

# 暨南大学本科实验报告专用纸

课程名称 python程序设计 成绩评定           

实验项目名称 python作业第七章

姓名 崔嘉容 学号 2020100069 学院 网络空间安全 专业 网络空间安全

实验时间 2023 年 10 月 30 日 ~ 11 月 6 日 实验地点: 516

## 一、实验目的

1. 了解掌握python异常处理机制。
2. 能够自定义异常处理类。
3. 能够将异常处理熟练地应用到实践中。

## 二、实验环境和设备

实验环境：操作系统-Windows10，python版本-3.11.3，开发环境-pycharm

实验设备：华为MateBook14-2020，处理器-i7-10510U，内存-16GB

## 三、实验内容和结果

题目一：1. 编写一个函数以计算5/0并使用try / except捕获异常。

实验代码：

```
def divide_by_zero():
    try:
        result = 5/0
        return result
    except ZeroDivisionError as e:
        return f"Error: {e}"

result = divide_by_zero()
print(result)
```

# 暨南大学本科实验报告专用纸(附页)

## 实验结果:

A screenshot of a terminal window showing a Python error. The window title is "1\_error0 x". The command executed is "E:\Python3.11\python.exe D:\学习\大四上\python\_learn\homework\_chapter7\1\_error0.py". The output is "Error: division by zero".

```
1_error0 x
E:\Python3.11\python.exe D:\学习\大四上\python_learn\homework_chapter7\1_error0.py
Error: division by zero
```

## 实验分析和总结:

在这个代码中,定义了一个名为 `divide_by_zero` 的函数。在该函数内部,通过尝试计算 `5/0`,这会引发一个 `ZeroDivisionError` 异常。然后使用 `try / except` 块来捕获这个异常,并返回一个相应的错误消息。当调用 `divide_by_zero()` 时,它会返回错误消息 "Error: division by zero"。

除零是一个会引发 `ZeroDivisionError` 异常的操作。在实际编程中,应该避免这样的操作,或者在可能发生除零的情况下,事先进行检查以避免异常的发生。

# 暨南大学本科实验报告专用纸(附页)

**题目二:** 定义一个自定义异常类，该类将字符串消息作为属性。

**实验代码:**

```
class CustomException(Exception):
    def __init__(self, message):
        self.message = message
        super().__init__(self.message)

try:
    raise CustomException("This is a custom exception message.")
except CustomException as e:
    print(e.message)
```

**实验结果:**



**实验分析和总结:**

首先定义了一个名为 CustomException 的自定义异常类，它继承自内置的 Exception 类。在 CustomException 类中，定义构造函数 \_\_init\_\_，它接受 message 参数，并将其存储为实例的 message 属性。在 try 块中，使用 raise 关键字抛出 CustomException 的实例，并传递了一个自定义的消息作为参数。在 except 块中，捕获了这个自定义异常，并打印了它的消息属性。这样实现成功地定义了一个自定义异常类，并成功地使用它来抛出和捕获异常。

# 暨南大学本科实验报告专用纸(附页)

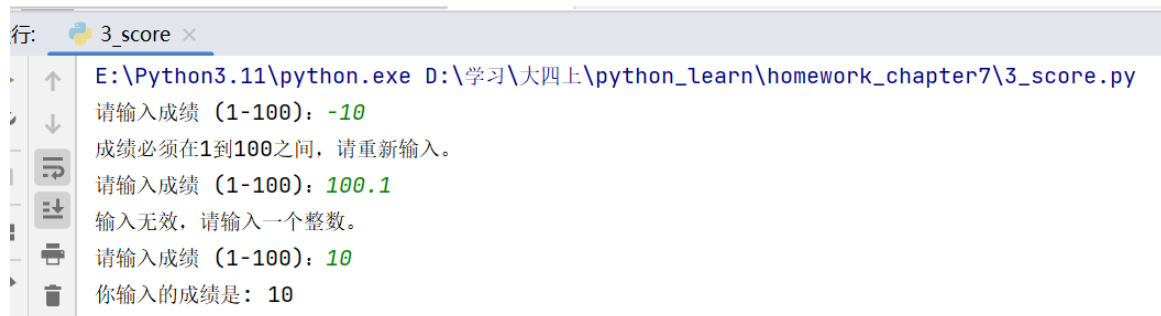
**题目三：**定义函数,在控制台中获取成绩(1-100)，如果输入有误,请重新输入。（使用异常处理）

