



Midterm Exam: 30%

Course Identification

Name of programs – Codes:	COMPUTER SCIENCE TECHNOLOGY– PROGRAMMING (420.BP) INFORMATION TECHNOLOGY PROGRAMMERANALYST (LEA.3Q)
Course title:	SCRIPTING LANGUAGES
Course number:	420-LS3-AS
Group:	7178
Teachers' names:	Renan Cavalcanti
Duration:	2 hours (120 minutes)
Semester:	Fall 2023

Student Identification

Name: _____

Student number: _____

Date: October 27 , 2023

Result: _____

☒ I declare that this is an original work, and that I credited all content sources of which I am not the author (online and printed, images, graphics, films, etc.), in the required quotation and citation style for this work.

Standard of the Evaluated Competencies

Statement of the evaluated competency – Code

Use programming languages - 00Q2

Evaluated elements of the competencies

1. Analyze the problem.
2. Translate the algorithm into a programming language.
3. Debug the code.
4. Implement the functional test plan.

Evaluated Performance Criteria

- 1.1 Correct breakdown of the problem
 - 1.2 Proper identification of input and output data and of the nature of the processes
 - 1.3 Appropriate choice and adaptation of the algorithm

 - 2.1 Appropriate choice of instructions and types of elementary data
 - 2.2 Efficient modularization of code
 - 2.3 Logical organization of instructions
 - 2.4 Compliance with the language syntax
 - 2.5 Computer code consistent with the algorithm

 - 3.1 Efficient use of the debugger
 - 3.2 Identification of all errors
 - 3.3 Astute choice of debugging strategies
 - 3.4 Relevance of the corrective actions
 - 3.5 Clear record of solutions to the problems encountered

 - 4.1 Attitudes and behaviors that demonstrate thoroughness
 - 4.2 Identification of all operational errors
 - 4.3 Relevance of the corrective actions
 - 4.4 Proper functioning of the program
 - 4.5 Clear record of information concerning tests and their results
- Critical thinking, methodical, analytic and synthetic;
 - Programming efficiency;
 - Autonomy, initiative.
 - Time management

Instructions

- No break is allowed in this exam. Students are not allowed to exit the examination room before half of the allotted time has passed. Once a student has exited the classroom, he or she may not re-enter. (PIEA – Article 5.12.4)
- Internet access is not allowed during the exam.
- Class notes are not allowed.
- The teacher will not answer questions during the final exam.
- Students must keep silent during the exam time.
- It is the teacher's responsibility to identify language errors. If such errors are found, the teacher has the right to apply a penalty of up to 20% of the grade. (PIEA – Article 5.7)
- Plagiarism, any attempt at plagiarism or complicity in plagiarism during an evaluation representing 20% and more of the final grade, will result in a course failure. (PIEA – Article 5.16).
- Wait for the teacher's signal to start the exam.

Mark Breakdown

Questions 1 to 6 -> 20 points (Choose 5)

Questions

Developing Application with Python

- 1) Create a Python program that asks the user for their hourly wage and the number of hours they work per week. The program should calculate and return the annual gross income, taxes based on the tax brackets, and the net income. The program should continue running, allowing the user to enter new values until they enter "0" for the hourly wage. Assume a standard year of 52 weeks for the calculations.

The tax brackets are as follows:

- 10% on the first \$30,000 of taxable income
- 15% on the next \$30,000 of taxable income (from \$30,001 to \$60,000)
- 20% on the next \$40,000 of taxable income (from \$60,001 to \$100,000)
- 25% on the taxable income over \$100,000

- 2) There is an array of integers. There are also 2 **disjoint sets**, A and B, each containing integers. You like all the integers in set A and dislike all the integers in set B. Your initial happiness is 0. For each integer in the array, if the integer is also in the set A, you add 1 to your happiness. If the integer is in the set B, you add -1 to your happiness. Otherwise, your happiness does not change. Output your final happiness at the end.

Note: Since A and B are sets, they have no repeated elements.

However, the array might contain duplicate elements.

A = {1, 2, 3, 7, 9, 12, 15, 18, 21, 24}

B = {4, 5, 6, 10, 11, 13, 16, 19, 22, 25}

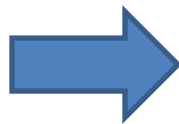
my_list = [1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 7, 9, 12, 15, 18, 21, 24, 10, 11, 13, 16, 19, 22, 25, 1, 2, 3, 4, 5]

3) Given a nested list with name and last name from a person:

people_list = [['John', 'Doe'], ['Jane'], ['Michael', 'Smith']]

We could use the following code to access it:

```
for people in people_list:
    for name in people:
        print(name, end=' ')
    print()
```



```
John Doe
Jane
Michael Smith
```

Write a code in python to access the list but using WHILE only to have the same output.

4) Create a function in python that will receive a list as a parameter and will return a nested dictionary where each element will be the key of one level deeper and the last element will be the value of the last key.

Ex.

Input: [1, 2, 3, 4, 5, 6]

Output: {1:{2: {3: {4: {5: 6}}}}}

- 5) Given a string from the user, create a python program that will take this string and show in a dictionary how many times each combination of word appears in that string. All the combinations with the same size must be inside a dictionary where the key is the amount of letters.

Ex: input -> Weekend

Output: {1: {'W': 1, 'e': 3, 'k': 1, 'n': 1, 'd': 1}, 2: {'We': 1, 'ee': 1, 'ek': 1, 'ke': 1, 'en': 1, 'nd': 1}, 3: {'Wee': 1, 'eek': 1, 'eke': 1, 'ken': 1, 'end': 1}, 4: {'Week': 1, 'eeke': 1, 'eken': 1, 'kend': 1}, 5: {'Weeke': 1, 'eeken': 1, 'ekend': 1}, 6: {'Weeken': 1, 'eekend': 1}, 7: {'Weekend': 1}}

- 6) Using the program attached, the program is used to calculate the time for the user to type each word.

Modify the code and before start the game ask the user for the level.

Level 1 if the user types a word in more than 8s the program should stop and show “game over”, only words up to size 6 are allowed in this level.

Level 2 if the user types a word in more than 8s the program should stop and show “game over”, only words up to size 10 are allowed in this level.

Level 3 if the user types a word in more than 8s the program should stop and show “game over”, all the words are allowed in this level.