

Assessment – 1

NAME	SURARAPU MANIKANTA
ROLL NO	20T91A0585
COLLEGE NAME	GIET ENGINEERING COLLEGE
EMAIL	manisurarapu99@gmail.com

1. Write a Python program to calculate the area of a rectangle given its length and width

Program:

```
def calculate_rectangle_area(length, width):  
    area = length * width  
    return area  
  
def main():  
    length = float(input("Enter the length of the rectangle: "))  
    width = float(input("Enter the width of the rectangle: "))  
    area = calculate_rectangle_area(length, width)  
    print("The area of the rectangle is:", area)  
  
main()
```

Output:

```
Enter the length of the rectangle: 5  
Enter the width of the rectangle: 6  
The area of the rectangle is: 30.0
```

2. Write a program to convert miles to kilometres.

Program:

```
def miles_to_kilometers(miles):  
    kilometers = miles * 1.60934  
    return kilometers  
  
def main():  
    miles = float(input("Enter the distance in miles: "))  
    kilometers = miles_to_kilometers(miles)  
    print(f"{miles} miles is equal to {kilometers} kilometers.")  
  
main()
```

Output:

```
Enter the distance in miles: 55
55.0 miles is equal to 88.5137 kilometers.
```

Process finished with exit code 0

3. Write a function to check if a given string is a palindrome

Program:

```
def is_palindrome(s):
    s = s.lower()
    s = ''.join(char for char in s if char.isalnum())
    return s == s[::-1]

string = input("Enter a string:")
print(is_palindrome(string))
```

Output:

```
Enter a string:madam
True
```

Process finished with exit code 0

4. Write a Python program to find the second largest element in a list

Program:

```
def second_largest(arr):
    arr.sort()
    return arr[-2]

my_list = list(map(int, input().split()))
print("Second largest element:", second_largest(my_list))
```

Output:

3 4 67 99 34

Second largest element: 67

Process finished with exit code 0

5. Explain what indentation means in Python

Program:

Indentation in Python refers to the use of spaces or tabs at the beginning of lines of code to visually structure and organize the program. It helps indicate which lines of code belong together in the same block or scope, such as within loops, conditional statements, or function definitions. In Python, indentation is not just for visual clarity; it is a fundamental aspect of the language's syntax and is used by the interpreter to understand the logical structure of the code.

Example Code:

```
if temperature > 30:
    print("It's hot!") # This line is part of the if block
    print("Stay cool!") # This line is also part of the if block
else:
    print("It's not too hot.") # This line is part of the else block
    print("Enjoy your day!") # This line is also part of the else block
```

6. Write a program to perform set difference operation

Program:

```
set1 = set(map(int,input().split()))
set2 = set(map(int,input().split()))
difference_operator = set1 - set2
print("Set difference :", difference_operator)
```

Output:

1 2 3 4 5

4 5 6 7 8

Set difference : {1, 2, 3}

7. Write a Python program to print numbers from 1 to 10 using a while loop.

Program:

```
num = 1
while num <= 10:
    print(num, end=", ")
    num += 1
```

Output:

```
1,2,3,4,5,6,7,8,9,10,
Process finished with exit code 0
```

8. Write a program to calculate the factorial of a number using a while loop

Program:

```
def factorial(n):
    result = 1
    while n > 1:
        result *= n
        n -= 1
    return result

number = int(input("Enter a number: "))
print("Factorial of", number, "is", factorial(number))
```

Output:

```
Enter a number: 5
Factorial of 5 is 120
```

9. Write a Python program to check if a number is positive, negative, or zero using if-elif-else statements

Program:

```
def check_sign(num):
    if num > 0:
        print("The number is positive.")
    elif num < 0:
        print("The number is negative.")
    else:
        print("The number is zero.")

number = float(input("Enter a number: "))
check_sign(number)
```

Output:

```
Enter a number: 0
The number is zero.
```

10. Write a program to determine the largest among three numbers using conditional statements.

Program:

```
def find_largest(num1, num2, num3):
    if num1 >= num2 and num1 >= num3:
        largest = num1
    elif num2 >= num1 and num2 >= num3:
        largest = num2
    else:
        largest = num3
    return largest

num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
num3 = float(input("Enter the third number: "))
largest = find_largest(num1, num2, num3)
print("The largest number is:", largest)
```

Output:

```
Enter the first number: 56
Enter the second number: 34
Enter the third number: 35
The largest number is: 56.0
```