## 实验代码：

```
def get_valid_score():
    while True:
        try:
            score = int(input("请输入成绩 (1-100): "))
            if 1 <= score <= 100:
                return score
            else:
                print("成绩必须在 1 到 100 之间，请重新输入。")
        except ValueError:
            print("输入无效，请输入一个整数。")

score = get_valid_score()
print(f"你输入的成绩是: {score}")
```

## 实验结果：



```
行: 3_score x
E:\Python3.11\python.exe D:\学习\大四上\python_learn\homework_chapter7\3_score.py
请输入成绩 (1-100): -10
成绩必须在1到100之间，请重新输入。
请输入成绩 (1-100): 100.1
输入无效，请输入一个整数。
请输入成绩 (1-100): 10
你输入的成绩是: 10
```

## 实验分析和总结：

在这段代码中，通过定义一个名为 `get_valid_score` 的函数，使用一个无限循环来反复尝试获取输入，直到用户提供了有效的成绩。

在 `try` 块中，将用户输入转换为整数。如果成功，我们检查该成绩是否在有效的范围内（1 到 100）。如果是，则返回该成绩；否则，提醒用户重新输入。如果用户提供的不是一个整数，会触发 `ValueError` 异常，在 `except` 块中捕获并提醒用户重新输入。当用户提供有效的成绩后，它将被返回并打印出来。如果用户输入的成绩不在 1 到 100 的范围内，程序会一直要求用户重新输入，直到提供有效的成绩。

# 暨南大学本科实验报告专用纸(附页)

## 题目四: 编写一个简易登录程序,

(1) 当登录用户不再用户列表时, 引发异常, 并将登录用户名以及登陆时间写入warnlog.txt中。

日志格式如下: [error] username is not exist- time

(2) 当登录名在用户列表中时, 若用户密码输错三次, 将用户登录锁定, 然后引发异常, 并将用户名、登陆时间和异常原因写入warnlog.txt中。

[waring] username-Password error three times-time

(3) 用户列表文件自己定义。

### 实验代码:

```
import getpass

def read_user_list(filename):
    user_list = {}
    with open(filename, 'r') as file:
        for line in file:
            username, password = line.strip().split(',')
            user_list[username] = password
    return user_list

def write_log(log_message):
    with open('warnlog.txt', 'a') as file:
        file.write(log_message + '\n')

def login(username, password):
    user_list = read_user_list('user_list.txt')
    if username not in user_list:
        log_message = f"[error] {username} is not exist - {time.strftime('%Y-%m-%d %H:%M:%S')}"
        write_log(log_message)
        raise Exception(f"User '{username}' does not exist.")

    if user_list[username] == password:
        print(f"Welcome, {username}!")
    else:
        password_wrong = False # 添加一个布尔变量来标记密码是否输入错误
        for _ in range(3):
            password_attempt = getpass.getpass("请输入密码: ")
            if user_list[username] == password_attempt:
```

# 暨南大学本科实验报告专用纸(附页)

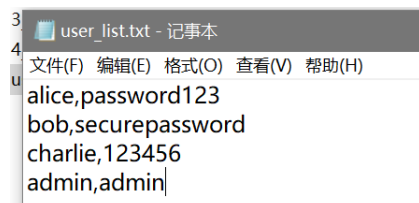
```
        print(f"Welcome, {username}!")
        return
    else:
        password_wrong = True

    if password_wrong:
        log_message = f"[warning] {username}-Password error three
times - {time.strftime('%Y-%m-%d %H:%M:%S')}"
        write_log(log_message)
        print(f"User '{username}' locked due to three incorrect
password attempts.")
        return

import time

try:
    username = input("请输入用户名: ")
    password = getpass.getpass("请输入密码: ")
    login(username, password)
except Exception as e:
    print(e)
```

