.txt", "r") as file:
  content = file.read()
except FileNotFoundError as e:
 print(f"File Not FOUND ERROR {e}")
```

```
##ATTRIBUTE ERROR
try:
 strng = "Hello"
 strng.append("World")
except AttributeError as e:
 print(f"{e}")


## IMPORT

try:
 import module1
except ImportError:
 print("Module not found")




## Assertion error

try:
 assert 5 > 10 , "Assertionfailed"
except AssertionError as e:
 print(f"Assertion error {e}")


try:
 num = int("Hello")
 result = 10/0
except (ValueError, ZeroDivisionError) as e:
 print(f"Error Occured {e}")
finally:
  pass



#### Custom exception ######
# User defined exception class and inherits from python's built in
Exception class

class CustomError(Exception):
 pass
try:
```

```
  raise CustomError("This is a custom error")
except CustomError as e:
 print(f"Custom Error {e}")

class BankAccount:
    def __init__(self, account_holder, initial_balance):
        self.account_holder = account_holder
        self.balance = initial_balance

    def withdraw(self, amount):
        """Handles withdrawal with custom exception handling"""
        if amount <= 0:
            raise InvalidAmountException()

        if amount > self.balance:
            raise InsufficientFundsException(self.balance)

        self.balance -= amount
        print(f"Withdrawal successful! New balance:
${self.balance:.2f}")

    def get_balance(self):
        """Returns the current balance"""
        return self.balance


def main():
    account = BankAccount("John Doe", 500.00)

    try:
        amount = float(input("Enter withdrawal amount: "))
        account.withdraw(amount)
    except InvalidAmountException as e:
        print(f"Error: {e}")
    except InsufficientFundsException as e:
        print(f"Error: {e}")
    except Exception as e:
        print(f"An unexpected error occurred: {e}")
    finally:
        print("Thank you for using our banking system.")

if __name__ == "__main__":
    main()
```