**Annexure 1**

**Programming for Data Science**

**(CSL 225)**

**Lab Workbook**



**Faculty name: Dr. Akanksha Kaushik**

**Student name: Chaitali Singh**

**Roll No.: 24csu246**

**Semester: 3**

**Group: AIML 6**

**Department of Computer Science and Engineering**

**NorthCap University, Gurugram- 122001, India**

**Session 2025-26**

**INDEX**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **SN** | **Experiment** | **Page No.** | **Date of**  **Experiment** | **Date of**  **submission** | **Marks** | **CO Covered** | **Sign** |
| **1** | **Write a program to assign different  types of variables and perform math operations.** |  | **7/8/25** |  |  |  |  |
| **2** | **Write a Program to display a number if  it is positive or negative and check if  integer is odd or even. Apply it to n  numbers.** |  | **7/8/25** |  |  |  |  |
| **3** | **Write a program to print each letter of a  word. Repeat this process for at least  five words.** |  | **21/8/25** |  |  |  |  |
| **4** | **Check prime and Armstrong number by  making functions.** |  | **21/8/25** |  |  |  |  |
| **5** | **Program to illustrate use of existing  math functions in Python (log, sin, cos,  abs etc.)** |  | **21/8/25** |  |  |  |  |
| **6** | **Program to illustrate use of existing  string functions in Python** |  |  |  |  |  |  |
| **7** | **Display Factorial of a number using  recursive function.** |  |  |  |  |  |  |
| **8** | **Give the commands to print each fruit  in a fruit list, add a fruit to the list and  remove a fruit from the list.** |  |  |  |  |  |  |
| **9** | **Give the commands to find length of  the list and check if a fruit exists in the  list, create copy of the list.** |  |  |  |  |  |  |
| **10** | **Give the commands to create a tuple of  computer parts, print the tuple. Create  another tuple and join with the existing  one.** |  |  |  |  |  |  |
| **11** | **Conversion of list to tuple, tuple to list,  list to dictionary etc.** |  |  |  |  |  |  |
| **12** | **Give commands to create a dictionary  of brand, model and year, access a  specific item, change value of any one  item.** |  |  |  |  |  |  |
| **13** | **Give the commands to find length of  the dictionary, add an item, remove an  item, create copy of the dictionary.** |  |  |  |  |  |  |
| **14** | **Use NumPy to generate an array of 25  random numbers sampled from a  standard normal distribution. Further  compute the min and max, values of  the array; and their index locations.** |  |  |  |  |  |  |
| **10** | **Give the commands to create a tuple of  computer parts, print the tuple. Create  another tuple and join with the existing  one.** |  |  |  |  |  |  |
| **11** | **Conversion of list to tuple, tuple to list,  list to dictionary etc.** |  |  |  |  |  |  |
| **12** | **Give commands to create a dictionary  of brand, model and year, access a  specific item, change value of any one  item.** |  |  |  |  |  |  |
| **13** | **Give the commands to find length of  the dictionary, add an item, remove an  item, create copy of the dictionary.** |  |  |  |  |  |  |
| **14** | **Use NumPy to generate an array of 25  random numbers sampled from a  standard normal distribution. Further  compute the min and max, values of  the array; and their index locations.** |  |  |  |  |  |  |
| **15** | **Create a class “Person” with atleast five attributes. Create five objects and  display.** |  |  |  |  |  |  |
| **16** | **Write a program to show the OOPS  concepts: Encapsulation, Inheritance  and Polymorphism.** |  |  |  |  |  |  |

