# PYTHON MASTERY – PRACTICE QUESTIONS AND VIVA

# VIVA QUESTIONS:

## Unit I - Setting up your Programming Environment and Variables, Expression and Statements:

What is a keyword in Python, and why is it important to avoid using keywords as variable names?

Answer: A keyword is a reserved word in Python that has a specific meaning and purpose. It is important to avoid using keywords as variable names because doing so will result in a syntax error.

What is the difference between a string and a variable in Python?

Answer: A variable is a name that refers to a value, whereas a string is a type of value that represents a sequence of characters. In Python, a variable can be assigned a string value, among other types of values.

What is the order of operations in Python, and how can it be modified?

Answer: The order of operations in Python is determined by the following hierarchy: parentheses, exponentiation, multiplication and division (left to right), addition and subtraction (left to right). The order can be modified by using parentheses to group operations.

What is a comment in Python, and how can it be used to improve code readability?

Answer: A comment is a piece of text that is ignored by the Python interpreter and is used to provide additional information or explanations for code. Comments can improve code readability by providing context, explanations, or documentation for the code.

How can errors related to variables be avoided in Python?

Answer: Errors related to variables can be avoided in Python by using meaningful variable names, avoiding naming variables after Python keywords, and ensuring that variables are initialized before they are used in the code.

What is the purpose of a module in Python, and how can it be imported into a program?

Answer: A module in Python is a file that contains Python definitions and statements. It can be imported into a program using the "import" statement, which allows the program to use the functions and variables defined in the module.

How can multiple variables be assigned values in a single line of Python code?

Answer: Multiple variables can be assigned values in a single line of Python code by separating the variables with commas and assigning values to them in the same order.

What are the different types of operators in Python, and how are they used in expressions?

Answer: The different types of operators in Python include arithmetic, comparison, logical, bitwise, and membership operators. They are used in expressions to perform operations on values or variables.

What is the purpose of a function in Python, and how can it be defined and called in a program?

Answer: A function in Python is a block of reusable code that performs a specific task. It can be defined using the "def" keyword and called in a program by using its name followed by parentheses.

What is the purpose of a string in Python, and how can it be manipulated using built-in methods?

Answer: A string in Python is a sequence of characters that can be used to represent text or data. It can be manipulated using built-in methods, such as the "len()" function to get the length of a string, or the "upper()" method to convert a string to uppercase.

## Unit II - Conditional and Iterative Statements:

What is the modulus operator in Python, and how is it used in conditional statements?

Answer: The modulus operator in Python is represented by the percent sign (%), and it returns the remainder of a division operation. It can be used in conditional statements to test whether a number is even or odd, or to perform other similar operations.

What are Boolean expressions, and how are they used in conditional statements?

Answer: Boolean expressions are expressions that evaluate to either True or False. They are commonly used in conditional statements to determine whether certain code should be executed based on a particular condition.

What is the difference between a while loop and a for loop in Python?

Answer: A while loop in Python will continue to execute as long as its condition remains True, whereas a for loop will iterate over a sequence of values and then terminate. While loops are often used when you don't know in advance how many times you need to repeat a particular operation, whereas for loops are typically used when you do.

What are nested conditionals in Python, and how are they used?

Answer: Nested conditionals are conditional statements that are nested within other conditional statements. They are often used to perform more complex tests that involve multiple conditions.

What is the difference between the and and or operators in Python?

Answer: The and operator in Python returns True if both of its operands are True, whereas the or operator returns True if either of its operands are True. These operators can be used in conditional statements to perform more complex tests.

What is the difference between a while loop and a do-while loop in Python?

Answer: Python does not have a do-while loop construct, unlike some other programming languages. A while loop in Python can be used to achieve the same effect as a do-while loop by using a break statement to terminate the loop under certain conditions.

How can you use random numbers in Python to create more interesting programs?

Answer: Python includes a random module that can be used to generate random numbers. These numbers can be used in various ways to create more interesting programs, such as generating random passwords, creating random game levels, or simulating random events.

What are Boolean operators in Python, and how do they work?

Answer: Boolean operators in Python include and, or, and not. These operators can be used to combine Boolean expressions in various ways to create more complex tests. For example, the and operator returns True only if both of its operands are True, whereas the or operator returns True if either of its operands are True.

What is encapsulation in object-oriented programming, and how is it used in Python?

Answer: Encapsulation is a concept in object-oriented programming that refers to the practice of hiding internal details of an object from the outside world. This can be achieved in Python by using private attributes and methods, which can only be accessed from within the object itself.

How can you use recursion in Python to solve complex problems?

Answer: Recursion is a technique in programming where a function calls itself. It can be used to solve various types of problems, such as calculating factorials or finding the nth Fibonacci number. However, recursion can be computationally expensive, so it should be used judiciously.

