第七章

题目一（类的定义与调用）：

1、构造一个银联信用卡的类：

- 包含如下属性：

  顾客姓名

  信用卡授信额度

  当前额度

  单次刷卡金额上限

- 包含如下方法：

分别获得上述属性的方法；

对授信额度进行修改的方法；

对单次刷卡金额上限修改的方法；

实现刷卡方法，传入一个刷卡金额，先判断是否超过单次刷卡金额上限以及当前额度是否够用，如合理，则执行刷卡，将当前额度减去刷卡金额。

题目二（类的继承）

2、通过继承银联信用卡的类，构造中国银行信用卡的类：

- 实现对银联信用卡类的继承；
- 新增加属性：中国银行信用卡积分、优惠店铺列表；
- 重写刷卡方法：

传入消费店铺名称和消费金额，如果店铺名称在优惠店铺列表中，则刷卡金额打95折；

每消费10元，信用卡积分增加1分；

保留父类刷卡方法的其他功能。

- 新增如下方法：

获得用户积分的方法；

设置优惠店铺列表的方法。

第八章

题目一（文件的读写）：

1、读取文件“笑傲江湖.txt”，统计文本中所有字符的出现频次，将结果按如下格式保存到文件“笑傲江湖_字符统计.txt”。

- 令：3904，狐：3904，冲：4728

题目二（异常处理）

2、通过input输入数字的时候，我们通常使用eval(input(“请输入一个数字”))来实现。

- 考虑该代码可能出现的错误;
- 应该如何应对?

题目三（模块）：

3、成功实现一个模块的导入及调用

- 构造一个筛选n以内所有素数的函数，将其保存在screening_prime.py文件中。
- 新建一个py文件，导入screening_prime模块中筛选素数的函数，并成功调用，获得1000以内的所有素数。

第九章

题目一 简答题（数据类型的底层实现）

以下两段程序分别会输出怎样的结果？请运行验证，并给出理由

代码1：

```
def f1(ls=[]):
    ls.append(1)
    return ls


print(f1())
print(f1())
print(f1())
```

代码2：

```
person = {"name": "", "id": 0}
team = []

for i in range(3):
    x = person
    x["id"] = i
    team.append(x)

team[0]["name"] = "Peter"
team[1]["name"] = "Paul"
team[2]["name"] = "Mary"

print(team[1])
print(team)
```

题目二 编程题（更加简洁的语法）

1、用列表推导实现一个过滤器

- 实现一个判断数字n是否为素数的函数isprime(n)；
- 利用列表推导，获得100以内的素数列表；

2、使用列表推导，实现两列表对应元素的操作

- 获得新列表z，列表内元素

z[i] = pow(x[i], y[i])

x = [1，2, 3, 4]

y = [0，2, 3, 1]

3、使用条件表达式，将x，y 中的最大值赋值给z

题目三 编程题（三大神器）

1、构造一个生产n 以内素数的生成器，与编程题2-1 进行相互验证。

2、某大型网上购物网站进行中秋节优惠促销活动，请在不改变原计费函数 charge()的基础上，输出“中秋节快乐!”，并实现总价打8 折的优惠。

□ 实现原计费函数 charge(商品名称, 商品数量)。

□ 假设只有三种商品，商品名称和单价存储在字典{“water”：1.5，“egg”: 1，“meat”: 15}中；

□ 假设每次只购买一种商品，购买数量1 ~ 5 件, 不打折；购买数量6 ~ 10 件，打95 折；购买数量大于10 打9 折；

□ 返回商品应付总货款。

- 在不改变原计费函数源代码和调用方式的前提下，增加如下功能：

□ 每次 charge 函数被调用时，输出“中秋节快乐!”；

□ 在原计费函数 charge 返回总价的基础上，对总价打8 折。

第十章

题目一 模拟经典的“三门问题”

题目二 求解经典的“24点问题” ¶

第十一章

题目一 简单编程题（numpy数组的创建和性质）

1、创建一个4*6的二维数组，元素由0~1之间均匀分布的随机数构成，输出其形状、大小、维度和数据类型。

2、将（1）中的二维数组分别转换为3*8的数组和摊平为一维数组。

3、创建2个4*4的二维数组，元素为整数，随机取自区间[0, 10)，请将两个数组分别进行水平拼接和垂直拼接。

4、创建一个16*16的数组，其中，数组的外围均为1，内部均为0，提示：用到切片赋值。 ¶

题目二 简单编程题（numpy四大运算和通用函数）

5、创建一个由10000个元素构成的一维数组，元素随机取自[0, 100)

- 求数组元素的均值、方差和标准差；
- 通过 (x-x.mean)/x.std的方法，将数组归一化。
- 求数组的最大值和最小值，并获取其索引位置；
- 通过 (x-x.min)/(x.max-x.min)的方法，将数组归一化。

6、创建一个由30个元素构成的一维数组，元素随机取自[80, 100)

- 获得大于等于95分的成绩；
- 获得小于等于85分的成绩；
- 获得大于85分，小于95分的成绩；
- 对成绩进行排序。

7、构建两个3*3的二维列表A和B

（1）乘法运算

- 用Python for循环实现A*B的乘法运算；
- 用Numpy 实现A*B 的乘法运算；
- 对比两种实现方式的运行时间。

（2）矩阵乘法

- 用Python for循环实现A和B的矩阵乘法；
- 用Numpy实现np.dot(A, B)的矩阵乘法；
- 对比两种实现方式的运行时间。

（3）对比矩阵乘法跟乘法运算的不同。