Question 1

#a

d= {}

L1 = ['HTTP',***'HTTPS'***,'FTP','DNS']

L2 = [80,433,21,53]

For r, z in zip(L1, L2):

d[r]=z

Print(d)

#b

def factorial(a):

**if** a ==0:

**return** 1

**else**:

**return** a\*factorial(a\_1)

num =**int**(input(Enter a number))

result=factorial(num)

print("The factorial of {num} is {result}")

#c

L = [***'Network'***,'Bio',***'programming'***, ***'Physics'***,***'Music'***]

r = 0

**for** r in range(len(L)):

**if** L[r].startswith("B"):

print(L[r])

#d

d = {g:g+1 **for** g in range(0,11)}

print(d)

Question 2

r\_num = list(input("Input a binary number:"))

v = 0

**for** z in range(len(r\_num)):

digit = r\_num.pop()

**if** digit =='1':

v = v + pow(2,z)

print("The decimal value of the number is",v)

Question 3

#load question and answer from the csv file

**import** csv

def load\_quiz\_data(file\_path):

quiz\_data = []

with open(file\_path,'r') as file:

csv\_reader = csv.reader(file)

**for** row in csv\_reader:

question, answer = row

quiz\_data.append((question, answer))

**return** quiz\_data

#Ask question to the user and calculate the score

def take\_quiz (quiz\_data):

score =0

**for** question,answer in quiz\_data:

user\_answer =input(question +" ")

**if** user\_answer.lower() ==answer.lower():

score +=1

**return** score

#Save user name and result to a csv file

def save\_user\_result (user\_name,score):

with open(***'user\_result.csv'***,'a',newline =") as file***:***

csv\_writer = csv.writer(file***)***

csv\_writer.writerow([user\_name, score])

#Main functio***n***

def main()***:***

file\_path = 'quiz\_data.csv***'***

quiz\_data = load\_quiz\_data(file\_path)

user\_name = input("Enter your name :")

score = take\_quiz(quiz\_data)

print("your score :***{***score***}***" ***)***

save\_user\_result(user\_name,score)

if\_name\_ == "\_main\_"***:***

main()

Question ***4***

class BankAccount***:***

def\_init\_(self,account\_number, account\_holder,balance = 0.0)***:***

self.accunt\_number =account\_numbe***r***

self.account\_holder = account\_holde***r***

self.balance =balanc***e***

def deposit(self,amount)***:***

self.balance +=amoun***t***

return self.balanc***e***

def withdraw(self,amount)***:***

if amount > self.balance***:***

print("Insufficient funds")

else***:***

self.balance = amoun***t***

return self.balanc***e***

def get\_balance(self)***:***

return self.balanc***e***

class savings Account(Bank Account)***:***

def\_init\_(self,account\_number, account\_holder, balance =0.0,interest\_rate =0.0)***:***

super().\_init\_(account\_number, account\_holder,balance)

self.interest\_rate = interest\_rat***e***

def apply\_interest (self)***:***

interest\_amount = self.balance\*self.interest\_rat***e***

self.deposit(interest\_amount)

return interest\_amoun***t***

def print\_details(self) ***:***

print("Account Holder:***{***self.account\_holder***}***"***)***

print("Account Number: ***{***self.account\_number***}***"***)***

print("Balance: ***{***self.balance***}***"***)***

print("Interest Rate: ***{***self.interest\_rate***}***"***)***

#Create an instance of Bank Accoun***t***

Bank\_account = BankAccount("123456789","joel"***)***

Print("Bank Account\_Initial Balance:", bank account.get\_balance()***)***

Bank\_account: deposit(1000***)***

Print("Bank Account\_Balance after deposit:", bank\_account.get\_balance()***)***

bank\_account.withdraw(500***)***

print("Bank Account\_Balance after withdraw) :", bank\_account.get.balance()***)***

#Create aninstance of savings Accoun***t***

Savings\_account = savings Account("987654321,"Jana", interest\_rate =0.05***)***

print("\n savings Account\_Initial Balance and Rate:")

savings\_account.print\_details()

savings\_account.apply\_interest()

print("\n savings Account\_Balance after applying interest:")

savings\_account.print\_details()