## Unit III - Functions and Recursion:

What is a function in Python?

A function in Python is a block of reusable code that performs a specific task. It can take input arguments, and it can return output values.

How do you define a function in Python?

To define a function in Python, you use the def keyword followed by the function name and a set of parentheses that may contain input arguments. The function code block is indented under the def statement.

What is recursion, and how is it used in Python?

Recursion is a programming technique where a function calls itself one or more times to solve a problem. In Python, recursion can be used to solve problems that have a recursive structure, such as computing the factorial of a number or generating the Fibonacci sequence.

What are function parameters, and how are they used in Python?

Function parameters are the input values that a function receives when it is called. In Python, function parameters can be specified with default values, and they can be of any data type.

What is a lambda function in Python, and how is it used?

A lambda function in Python is a small anonymous function that can take any number of arguments, but it can only have one expression. Lambda functions are often used as a shorthand way of defining small functions that are used only once.

What is type conversion and coercion in Python, and how are they used in functions?

Type conversion is the process of converting one data type to another, while type coercion is the automatic conversion of one data type to another. In Python, both type conversion and coercion are used in functions to ensure that input values have the correct data type.

What are math functions in Python, and how are they used?

Math functions in Python are built-in functions that perform mathematical operations. They include functions for basic arithmetic, trigonometry, logarithms, and more. Math functions are often used in scientific and mathematical applications.

How do you add a new function to a Python program?

To add a new function to a Python program, you define the function using the def keyword and then call the function from other parts of the program. The new function can also take input arguments and return output values.

What is parameter passing in Python, and how is it used?

Parameter passing in Python is the process of passing input arguments to a function when it is called. There are several ways to pass parameters in Python, including positional arguments, keyword arguments, and default arguments. Parameter passing is essential in Python programming because it enables functions to work with different input values.
What is variable scope in Python, and how does it affect functions? Variable scope in Python refers to the part of a program where a variable can be accessed. In Python, variables can have either global or local scope, which affects how they are accessed in functions. Global variables are accessible throughout the entire program, while local variables are accessible only within the function where they are defined.

## Unit IV - Strings, Lists, Tuples, and Dictionaries:

What is a string in Python and how is it different from other data types?

Answer: A string is a sequence of characters in Python. It is different from other data types in that it is immutable, meaning that once a string is created, it cannot be changed.

How can you find the length of a string in Python?

Answer: The length of a string can be found using the built-in function len(). For example, len("hello") would return 5.

What is a list in Python and how is it different from other data types?

Answer: A list is an ordered collection of items in Python. It is different from other data types in that it is mutable, meaning that items can be added, removed, or modified.

How can you add an element to a list in Python?

Answer: An element can be added to a list using the append() method. For example, my_list.append(5) would add the integer 5 to the end of the list.

What is a dictionary in Python and how is it different from other data types?

Answer: A dictionary is an unordered collection of key-value pairs in Python. It is different from other data types in that it uses keys to access values, rather than numerical indices.
Hard:

What is string slicing in Python and how is it used?

Answer: String slicing is the process of creating a substring from a larger string in Python. It is done using the syntax string[start:end:step], where start is the starting index, end is the ending index (exclusive), and step is the step size. For example, "hello"[1:4:2] would return "el".

How can you reverse a string in Python?

Answer: A string can be reversed using slicing and the step size -1. For example, "hello"[::-1] would return "olleh".

What is a tuple in Python and how is it different from a list?

Answer: A tuple is an ordered collection of items in Python, similar to a list. However, tuples are immutable, meaning that once a tuple is created, its contents cannot be changed.

How can you sort a list in Python?

Answer: A list can be sorted using the built-in sort() method. For example, my_list.sort() would sort the list in ascending order.

What is aliasing in Python and how can it be avoided when working with lists?

Answer: Aliasing is the phenomenon where two variables refer to the same object in memory. When working with lists, it can cause unexpected behavior if one variable is modified. To avoid aliasing, a copy of the list can be made using the copy() method or by using slicing to create a new list.

## Unit V - Classes and Objects:

What is a class in Python?

Answer: A class in Python is a blueprint for creating objects. It defines a set of attributes and methods that are common to all objects of that class.

What is the init method in Python classes?

Answer: The init method is a special method in Python classes that is used to initialize the attributes of an object when it is created.

How do you create an instance of a class in Python?

Answer: To create an instance of a class in Python, you simply call the class like a function, passing any required arguments.

What is encapsulation in Python classes?

Answer: Encapsulation is the practice of hiding the internal details of a class from the outside world, and providing a well-defined interface for interacting with it.

How do you access the attributes of an object in Python?

Answer: You can access the attributes of an object in Python using the dot notation, like obj.attribute_name.

What is inheritance in Python classes?