**EXPERIMENT NO. 1**

|  |
| --- |
| **Student Name and Roll Number: Chaitali Singh 24csu246** |
| **Semester /Section: 3 E** |
| **Link to Code:** |
| **Date: 7/8/25** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective(s):**   * Perform variable assignment. * Apply command to enter inputs from user * Use comments in code * Use operator precedence and operator associativity. * Define a data type, and understand type coercion vs. type conversion through code. * Effectively use arithmetic expressions in Python |
| **Outcome:**  Student will be familiarizing with the data types and math operations in Python. |
| **Problem Statement:**  Write a program to assign different types of variables and perform math’s operations. |
| **Background Study:**  Python supports integers, floating-point numbers and complex numbers. They are defined as int, float, and complex classes in Python. Integers and floating points are separated by the presence or absence of a decimal point. For instance, 5 is an integer whereas 5.0 is a floating-point number. |
| **Question Bank:**   1. Does Python support both integer and floating-point kinds of numeric values and variables?   Yes   1. How multiple variables are are assigned?   Multiple variables are generally assigned via loops using sequence data types like lists.   1. Explain various data types in one line each in Python.   Python's core data types include: int (whole numbers), float (decimal numbers), complex (numbers with real and imaginary parts), str (sequences of characters/text), bool (True/False values), list (ordered, mutable sequences), tuple (ordered, immutable sequences), dict (unordered key-value pairs), set (unordered collections of unique items), and NoneType (representing the absence of a value).   1. Do we need to declare variables in Python? How?   No, you do not need to explicitly declare variables or their data types in Python before using them. Python is a dynamically typed language   1. What is mixed-type expression? Why is type casting required for it?   Expression with multiple data types. It needs type casting as operations can only be performed on similar data types.   1. What do you mean by type coercion and type conversion? How are they different?   type coercion is an automatic process performed by the system, while type conversion is a manual process initiated by the programmer. While both achieve the same goal of changing data types, explicit type conversion offers more control and can help prevent unexpected behavior that might arise from implicit coercions.   1. In the expression 1+1.5 which type of conversion will happen? (Implicit or Explicit?)   implicit   1. What is a data type? Give some examples of built-in data types in Python?   A data type in Python is a classification that specifies the type of value a variable can hold and the operations that can be performed on that value.  Eg. Int, bool, string, list etc.   1. What do you mean by immutable data types? Give examples.   Data types that cannot be modified after creation. Eg. String, tuple.  10. In Python, a variable must be declared before it is assigned a value. Is this statement True or False?  False |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**#Experiment 1**

**#Variable assignment and math operations program**

**# integers**

**a = 10**

**b = 3**

**# float**

**x = 4.5**

**y = 2.0**

**# string (just for fun, won't be used in math)**

**name = "Math Demo"**

**print("----", name, "----")**

**# basic operations**

**sum\_int = a + b**

**diff\_int = a - b**

**prod\_int = a \* b**

**div\_int = a / b # division gives float**

**print("Integer operations:")**

**print(a, "+", b, "=", sum\_int)**

**print(a, "-", b, "=", diff\_int)**

**print(a, "\*", b, "=", prod\_int)**

**print(a, "/", b, "=", div\_int)**

**print()**

**# float operations**

**sum\_float = x + y**

**pow\_float = x \*\* y**

**mod\_float = x % y**

**print("Float operations:")**

**print(x, "+", y, "=", sum\_float)**

**print(x, "\*\*", y, "=", pow\_float)**

**print(x, "%", y, "=", mod\_float)**

**print()**

**# mixing int and float**

**mix = a \* y + x**

**print("Mixing int and float gives:", mix)**

**EXPERIMENT NO. 2**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Effectively use if statements in Python for selection control. * Effectively implement multi-way selection in Python. |
| **Outcome:**  Students will be familiarized with the use of if-then statements in Python |
| **Problem Statement:**  Write a Program to display a number if it is positive or negative and check if integer is odd or even. Apply it to n numbers**.** |
| **Background Study:**  Decision making is required when we want to execute a code only if a certain condition is satisfied.  The if…elif…else statement is used in Python for decision making. Python if Statement Syntax is  >>> if test expression:  statement(s) |
| **Question Bank:**   1. Once a variable has been properly assigned can its value be changed?   Yes, provided its data type is not immutable.   1. How is the value 2.45x10—-5 expressed as a Python literal?   2.45e-5   1. Can a Python programmer do anything to ensure that a variable’s value can never be changed after its initial assignment?   Yes, store it in an immutable data type   1. Sort the following binary operators in order of high to low precedence: +, -, \*, //, /, %, =.   //,/,%,\*,+,-,=   1. What symbol signifies the beginning of a comment in Python?   #   1. What is Control Flow, Control Statement and Control Structure?   Control Flow: This refers to the order in which individual statements, instructions, or function calls within an imperative program are executed or evaluated. By default, code execution is sequential, moving from top to bottom. However, control flow mechanisms allow for deviations from this linear path, enabling programs to make decisions, repeat actions, or jump to different sections of code.  Control Statement: A control statement is a specific instruction within a programming language that dictates a change in the control flow. These statements implement the logic and conditions that determine the program's execution path.  Control Structure: This term broadly refers to the design or organization of code that utilizes control statements to manage the flow of execution. Control structures are the conceptual frameworks that encapsulate and define how control statements are used to achieve specific program behaviors. Common control structures include:   1. Do a=1 and a==1 denoted the same thing? If not, then what is the difference?   No, a=1 is assigning value 1 to a, and a==1 is checking if a is equal to 1.   1. What is short-circuit (lazy) evaluation?  * Short-circuit evaluation, also known as minimal evaluation or McCarthy evaluation, is a strategy for evaluating Boolean expressions in programming languages. It is a specific instance of lazy evaluation. Eg. **Boolean AND (&&) operator:** If the first operand of a logical AND expression evaluates to false, the entire expression must be false, regardless of the second operand's value. In this case, the second operand is not evaluated.  1. What is the difference between definite loop and an indefinite loop?   Definite loop has a clearly defined end point while indefinite does not.   1. What are the two means of constructing multi-way selection in Python?   Using if elif statements, or dictionaries. |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**#EXPERIMENT 2**