11. Write a Python program to create a numpy array filled with ones of given shape.

Program:

```
import numpy as np
def create_ones_array(shape):
    ones_array = np.ones(shape)
    return ones_array
shape = list(map(int,input("Enter shape").split()))
ones_array = create_ones_array(shape)
print("Array filled with ones of shape", shape, ":\n", ones_array)
```

Output:

```
Enter shape3 4
Array filled with ones of shape [3, 4] :
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]
```

12. Write a program to create a 2D numpy array initialized with random integers.

Program:

```

import numpy as np

def create_random_array(rows, cols, min_val, max_val):
    random_array = np.random.randint(min_val, max_val + 1, size=(rows, cols))
    return random_array

rows = 3
cols = 4
min_val = 1
max_val = 100

random_array = create_random_array(rows, cols, min_val, max_val)
print("2D NumPy array initialized with random integers:")
print(random_array)

```

Output:

```

2D NumPy array initialized with random integers:
[[10 56 46 12]
 [21 92 23  8]
 [62 19 69 39]]

```

13. Write a Python program to generate an array of evenly spaced numbers over a specified range using linspace.

Program:

```

1 import numpy as np
2 start = 0
3 end = 5
4 num_points = 10
5 evenly_spaced_array = np.linspace(start, end, num_points)
6 print("Evenly spaced array (including the right endpoint):")
7 print(evenly_spaced_array)
8 evenly_spaced_array_excluding_endpoint = np.linspace(start, end, num_points, endpoint=False)
9 print("\nEvenly spaced array (excluding the right endpoint):")
10 print(evenly_spaced_array_excluding_endpoint)
11
12

```

Run: even spaced

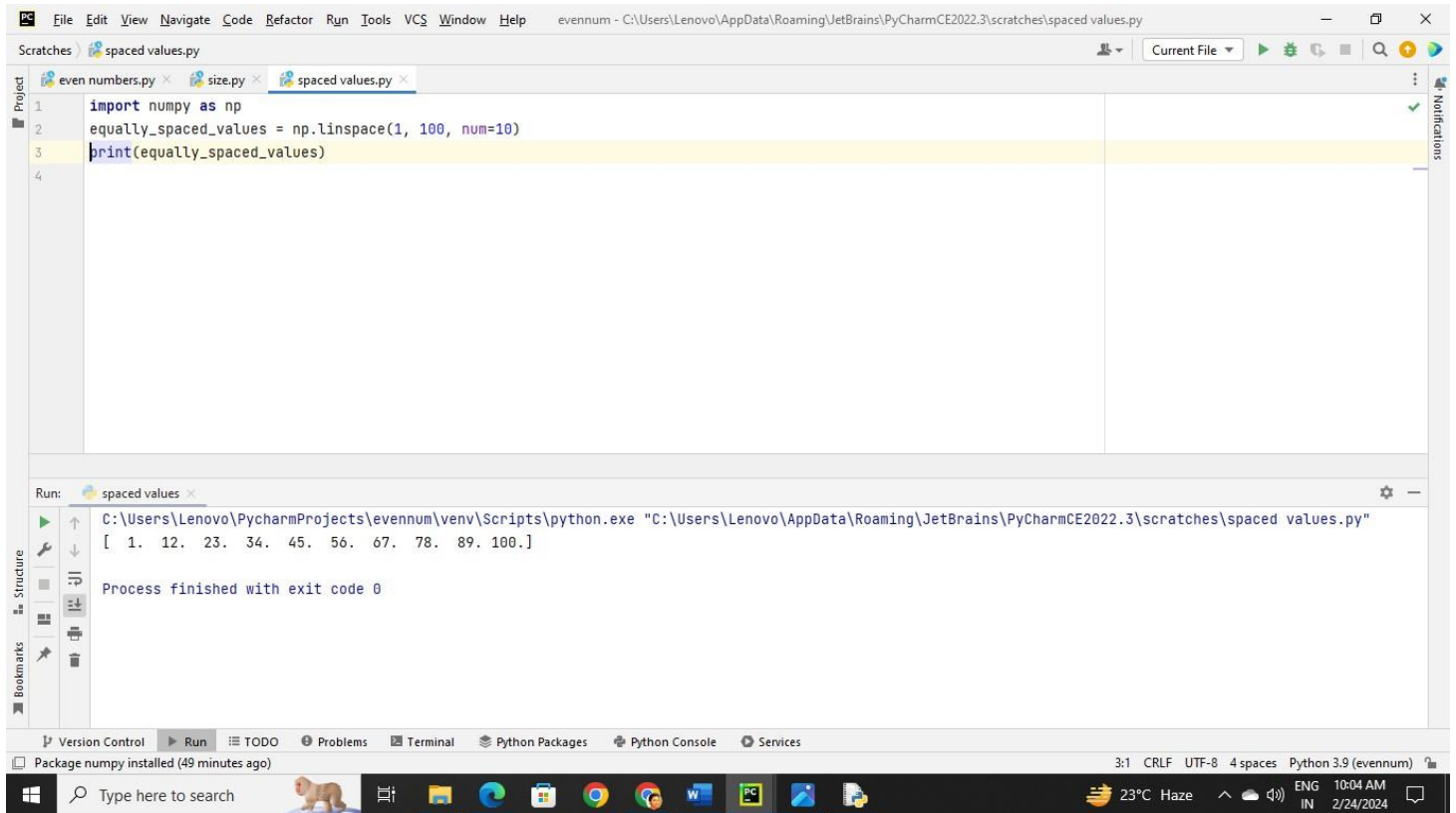
```

C:\Users\Lenovo\PycharmProjects\evennum\venv\Scripts\python.exe "C:\Users\Lenovo\AppData\Roaming\JetBrains\PyCharmCE2022.3\scratches\even spaced.py"
Evenly spaced array (including the right endpoint):
[0. 0.55555556 1.11111111 1.66666667 2.22222222 2.77777778
 3.33333333 3.88888889 4.44444444 5. ]
Evenly spaced array (excluding the right endpoint):
[0. 0.5 1. 1.5 2. 2.5 3. 3.5 4. 4.5]
Process finished with exit code 0

```

14. Write a program to generate an array of 10 equally spaced values between 1 and 100 using line space.

Program:



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python script in a file named 'spaced values.py'. The script consists of three lines of code: `import numpy as np`, `equally_spaced_values = np.linspace(1, 100, num=10)`, and `print(equally_spaced_values)`. The third line is highlighted in yellow. Below the editor, the 'Run' panel shows the execution output. The command executed is `C:\Users\Lenovo\PycharmProjects\evennum\venv\Scripts\python.exe "C:\Users\Lenovo\AppData\Roaming\JetBrains\PyCharmCE2022.3\scratches\spaced values.py"`, and the output is `[1. 12. 23. 34. 45. 56. 67. 78. 89. 100.]`. The status bar at the bottom indicates 'Process finished with exit code 0'. The bottom of the screen shows the Windows taskbar with various application icons and system information like '23°C Haze' and '10:04 AM 2/24/2024'.

```
1 import numpy as np
2 equally_spaced_values = np.linspace(1, 100, num=10)
3 print(equally_spaced_values)
4
```

Run: spaced values

C:\Users\Lenovo\PycharmProjects\evennum\venv\Scripts\python.exe "C:\Users\Lenovo\AppData\Roaming\JetBrains\PyCharmCE2022.3\scratches\spaced values.py"

[1. 12. 23. 34. 45. 56. 67. 78. 89. 100.]

Process finished with exit code 0

15. Write a Python program to create an array containing even numbers from 2 to 20 using arrange.

Program:

The screenshot shows the PyCharm IDE interface. The top toolbar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The main editor window displays a file named `even numbers.py` with the following code:

```
1 import numpy as np
2 even_numbers = np.arange(2, 21, 2)
3 print("Array of even numbers from 2 to 20:", even_numbers)
4
```

The Run window at the bottom shows the execution output:

```
C:\Users\Lenovo\PycharmProjects\evennum\venv\Scripts\python.exe "C:\Users\Lenovo\AppData\Roaming\JetBrains\PyCharmCE2022.3\scratches\even numbers.py"
Array of even numbers from 2 to 20: [ 2  4  6  8 10 12 14 16 18 20]

Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8 and the current file is `evennum`.

16. Write a program to create an array containing numbers from 1 to 10 with a step size of 0.5 using `arange`.

The screenshot shows the PyCharm IDE interface. The top toolbar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The main editor window displays a file named `size.py` with the following code:

```
1 import numpy as np
2 numbers = np.arange(1, 10.5, 0.5)
3 print(numbers)
4
```

The Run window at the bottom shows the execution output:

```
C:\Users\Lenovo\PycharmProjects\evennum\venv\Scripts\python.exe C:\Users\Lenovo\AppData\Roaming\JetBrains\PyCharmCE2022.3\scratches\size.py
[ 1.  1.5  2.  2.5  3.  3.5  4.  4.5  5.  5.5  6.  6.5  7.  7.5
 8.  8.5  9.  9.5 10.]

Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8 and the current file is `size.py`.