Answer: Inheritance is the mechanism by which a new class is created from an existing class, inheriting all of its attributes and methods.

What is method overriding in Python classes?

Answer: Method overriding is the practice of redefining a method in a subclass that was already defined in its superclass, to modify its behavior.

What is data hiding in Python classes?

Answer: Data hiding is the practice of making the attributes of a class private, so that they can only be accessed and modified by methods of the same class.

What is function overloading in Python classes?

Answer: Function overloading is the practice of defining multiple methods with the same name in a class, but with different parameters or arguments.

How do you create a subclass in Python?

Answer: To create a subclass in Python, you define a new class that inherits from an existing class, and then add any additional attributes or methods that are specific to the subclass.

## Unit VI - Files and Exceptions:

What is a text file in Python?

Answer: A text file in Python is a file that contains plain text, as opposed to binary data. Text files can be opened and read using Python's built-in file handling functions.

What is pickling in Python?

Answer: Pickling is the process of converting a Python object into a binary representation that can be stored in a file or transmitted over a network, and then restoring the object back to its original state later.

How do you write to a file in Python?

Answer: You can write to a file in Python using the 'w' mode when opening the file, and then using the write() method to write data to the file.

How do you read from a file in Python?

Answer: You can read from a file in Python using the 'r' mode when opening the file, and then using the read() method to read data from the file.

What is an exception in Python?

Answer: An exception in Python is an error that occurs during the execution of a program, such as a division by zero or an invalid file name.

How do you handle exceptions in Python?

Answer: You can handle exceptions in Python using the try-except statement, which allows you to catch and handle specific types of exceptions that might occur during program execution.

What is the else block in a try-except statement in Python?

Answer: The else block in a try-except statement in Python is a block of code that is executed if no exceptions are raised during the execution of the try block.

What is the finally block in a try-except statement in Python?

Answer: The finally block in a try-except statement in Python is a block of code that is executed regardless of whether an exception was raised or not during the execution of the try block.

What is the zero division error in Python?

Answer: The zero division error in Python is an exception that occurs when you try to divide a number by zero.

What is web scraping in Python?

Answer: Web scraping in Python is the process of extracting data from websites, by parsing the HTML code and extracting the relevant information using regular expressions or other techniques.

# PRACTICE QUESTIONS

## UNIT 1 :

Easy:

1. Write a Python program to print "Hello World" on the console.
2. Declare a variable 'x' and assign the value 5 to it. Print the value of 'x' on the console.
3. Declare two variables 'a' and 'b' and assign values 10 and 20 respectively. Add 'a' and 'b' and print the result on the console.
4. Declare a string variable 'name' and assign your name to it. Print a message on the console that says "My name is [name]".
5. Declare a boolean variable 'is_raining' and assign the value True to it. Print a message on the console that says "It is raining: [is_raining]".

Medium:

1. Declare a variable 'x' and assign the value '5' to it as a string. Convert the string to an integer and print the value of 'x' on the console.

   Declare two variables 'a' and 'b' and assign values 10 and 20 respectively. Subtract 'b' from 'a' and print the result on the console.
2. Declare a string variable 'sentence' that says "The quick brown fox jumps over the lazy dog". Extract the word "quick" from the sentence and print it on the console.
3. Declare a variable 'age' and assign your age to it. Write a conditional statement to check if 'age' is greater than or equal to 18. If true, print "You are an adult", else print "You are a minor".
4. Declare a variable 'number' and assign a number to it. Write a conditional statement to check if 'number' is even or odd. Print "even" if it is even, and "odd" if it is odd.

Hard:

1. Declare a variable 'x' and assign the value 10. Write a WHILE loop that subtracts 1 from 'x' in each iteration until 'x' is equal to 0. Print the value of 'x' on each iteration.

   Declare a variable 'x' and assign the value 1. Write a FOR loop that multiplies 'x' by 2 in each iteration until 'x' is greater than or equal to 64. Print the value of 'x' on each iteration.

2. Declare a string variable 'sentence' that says "The quick brown fox jumps over the lazy dog". Replace the word "quick" with "slow" and print the updated sentence on the console.
3. Declare a list variable 'numbers' that contains the values [10, 20, 30, 40, 50]. Write a FOR loop that calculates the sum of the numbers in the list and print the result on the console.
4. Declare a variable 'sentence' and assign a sentence to it. Write a function that takes a string as input and returns the number of vowels in the string. Call the function with 'sentence' as input and print the result on the console.

Story-based:

1. You are creating a Python program to calculate the area of a rectangle. Write a program that asks the user for the length and width of the rectangle using input(), calculates the area using a formula, and prints the result on the console.
2. You are creating a Python program to calculate the price of a meal at a restaurant. Write a program that asks the user for the price of the meal, the tax rate, and the tip rate using input(), calculates the total price including tax and tip, and prints the result on the console.

