Task 1

The favorite\_languages dictionary contains student names and programming language

names that each student prefers:

Programming\_Languages = {  
 'George': ['Java', 'JavaScript'],  
 'David': ['C#', 'Ruby', 'JavaScript']  
}

In the Python programming language, write a loop that displays the name of each student

from the favorite\_languages dictionary and the names of the his/her favorite programming

languages as follows:

Favorate language of student Jen is python

Favorate language of student Jen is c

. . . . . . .

Task 2.

We have two lists:

Users = ['Andrew', 'Carolina', 'David', 'John', 'Clark', 'Alice']

Privileged\_Users = ['Andrew', 'Carolina', 'David', 'George']

In the Python programming language, write software code that takes each item from the

Users list one by one and checks whether it is a member of the second Privileged\_Users list.

Task 3.

Modify the software code obtained in the previous task so that the elements of the Users list

that are in the Privileged\_Users list are placed separately in the new Sublist\_Privileged\_Users

list. Calculate the number of elements in Sublist\_Privileged\_Users list.

Task 4

In the Python programming language, write a function that has two parameters var1, var2.

When calling a function, var1 parameter should be assigned the numeric value and var2

must be assigned a list of integer numbers. The function must select an odd numbers from

the list of transmitted integers that are greater than var1 and then return it as a separate

list.

Task 5

Create a class - Person, which will have the following member variables:

**country**, **name** and **age**;

Class Person must contain a constructor \_\_init \_\_ (), which will initialize member variables.

Task 6

In the Personclass add method details(self) , which returns the following string when called:

**Nam**e is **age** years old and he / she is from **Country**.

For example, if the Instance person1 of the Person class is created whose country, name and age

member variables will be defined Georgia, David, 23

respectively, then after calling of the details (self) method should return the following string:

**David** is **23** years old and he / she is from **Georgia**.

Task 7

Create a class Student that will be inherited from the Person class and additionally will have

one aditional member variable - **university**. In the Student class, create a class constructor

that uses the parent class constructor.

Task 8

In order to create the person1 and student1 objects of the Person and Student Classes use

Person and Student class constructors and pass the following arguments:

Person class Constructor: name=’John Doe’, age=23, country=Georgia;

Student class Constructor: name=’John Doe’, age=23, country=Georgia, university=’UG’;

Use person1 and student1 and call details(self) methods: