

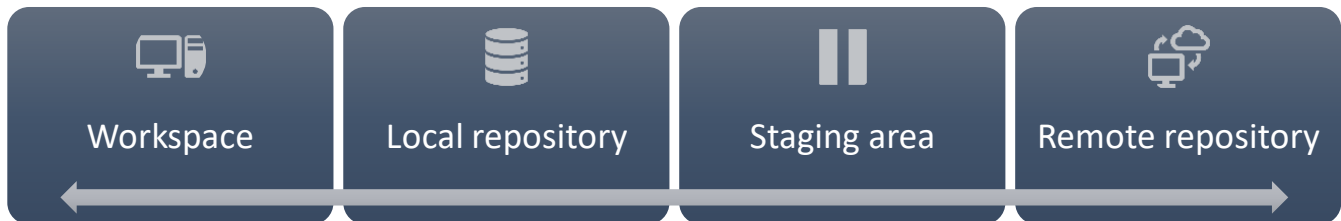
Student: \_\_\_\_\_ Student ID#: \_\_\_\_\_

## MIDTERM EXAM

### README

1. The answers to the questions in this exam must be your own original work.
2. Sign every page of this test with your name and NAU id.
3. Read the instructions and questions carefully. Correctly interpreting the questions is part of the examination.
4. The exam starts at 2:20 PM and ends at 3:35 PM. If you write in your exam after time runs up, your exam may be taken from you, and you'll receive a zero as a grade.
5. Phones, laptops, or any other electronic devices are not allowed during exam time. Please turn your phones off and store them out of your reach. If you use any devices during exam time, your exam may be taken from you, and you'll receive a zero as a grade.
6. Provide your answer using a non-erasable pen. If you use pencil or erasable pen, you cannot request grade review.
7. The exam is individual and closed book, closed notes.
8. Your grade will appear on BBLearn as soon as all the tests are graded.

**(20 points) Question 1 - Git/GitHub:** Git is a version control system that allows multiple people to work on the same repository. To allow this, Git works with four core areas: the workspace, the staging area, the local repository, and the remote repository. See the image below:



Write one **example** of a git command that represents the interaction between each of the areas listed below. Follow the example:

Remote repository ⇔ workspace: \_\_\_\_\_ git pull \_\_\_\_\_

Workspace ⇔ staging area: \_\_\_\_\_

Staging area ⇔ local repository: \_\_\_\_\_

Local repository ⇔ remote repository: \_\_\_\_\_

Workspace ⇔ local repository: \_\_\_\_\_

## INF 502 – Software Development Methodologies

Instructor: Dr. Ana Paula Chaves

**(28 points) Question 2 – Python basics:** A Python programmer wants to create a program to calculate the nth term of a Geometric Progression and/or of an Arithmetic Progression. The program asks the user which progression they want to calculate and print out the result. The code is written below but some portions are missing. Fill in the blanks that completes the code. Use the code documentation (next page) as helpful hints. Follow the instructions below:

1. **Line 1:** define the function `get_term_GP()`
2. **Line 2:** check the given GP formula and make the necessary adjustments to perform the correct calculation (considering that the PG formula is  $A_n = a_1 \cdot q^{n-1}$ )
3. **Line 8:** write a statement to read the variable `option` from the user input.
4. **Line 9:** write a conditional statement that decides whether the informed `option` is a valid value.
5. **Lines 14, 19 and 24:** call the appropriate functions. Make sure you capture the returns in the appropriate variable when needed.

```
1  _____
2      An =    a1    *    q    **    n    -    1
3      return An

# get_term_AP() function: assume that it is implemented here and working correctly.

4  def main():
5      print("What progression you want to calculate?")
6      print("Type 1 for the n-th term of an A.P.")
7      print("Type 2 for the n-th term of a G.P.")

8      option = _____

9      _____#test if option is valid
10         if option == 1: #the user wants to calculate the A.P.
11             a1 = int(input("Inform the first term of the A.P. (a1): "))
12             n = int(input("Inform the position of the term to be found (n): "))
13             r = int(input("Inform the common difference (d): "))

14         _____#call the appropriate function
15         else: #the user does NOT want to calculate the A.P.
16             a1 = int(input("Inform the first term of the G.P. (a1): "))
17             n = int(input("Inform the position of the term to be found (n): "))
18             r = int(input("Inform the ratio of the G.P. (q): "))

19         _____#call the appropriate function
20             print("The n-th termo is: ", An)
21         else: #option is invalid
22             print("Option is invalid. Please type 1 for A.P. or 2 for G.P.")
23             print("Try again soon!")

24         _____#Begin the program
```