## UNIT 2 :

Easy:

1. Write a Python program to check if a number is even or odd using a conditional statement.
2. Write a Python program to generate a random number between 1 and 10 using the random module.
3. Write a Python program to check if a string is empty using a boolean expression.
4. Write a Python program to check if a number is positive using a conditional statement.
5. Write a Python program to check if a number is divisible by 3 using the modulus operator.
6. Write a Python program to check if a number is between 1 and 100 using a conditional statement.

7. Write a Python program to check if a number is a multiple of 5 using the modulus operator.
8. Write a Python program to check if a string is a palindrome using a conditional statement.
9. Write a Python program to check if a number is prime using a nested conditional statement.
10. Write a Python program to print the first 10 even numbers using a FOR loop.

Medium:

1. Write a Python program to calculate the factorial of a number using a WHILE loop.
2. Write a Python program to find the largest number in a list using a FOR loop.
3. Write a Python program to find the sum of all even numbers between 1 and 50 using a WHILE loop.
4. Write a Python program to generate a list of random numbers between 1 and 100 using the random module and a FOR loop.
5. Write a Python program to find the second largest number in a list using a FOR loop.
6. Write a Python program to reverse a string using a WHILE loop.
7. Write a Python program to calculate the area of a circle using a conditional statement and a FOR loop.
8. Write a Python program to find the greatest common divisor (GCD) of two numbers using a WHILE loop.
9. Write a Python program to check if a number is a perfect square using a FOR loop and a conditional statement.
10. Write a Python program to print the Fibonacci sequence up to a given number using a WHILE loop.

Hard:

1. Write a Python program to sort a list of numbers in ascending order using a FOR loop and a nested conditional statement.

2. Write a Python program to implement the binary search algorithm using a WHILE loop and a conditional statement.
3. Write a Python program to find the prime factors of a number using a FOR loop and a nested conditional statement.
4. Write a Python program to implement the bubble sort algorithm using a FOR loop and a nested conditional statement.
5. Write a Python program to implement the merge sort algorithm using a WHILE loop and a nested conditional statement.
6. Write a Python program to implement the selection sort algorithm using a FOR loop and a nested conditional statement.
7. Write a Python program to find the least common multiple (LCM) of two numbers using a WHILE loop and a conditional statement.
8. Write a Python program to implement the insertion sort algorithm using a FOR loop and a nested conditional statement.
9. Write a Python program to implement the quick sort algorithm using a WHILE loop and a nested conditional statement.
10. Write a Python program to calculate the distance between two points using the Pythagorean theorem and a FOR loop.

Story-based:

1. You are creating a Python program to calculate the grade of a student based on their score. Write a program that uses a conditional statement to assign a grade of A, B, C, D, or F based on the score entered by the user.
2. You are creating a Python program to simulate a game of rock-paper-scissors. Write a program that uses a nested conditional statement to determine the winner of the game based on the choices made by the player and the computer.
3. You are creating a Python program to generate a quiz for students. Write a program that uses a FOR loop to display a series of questions and a conditional statement to determine if the answer entered by the student is correct or incorrect.

   You are creating a Python program to simulate a simple banking system. Write a program that uses a WHILE loop and a nested conditional statement to allow the user to deposit or withdraw money from their account and display their balance.

4. You are creating a Python program to generate a random password for users. Write a program that uses the random module and a FOR loop to generate a password with a specific length and a conditional statement to ensure that the password meets certain criteria (e.g. contains both letters and numbers).
5. You are creating a Python program to calculate the total cost of a trip for a group of people. Write a program that uses a FOR loop to ask for the number of people and their expenses and a conditional statement to calculate the total cost based on the number of people and the expenses entered.
6. You are creating a Python program to simulate a game of blackjack. Write a program that uses a WHILE loop and a nested conditional statement to allow the player to play the game and determine if they win or lose based on the cards they are dealt and the rules of the game.
7. You are creating a Python program to generate a report on the performance of a company. Write a program that uses a FOR loop to display the sales data for each department and a conditional statement to determine the highest and lowest performing departments.
8. You are creating a Python program to generate a recipe for users based on their preferences. Write a program that uses a WHILE loop to ask for the user's dietary restrictions and a nested conditional statement to generate a recipe that meets their preferences.
9. You are creating a Python program to simulate a game of tic-tac-toe. Write a program that uses a FOR loop and a nested conditional statement to allow the player to play the game and determine if they win or lose based on the positions of their markers and the rules of the game.

## UNIT 3 :

Easy:

1. Write a function that takes two numbers as input and returns their sum.
2. Write a function that takes a string as input and prints it on the console.
3. Write a function that takes a list as input and returns its length.
4. Write a function that takes a number as input and returns its square.

