**MCA 3rd Sem**

**BMC301Python Programming**

**BMC351 Python Programming Lab  
Assignments with CO Mapping**

# Unit I – Introduction to Python, Variables, Operators & Control Structures

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Assignment** | **Type** | **CO** |
| 1 | Explain the importance of Python as a programming language compared to other languages. | Theory | CO1 |
| 2 | Discuss Python coding standards and naming conventions with examples. | Theory | CO1 |
| 3 | Differentiate between mutable and immutable data variables in Python with examples. | Theory | CO1 |
| 4 | Write short notes on id() and type() functions with suitable examples. | Theory | CO1 |
| 5 | Explain the working of if, elif, and nested if with examples. | Theory | CO1 |
| 6 | Discuss the role of iteration control structures in Python. | Theory | CO1 |
| 7 | Explain the differences between break, continue, and pass with examples. | Theory | CO1 |
| 8 | Describe the process of setting up a Python environment and path variables. | Theory | CO1 |
| 9 | Write short notes on Python operators with suitable examples. | Theory | CO1 |
| 10 | Explain how to take user input in Python using input() function with examples. | Theory | CO1 |
| 11 | Write a Python program to demonstrate use of id() and type() functions. | Practical | CO1 |
| 12 | Write a program to check whether a number entered by the user is even or odd using if-else. | Practical | CO1 |
| 13 | Write a program to input marks of a student and display grade using nested if. | Practical | CO1 |
| 14 | Write a program to demonstrate the use of break in a loop. | Practical | CO1 |
| 15 | Write a program to demonstrate the use of continue in a loop. | Practical | CO1 |
| 16 | Write a program to demonstrate the use of pass in a loop. | Practical | CO1 |
| 17 | Write a program to accept three numbers and display the largest using if-elif-else. | Practical | CO1 |
| 18 | Write a program to print the sum of digits of a given number using iteration. | Practical | CO1 |
| 19 | Write a program to swap two variables without using a third variable. | Practical | CO1 |
| 20 | Write a program to read user details (name, age, city) and display them in a formatted way. | Practical | CO1 |

# Unit II – Strings, Lists & Tuples

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Assignment** | **Type** | **CO** |
| 1 | Explain string slicing and indexing in Python with examples. | Theory | CO2 |
| 2 | Discuss various string functions and methods with examples. | Theory | CO2 |
| 3 | Compare lists and tuples with respect to mutability, performance, and use cases. | Theory | CO3 |
| 4 | Explain list operations like concatenation, repetition, and membership. | Theory | CO2 |
| 5 | Discuss common tuple methods and their uses. | Theory | CO3 |
| 6 | Differentiate between shallow copy and deep copy in lists with examples. | Theory | CO3 |
| 7 | Explain how negative indexing works in lists and strings with examples. | Theory | CO3 |
| 8 | Describe the role of built-in functions (len, min, max, sum) for lists and tuples. | Theory | CO3 |
| 9 | Write notes on nested lists and list comprehensions with examples. | Theory | CO3 |
| 10 | Explain the immutability of strings with a program. | Theory | CO2 |
| 11 | Write a program to count vowels and consonants in a given string. | Practical | CO2 |
| 12 | Write a program to check whether a string is a palindrome. | Practical | CO2 |
| 13 | Write a program to demonstrate string slicing and concatenation. | Practical | CO2 |
| 14 | Write a program to find the largest element in a list. | Practical | CO3 |
| 15 | Write a program to reverse a list without using built-in functions. | Practical | CO3 |
| 16 | Write a program to demonstrate tuple packing and unpacking. | Practical | CO3 |
| 17 | Write a program to create a nested list and access its elements. | Practical | CO3 |
| 18 | Write a program to merge two lists into a dictionary. | Practical | CO3 |
| 19 | Write a program to demonstrate use of list methods like append, insert, remove. | Practical | CO3 |
| 20 | Write a program to find the frequency of each character in a string. | Practical | CO2 |