## INF 502 – Software Development Methodologies

**Instructor:** Dr. Ana Paula Chaves**Code 1. Fill in the blanks****Function get\_term\_GP()**

Parameters: a1 – first term of the GP (integer)

n – position of the term to be calculated (integer); e.g., if n=10, the function returns the 10th term of the GP

q – quotient of the GP (integer)

Return: the n-th term of the GP

**Function get\_term\_AP()**

Parameters: a1 – first term of the AP (integer)

n – position of the term to be calculated (integer); e.g., if n=10, the function returns the 10th term of the AP

d – common difference of the terms in the AP (integer)

Return: the n-th term of the AP

**Code documentation****(10 points) Question 3 – Exception:** What will be the output of the following code? Briefly explain your answer.

```
1 def myFunction(number):
2     try:
3         y = 1/number
4     except ZeroDivisionError:
5         y = 0
6     else:
7         print("else")
8     finally:
9         return y
10 print(myFunction(2))
```

---

---

---

---

---

---

---

---

**(10 points) Question 4 – Files:** Briefly explain what the following code do. What will be the output?

```
1 dna_sequence = ['A', 'T', 'T', 'G', 'A',
2 'C']
3 myfile = open('dna.txt', 'w')
4 for nucleotides in dna_sequence:
5     myfile.write(nucleotides+"\n")
6 myfile.close()
7
8 myfile = open('dna.txt')
9 print(myfile.read())
10 myfile.close()
```

---

---

---

---

---

---

---

---

**(20 points) Question 5 – Object-orientation:** A programmer wishes to create Earthquake objects to make it easier to manipulate the information about a single earthquake. They created the following Python class. However, when they tested the class, they realized they made some mistakes. Find the **four** lines with errors in the code and fix them.

## INF 502 – Software Development Methodologies

**Instructor:** Dr. Ana Paula Chaves

**Important note:** leave the line blank if you don't want to change them. Any line that is written will be considered as an attempt to fix an error. If there are no errors in the line, it will be marked as an incorrect attempt. DO NOT REWRITE THE ENTIRE CLASS!

1 class Earthquake:	1 _____
2     def __init__():	2 _____
3         self.__country = ""	3 _____
4         self.__place = ""	4 _____
5         self.__mag = 0	5 _____
6         self.__tsunami = False	6 _____
7     def __str__(self):	7 _____
8         return self.__country + " <" +	8 _____
self.__place + ", " + str(self.__mag) + ",	_____
" + str(self.tsunami) + ">"	_____
9     def record_measurement(self, country,	9 _____
place, mag, tsunami):	_____
10         self.__country = country	10 _____
11         self.__place = place	11 _____
12         self.__mag = magnitude	12 _____
13         self.__tsunami = tsunami	13 _____
15     def getCountry(self):	15 _____
16         return self.__country	16 _____
17     def getPlace(self):	17 _____
18         return self.__place	18 _____
19     def getMagnitude(self):	19 _____
20         return self.__mag	20 _____
21     def getTsunami(self):	21 _____
22         print self.__tsunami	22 _____
23 eq = Earthquake()	23 _____
24 eq.record_measurement("Taiwan", "66 km	24 _____
E of Hualien City", 6.2, False)	_____
25 print(eq)	25 _____

**INF 502 – Software Development Methodologies**  
**Instructor: Dr. Ana Paula Chaves**

**(12 points) Question 6 – Dictionaries:** Consider the following dictionary:

```
earthquake = {"Taiwan":[{"mag": 6.2, "place": "66 km E of Hualien City", "tsunami": False}],  
              "Indonesia":[{"mag":6.1,"place":"65 km NNW of Bukittinggi","tsunami": False},  
                           {"mag": 6.6, "place": "80 km SW of Labuan", "tsunami": False}]}
```

Fill the blanks to create a code that prints all the magnitudes of the given earthquakes.

```
1 for _____ in earthquake:  
2     for eq in _____:  
3         print(eq[_____])
```