5. Write a function that takes two strings as input and concatenates them.
6. Write a function that takes a list of numbers as input and returns the sum of all the numbers.
7. Write a function that takes a number as input and returns True if it's even and False if it's odd.
8. Write a function that takes a string as input and returns its length.
9. Write a function that takes a list of strings as input and returns a new list with all the strings capitalized.
10. Write a function that takes a number as input and returns True if it's positive, False if it's negative, and "zero" if it's zero.

Medium:

1. Write a function that takes a list of numbers as input and returns the average.

   Write a function that takes a string as input and returns a new string with all the vowels removed.
2. Write a function that takes two lists as input and returns a new list with the elements that are common to both lists.
3. Write a function that takes a list of numbers as input and returns the largest number.
4. Write a function that takes a string as input and returns True if it's a palindrome and False if it's not.
5. Write a function that takes a list of numbers as input and returns a new list with all the even numbers.
6. Write a function that takes a list of strings as input and returns a new list with all the strings reversed.
7. Write a function that takes a list of numbers as input and returns a new list with all the numbers sorted in descending order.

8. Write a function that takes a string as input and returns a new string with all the words capitalized.
9. Write a function that takes a number as input and returns the factorial of that number.

Hard:

1. Write a function that takes a list of strings as input and returns the longest string.

   Write a function that takes a list of numbers as input and returns a new list with all the prime numbers.
2. Write a function that takes a list of strings as input and returns a new list with all the strings sorted in alphabetical order.
3. Write a function that takes a string as input and returns a new string with all the characters shifted by a specified number of positions.
4. Write a function that takes a list of numbers as input and returns the product of all the numbers.
5. Write a function that takes a string as input and returns the number of times a specified character appears in the string.
6. Write a function that takes a list of strings as input and returns a new list with all the duplicates removed.
7. Write a function that takes a list of numbers as input and returns a new list with all the numbers converted to binary.
8. Write a function that takes a string as input and returns a new string with all the non-alphanumeric characters removed.
9. Write a function that takes a number as input and returns the nth Fibonacci number.

Story-based:

1. You are building a program to calculate the cost of a pizza order based on the number of toppings and the size of the pizza. Write a function that takes the size and number

of toppings as inputs and returns the total cost. Call the function with size=12 and toppings=3 and print the result on the console.

2. You are developing a program to calculate the shipping cost of an order based on the weight and distance of the package. Write a function that takes the weight and distance as inputs and returns the total shipping cost. Call the function with weight=10 and distance=200 and print the result on the console.
3. You are creating a program to calculate the area and perimeter of a rectangle. Write a function that takes the length and width as inputs and returns the area and perimeter. Call the function with length=5 and width=10 and print the results on the console.

   You are building a program to calculate the discount for a sale based on the total amount spent. Write a function that takes the total amount spent as input and returns the discount percentage. Call the function with amount=100 and print the result on the console.

4. You are creating a program to calculate the average grade of a class. Write a function that takes a list of grades as input and returns the average. Call the function with grades=[85, 90, 75, 92, 88] and print the result on the console.

5. You are developing a program to convert temperatures from Celsius to Fahrenheit. Write a function that takes the temperature in Celsius as input and returns the equivalent temperature in Fahrenheit. Call the function with temperature=25 and print the result on the console.
6. You are building a program to check if a word is a palindrome. Write a function that takes a word as input and returns True if it's a palindrome and False if it's not. Call the function with word="racecar" and print the result on the console.
7. You are creating a program to calculate the area and circumference of a circle. Write a function that takes the radius as input and returns the area and circumference. Call the function with radius=5 and print the results on the console.
8. You are developing a program to find the largest common divisor of two numbers. Write a function that takes two numbers as input and returns their largest common divisor. Call the function with numbers 12 and 18 and print the result on the console.
9. You are building a program to generate a random password. Write a function that takes the length of the password as input and returns a random password of that

length. Call the function with length=8 and print the result on the console.

## UNIT 4 :

**Strings:**

Easy:

1. Create a string variable containing your name and print it.
2. What is the length of the string "Hello, World!"?
3. Print the character 'l' from the string "Hello, World!".
4. Concatenate the two strings "Hello" and "World" and print the result.
5. Convert the string "42" to an integer.

Medium:

1. Replace all occurrences of the character 'a' with 'x' in the string "banana".
2. Print the first three characters of the string "Hello, World!".
3. Print the last three characters of the string "Hello, World!".
4. Remove all leading and trailing whitespace characters from the string " Hello, World! ". Convert the string "3.14" to a float.

Hard:

1. Reverse the string "Hello, World!".
2. Check if the string "racecar" is a palindrome.
3. Replace all occurrences of the substring "oo" with "ee" in the string "good food".
4. Count the number of occurrences of the character 'l' in the string "Hello, World!".