# Unit III – Dictionaries & Functions

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Assignment** | **Type** | **CO** |
| 1 | Define dictionaries and explain their key properties. | Theory | CO3 |
| 2 | Compare lists and dictionaries with examples. | Theory | CO3 |
| 3 | Explain the concept of hashing in context of dictionaries. | Theory | CO3 |
| 4 | Explain different types of function arguments in Python with examples. | Theory | CO2 |
| 5 | Explain recursion with an example of factorial. | Theory | CO2 |
| 6 | Discuss local vs global variables with examples. | Theory | CO2 |
| 7 | Explain mutable and immutable arguments in Python with examples. | Theory | CO2 |
| 8 | Write short notes on dictionary functions like keys(), values(), items(). | Theory | CO3 |
| 9 | Explain the scope of variables (LEGB rule) in Python. | Theory | CO2 |
| 10 | Discuss the advantages of using functions in Python programming. | Theory | CO2 |
| 11 | Write a program to demonstrate creation and manipulation of a dictionary. | Practical | CO3 |
| 12 | Write a program to count word frequency in a paragraph using a dictionary. | Practical | CO3 |
| 13 | Write a program to merge two dictionaries. | Practical | CO3 |
| 14 | Write a function to calculate factorial of a number using recursion. | Practical | CO2 |
| 15 | Write a program to demonstrate default and keyword arguments in functions. | Practical | CO2 |
| 16 | Write a program to illustrate local and global variables. | Practical | CO2 |
| 17 | Write a program to check whether a key exists in a dictionary. | Practical | CO3 |
| 18 | Write a function to find Fibonacci numbers using recursion. | Practical | CO2 |
| 19 | Write a program to demonstrate mutable vs immutable arguments. | Practical | CO2 |
| 20 | Write a program to copy contents of one dictionary into another. | Practical | CO3 |

# Unit IV – Modules, Packages, Regex & File Handling

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Assignment** | **Type** | **CO** |
| 1 | Explain the difference between a module and a package. | Theory | CO4 |
| 2 | Discuss the use of standard libraries in Python with examples. | Theory | CO4 |
| 3 | Explain the importance of sys and math modules with examples. | Theory | CO4 |
| 4 | Discuss the use of Regular Expressions in string processing. | Theory | CO2 |
| 5 | Explain file modes in Python with examples. | Theory | CO4 |
| 6 | Explain the concept of binary files and their applications. | Theory | CO4 |
| 7 | Discuss file pointer methods (seek, tell) with examples. | Theory | CO4 |
| 8 | Write short notes on Python’s datetime module with examples. | Theory | CO4 |
| 9 | Explain the process of creating user-defined modules with an example. | Theory | CO4 |
| 10 | Discuss the use of numpy and scipy in scientific computing. | Theory | CO4 |
| 11 | Write a program to import a user-defined module and use its function. | Practical | CO4 |
| 12 | Write a program to generate random numbers using random module. | Practical | CO4 |
| 13 | Write a program to demonstrate mathematical operations using math module. | Practical | CO4 |
| 14 | Write a regex program to validate an email ID. | Practical | CO2 |
| 15 | Write a regex program to search and replace a substring in a text. | Practical | CO2 |
| 16 | Write a program to read contents of a text file and count number of lines. | Practical | CO4 |
| 17 | Write a program to copy contents of one file into another. | Practical | CO4 |
| 18 | Write a program to demonstrate use of binary file operations. | Practical | CO4 |
| 19 | Write a program to display current date and time using datetime module. | Practical | CO4 |
| 20 | Write a program to demonstrate seek and tell file pointer operations. | Practical | CO4 |

# Unit V – Exception Handling & Data Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Assignment** | **Type** | **CO** |
| 1 | Define exceptions and explain their importance in programming. | Theory | CO4 |
| 2 | Explain the working of try-except block with examples. | Theory | CO4 |
| 3 | Discuss the role of finally block with suitable examples. | Theory | CO4 |
| 4 | Differentiate between built-in and user-defined exceptions. | Theory | CO4 |
| 5 | Explain the importance of exception handling in robust software design. | Theory | CO4 |
| 6 | Discuss steps to create and raise user-defined exceptions in Python. | Theory | CO4 |
| 7 | Explain the role of Python in data analysis and its advantages. | Theory | CO5 |
| 8 | Explain the concept of Series in pandas with examples. | Theory | CO5 |
| 9 | Discuss the concept of DataFrame in pandas with examples. | Theory | CO5 |
| 10 | Compare pandas with NumPy in context of data analysis. | Theory | CO4 |
| 11 | Write a program to demonstrate try-except for division by zero. | Practical | CO4 |
| 12 | Write a program to demonstrate multiple except blocks. | Practical | CO4 |
| 13 | Write a program to illustrate the working of finally block. | Practical | CO4 |
| 14 | Write a program to create a user-defined exception for invalid age. | Practical | CO4 |
| 15 | Write a program to demonstrate raising and handling exceptions. | Practical | CO4 |
| 16 | Write a pandas program to create a Series from a list. | Practical | CO5 |
| 17 | Write a pandas program to create a DataFrame from a dictionary. | Practical | CO5 |
| 18 | Write a pandas program to access rows and columns in a DataFrame. | Practical | CO5 |
| 19 | Write a pandas program to calculate mean and sum of a Series. | Practical | CO5 |
| 20 | Write a pandas program to filter data from a DataFrame. | Practical | CO5 |