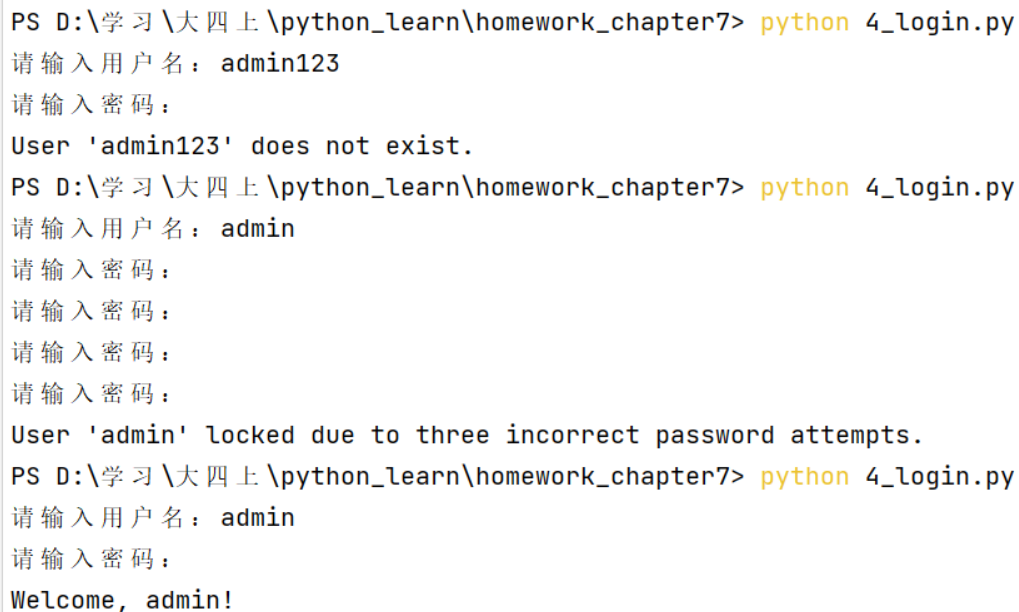
## 实验结果:



user\_list.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

alice,password123  
bob,securepassword  
charlie,123456  
admin,admin



```
PS D:\学习\大四上\python_learn\homework_chapter7> python 4_login.py
请输入用户名: admin123
请输入密码:
User 'admin123' does not exist.
PS D:\学习\大四上\python_learn\homework_chapter7> python 4_login.py
请输入用户名: admin
请输入密码:
请输入密码:
请输入密码:
请输入密码:
User 'admin' locked due to three incorrect password attempts.
PS D:\学习\大四上\python_learn\homework_chapter7> python 4_login.py
请输入用户名: admin
请输入密码:
Welcome, admin!
```

# 暨南大学本科实验报告专用纸(附页)

```
3 warnlog.txt - 记事本
4 文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
1 [error] admin123 is not exist - 2023-11-06 09:31:18
N [warning] admin-Password error three times - 2023-11-06 09:31:28
```

## 实验分析和总结:

`read_user_list` 函数用于从文件 `user_list.txt` 中读取用户列表, 该文件中每行格式为 `username,password`。

`write_log` 函数用于将日志消息写入 `warnlog.txt` 文件中。

`login` 函数尝试进行登录验证。如果用户名不存在, 将会写入错误日志并引发异常。如果密码错误三次, 将会写入警告日志并锁定用户。如果用户不存在或密码错误三次, 程序将会按照你所要求的处理并记录日志。

利用之前学到的`getpass`方法, 在控制台中实现了不显示密码的登录方式, 一定程度上保护了用户的隐私。