5. Capitalize the first letter of each word in the string "hello, world!".

**Lists:**

Easy:

1. Create a list containing the numbers 1, 2, 3 and print it.
2. What is the length of the list ['apple', 'banana', 'cherry']?
3. Print the element at index 1 in the list ['apple', 'banana', 'cherry'].
4. Append the element 'orange' to the list ['apple', 'banana', 'cherry'] and print the result.
5. Remove the first occurrence of the element 'banana' from the list ['apple', 'banana', 'cherry'].

Medium:

1. Create a list containing the first five even numbers (starting from 2) and print it.
2. Sort the list ['orange', 'apple', 'banana'] in alphabetical order and print the result.
3. Reverse the list ['apple', 'banana', 'cherry'] and print the result.
4. Insert the element 'grape' at index 1 in the list ['apple', 'banana', 'cherry'] and print the result.
5. Remove all occurrences of the element 'banana' from the list ['apple', 'banana', 'cherry', 'banana'].

Hard:

1. Flatten the list [[1, 2], [3, 4], [5, 6]] into a single list and print the result.
2. Check if the list ['apple', 'banana', 'cherry'] contains the element 'cherry'.
3. Create a list of all the words in the sentence "The quick brown fox jumps over the lazy dog" and print it.
4. Count the number of occurrences of the element 'apple' in the list ['apple', 'banana', 'cherry', 'apple'].
5. Create a list containing the first 20 Fibonacci numbers and print it.

**Tuples:**

Easy:

1. Create a tuple containing the numbers 1, 2, 3 and print it.
2. What is the length of the tuple ('apple', 'banana', 'cherry')?
3. Print the element at index 1 in the tuple ('apple', 'banana', 'cherry').
4. Concatenate the two tuples ('apple', 'banana') and ('cherry',) and print the result.
5. Convert the tuple (1, 2, 3) to a list.

Medium:

1. Create a tuple containing some integers, and write a program to iterate through the tuple and print each element multiplied by 2.

   Write a program that takes a string as input and prints out the first and last characters of the string.
2. Write a program that takes a string as input and prints out every other character of the string.
3. Write a program that takes a string as input and replaces all occurrences of the letter 'a' with the letter 'e'.
4. Write a program that takes a string as input and checks if the string contains the word "Python".
5. Write a program that takes a string as input and prints out the number of vowels in the string.
6. Write a program that takes two strings as input and checks if they are anagrams of each other.
7. Write a program that takes a string as input and returns a new string that contains only the consonants of the original string.
8. Write a program that takes a string as input and returns the string in reverse order.
   Write a program that takes a string as input and checks if it is a valid palindrome even if the string contains non-alphabetic characters.

9. Write a program that takes a string as input and removes all duplicate characters from the string.
10. Write a program that takes a string as input and returns the longest substring that contains only one unique character.

Hard:

1. Write a program that takes a string as input and checks if it can be split into two equal parts that are reverse of each other.
2. Write a program that takes a string as input and returns the smallest substring that contains all the characters of the input string.
3. Write a program that takes a string as input and finds the longest common substring between two given strings.
4. Write a program that takes a string as input and returns the shortest palindrome that can be formed by adding characters to the end of the string.
5. Write a program that takes a string as input and returns the longest palindrome that can be formed by removing characters from the string.

**Dictionaries:**

Easy:

1. Write a Python program to create an empty dictionary and add some key-value pairs to it.
2. Write a Python program to create a dictionary with the key-value pairs "name": "John", "age": 25, and "city": "New York". [Que-2]
3. Write a Python program to access t]he value of the key "age" in the dictionary from question 2.
4. Write a Python program to add the key-value pair "gender": "male" to the dictionary from question 2.
5. Write a Python program to update the value of the key "age" to 30 in the dictionary from question 2.

Medium:

1. Write a Python program to remove the key "city" from the dictionary from question 2.

2. Write a Python program to check if the key "name" exists in the dictionary from question 2.
3. Write a Python program to print all the keys in the dictionary from question 2.
4. Write a Python program to print all the values in the dictionary from question 2.
5. Write a Python program to print all the key-value pairs in the dictionary from question 2.
6. Write a Python program to iterate over the keys in the dictionary from question 2 and print their corresponding values.
7. Write a Python program to iterate over the key-value pairs in the dictionary from question 2 and print them.

Hard:

1. Write a Python program to create a dictionary where the keys are numbers from 1 to 5 and the values are their squares.
2. Write a Python program to create a dictionary where the keys are the letters from "a" to "e" and the values are their ASCII codes.
3. Write a Python program to create a dictionary where the keys are the names of fruits and the values are their prices.(Ques.15)
   Write a Python program to update the price of "apple" to 2.5 in the dictionary from question 15.
4. Write a Python program to calculate the total price of all the fruits in the dictionary from question 15.
5. Write a Python program to find the cheapest fruit in the dictionary from question 15.
6. Write a Python program to find the most expensive fruit in the dictionary from question 15.
7. Write a Python program to create a dictionary where the keys are the names of programming languages and the values are their creators.

## UNIT 5:

Easy:

1. Create a class named "Person" with the attribute "name".
2. Create an object of the "Person" class and assign the name "Alice" to it.
3. Create a class named "Dog" with the attributes "name" and "age".
4. Create an object of the "Dog" class and assign the name "Buddy" and age "3" to it.
5. Create a class named "Rectangle" with the attributes "width" and "height".
6. Create an object of the "Rectangle" class and assign the values "5" and "10" to its attributes.
7. Create a method named "get_name" in the "Person" class that returns the person's name.
8. Create a method named "get_age" in the "Dog" class that returns the dog's age.
9. Create a method named "area" in the "Rectangle" class that calculates and returns the rectangle's area.
10. Create a method named "perimeter" in the "Rectangle" class that calculates and returns the rectangle's perimeter.

Medium:

1. Create a class named "Circle" with the attributes "radius" and "color".
2. Create an object of the "Circle" class and assign the values "5" and "red" to its attributes.
3. Create a method named "diameter" in the "Circle" class that calculates and returns the circle's diameter.
4. Create a method named "area" in the "Circle" class that calculates and returns the circle's area.
5. Create a class named "Employee" with the attributes "name", "salary", and "employment_year".
   Create an object of the "Employee" class and assign the values "John", "50000", and "2010" to its attributes.

6. Create a method named "get_years_of_service" in the "Employee" class that calculates and returns the employee's years of service.
7. Create a method named "raise_salary" in the "Employee" class that increases the employee's salary by 10%.
8. Create a class named "Book" with the attributes "title", "author", and "pages".
9. Create an object of the "Book" class and assign the values "To Kill a Mockingbird", "Harper Lee", and "281" to its attributes.

Hard:

1. Create a class named "BankAccount" with the attributes "account_number", "account_holder_name", and "balance".
2. Create a method named "deposit" in the "BankAccount" class that adds a given amount to the account's balance.
3. Create a method named "withdraw" in the "BankAccount" class that subtracts a given amount from the account's balance.
4. Create a method named "transfer" in the "BankAccount" class that transfers a given amount from the account to another given account.
5. Create a class named "Point" with the attributes "x" and "y".
6. Create a method named "distance" in the "Point" class that calculates and returns the distance between two given points.
7. Create a method named "translate" in the "Point" class that moves a given point by a given amount in the x and y directions.
8. Create a class named "Vehicle" with the attributes "make", "model", and "year".
9. Create a method named "start_engine" in the "Vehicle" class that starts the vehicle's engine.
10. Create a method named "stop_engine" in the "Vehicle" class that stops the vehicle's engine.

Story-based:

1. You are building a game that requires a Player class. Write a Player class with attributes name, health, and level.
   In your game, a Wizard is a special type of Player with additional attributes mana and spellbook. Create a Wizard subclass that inherits from the Player class.
2. Your game also has a Monster class with attributes name, health, and attack_power. Create a Monster class.
   In your game, you want the Wizard to be able to cast spells. Add a cast_spell method to the Wizard class that takes a spell name as an argument and deducts mana from the wizard's mana attribute.
   In your game, you want the Monster to be able to attack the Player. Add an attack method to the Monster class that takes a Player object as an argument and reduces the Player object's health attribute by the Monster object's attack_power attribute.
3. You are building a banking system and need to store customer account information. Create a Customer class with attributes name, address, account_number, and balance.
   In your banking system, you want to allow customers to deposit and withdraw funds from their account. Add deposit and withdraw methods to the Customer class that add or subtract from the balance attribute.
4. Your banking system also has a Bank class that stores a list of Customer objects. Add a add_customer method to the Bank class that takes a Customer object as an argument and adds it to the list.
5. In your banking system, you want to be able to search for a Customer by account number. Add a find_customer method to the Bank class that takes an account number as an argument and returns the corresponding Customer object.
6. You are building a shopping cart for an online store. Create a Cart class with attributes items and total_price.
   In your shopping cart, you want to be able to add items to the cart. Add an add_item method to the Cart class that takes an item object as an argument and adds it to the items attribute.
   In your shopping cart, you want to be able to calculate the total price of all items in the cart. Add a calculate_total_price method to the Cart class that returns the total price of all items in the items attribute.
7. Your online store also has a Product class with attributes name and price. Create a Product class.
   In your online store, you want to be able to create new products and add them to your store's inventory. Add a create_product method to the Product class that takes a

name and price as arguments and returns a new Product object.
In your online store, you want to be able to add and remove products from your store's inventory. Add add_product and remove_product methods to the Cart class that add or remove a Product object from the store's inventory.

8. You are building a school administration system and need to store student information. Create a Student class with attributes name, grade, and courses.

## UNIT 6

Easy :

1. Write Python code to open a file named "example.txt" in read mode.
2. Write Python code to read the first line of a file named "example.txt".
3. Write Python code to write "Hello, World!" to a file named "example.txt".
4. Write Python code to create a directory named "mydir".
5. Write Python code to handle the ZeroDivisionError exception.

Medium:

1. Write Python code to append the text "Goodbye, World!" to a file named "example.txt".
2. Write Python code to read all the lines of a file named "example.txt" and print each line.
3. Write Python code to create a file named "example.txt" if it doesn't exist.
4. Write Python code to handle the FileNotFoundError exception.
5. Write Python code to delete a file named "example.txt".

Hard:

1. Write Python code to write a list of numbers to a file named "numbers.txt".
2. Write Python code to read the contents of a binary file named "example.bin".
3. Write Python code to handle the KeyboardInterrupt exception.

4. Write Python code to handle the TypeError exception.
5. Write Python code to recursively delete a directory named "mydir".

Regular Expressions: Easy:

1. Write Python code to import the re module.
2. Write Python code to search for the word "Python" in a string.
3. Write Python code to find all the occurrences of the word "Python" in a string.
4. Write Python code to replace all the occurrences of the word "Python" with "Java" in a string.
5. Write Python code to check if a string starts with the word "Hello".

Medium:

1. Write Python code to extract all the email addresses from a string using regular expressions.
2. Write Python code to extract all the phone numbers from a string using regular expressions.
3. Write Python code to extract all the dates from a string using regular expressions.
4. Write Python code to extract all the URLs from a string using regular expressions.
5. Write Python code to extract all the IP addresses from a string using regular expressions.

Hard:

1. Write Python code to extract all the hashtags from a string using regular expressions.
2. Write Python code to extract all the mentions (usernames) from a string using regular expressions.
3. Write Python code to extract all the numbers (integers and decimals) from a string using regular expressions.
4. Write Python code to extract all the uppercase words from a string using regular expressions.

5. Write Python code to extract all the lowercase words from a string using regular expressions.

Storybased:

1. You're working on a project that requires you to read a file containing data. However, the file is in a format that is not readable by Python. How would you handle this situation using Python code?
2. You're writing a program that reads data from a file and performs some calculations. How would you handle the situation where the file is empty, and there is no data to read? Write Python code to handle this scenario.
3. You want to create a backup of your Python program by saving it to a file. Write Python code to create a new file and write the contents of your program to it.
4. You're working with a large file that contains data in a specific format. You want to extract only specific lines from the file and save them to a new file. How would you accomplish this using Python code?
5. You want to store a dictionary object in a file using Python's pickling feature. Write Python code to create a new file, save the dictionary to it, and then load it back into memory.
6. You're working with a file that contains data in a specific format, and you want to extract information from it using regular expressions. Write Python code to extract the information and save it to a new file.
7. You're working on a web scraping project and need to extract information from a website using regular expressions. Write Python code to extract the information and save it to a file.
8. You're working on a project that requires you to read multiple files from a directory and combine the data into a single file. How would you accomplish this using Python code?
9. You're working on a project that requires you to read a file containing data. However, the file is in a format that is not readable by Python. How would you handle this situation using Python code?

10. You're writing a program that reads data from a file and performs some calculations. How would you handle the situation where the file is empty, and there is no data to read? Write Python code to handle this scenario.

11. You want to create a backup of your Python program by saving it to a file. Write Python code to create a new file and write the contents of your program to it. You're working with a large file that contains data in a specific format.

12. You want to extract only specific lines from the file and save them to a new file. How would you accomplish this using Python code?

13. You want to store a dictionary object in a file using Python's pickling feature. Write Python code to create a new file, save the dictionary to it, and then load it back into memory.

14. You're working with a file that contains data in a specific format, and you want to extract information from it using regular expressions. Write Python code to extract the information and save it to a new file.

15. You're working on a web scraping project and need to extract information from a website using regular expressions. Write Python code to extract the information and save it to a file.

16. You're working on a project that requires you to read multiple files from a directory and combine the data into a single file. How would you accomplish this using Python code?

17. You're writing a program that performs some calculations on a user input. How would you handle the scenario where the user inputs a string instead of a numerical value, and you need to prompt them to enter a valid input? Write Python code to handle this scenario using try-except blocks. You're working on a project that requires you to create a directory if it does not already exist. Write Python code to check if the directory exists, and if not, create it. You're working on a project that requires you to handle the scenario where a file is not found. Write Python code to handle this scenario using try-except blocks and prompt the user to enter a valid file name.