**#Write a Program to display a number if it is positive or negative and check if integer is odd or even. Apply it to n numbers.**

**n=int(input("Enter number of numbers you want to enter : "))**

**for i in range (0,n):**

**number=int(input("Enter a number"))**

**#check odd or even**

**if(number%2==0):**

**print(number," is an even number")**

**else:**

**print(number,"is an odd number")**

**#check positive negative**

**if(number<0):**

**print(number,"is a negative integer")**

**else:**

**print(number,"is a positive integer")**

**EXPERIMENT NO. 3**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Effectively use loop statements in Python for selection control. * Effectively implement multi-way selection in Python. |
| **Outcome:**  Students will be familiarized with the use of loop statements in Python |
| **Problem Statement:**  Write a program to print each letter of a word. Repeat this process for at least five words. |
| **Background Study:**  Loops allow us to repeat set of instructions. Repeating the same steps over and over again is called looping. Python loop Statement Syntax is: |
| **Question Bank:**   1. Is “1” a string literal or variable?   String literal   1. What is the difference between the following two strings? ’n’ and ’\n’?   ‘n’ is printed as n whereas ‘\n’ is newline character and gives a new line.   1. What symbol signifies the beginning of a comment in Python?   #, although “”” “”” are also used for multiline comments   1. What is header, suite and clause in Python?   In Python, the terms "header," "suite," and "clause" are used to describe the structure of compound statements, such as if, for, while, try, and def.   * **Header:**   The header is the first line of a clause within a compound statement. It begins with a specific keyword (like if, for, def) and ends with a colon (:). The header defines the type of control flow or structure being initiated.   * **Suite:**   The suite is the block of code that is controlled by the header. It consists of one or more statements that are indented relative to the header. The indentation visually defines the scope of the suite. A suite can contain simple statements on the same line as the header (separated by semicolons) or, more commonly, multiple indented statements on subsequent lines.   * **Clause:**   A clause is the combination of a header and its corresponding suite. Compound statements are built from one or more clauses. For example, an if statement can have an if clause, followed by optional elif clauses, and an optional else clause. Each of these components constitutes a clause, with its own header and suite.   1. Difference between use of indentation in Python and other languages like C?  * **Python:**   Indentation is mandatory and syntactically significant. It defines code blocks (e.g., within if statements, loops, functions). Incorrect indentation will lead to IndentationError and prevent code execution. This enforces a consistent and readable code structure.   * **C (and similar languages like Java, C++):**   Indentation is primarily for readability and stylistic purposes. Code blocks are defined by delimiters like curly braces {}. While good indentation practices are highly recommended for code clarity, the compiler largely ignores whitespace for block definition, and incorrect indentation will not typically cause syntax errors (though it can lead to logical errors if the visual structure misrepresents the actual code flow).   1. How does break, continue and pass work?   The break statement terminates the current loop or statement, the continue statement skips the remaining code inside a loop for the current iteration only, and the pass statement is a null statement that does nothing.   1. What are python iterators?   In Python, an iterator is an object that implements the iterator protocol.   1. What do you mean by range and what is its use?   In Python, range() is a built-in function that generates an immutable sequence of numbers. It is commonly used for iterating a specific number of times in loops or for creating sequences of integers. |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**#EXPERIMENT 3**

**#Write a program to print each letter of a word. Repeat this process for at least five words.**

**print("You will enter 5 words")**

**l=list()#creating list for words**

**#taking input 5 times**

**for i in range(5):**

**print("Enter word ",i+1)**

**word=input()**

**#adding word to list**

**l.append(word)**

**print("Displaying letters:")**

**for i in l:**

**#nesting loops to iterate through string**

**for n in i:**

**print(n)**

**print("\n")**

**EXPERIMENT NO. 4**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Effectively use loop statements in Python for selection control. * Effectively use while statements in Python for iterative control. |
| **Outcome:**  Students will be familiarized with the use of if-then and loop statements in Python |
| **Problem Statement:**  Check prime and Armstrong number by making functions. |
| **Background Study:**  The if…elif…else statement is used in Python for decision making. Python if Statement Syntax is  >>> if test expression:  statement(s)  Loops allow us to repeat set of instructions. Repeating the same steps over and over again is called looping. Python loop Statement Syntax is: |
| **Question Bank:**   1. What is type conversion in Python?   Type conversion in Python, also known as typecasting, is the process of changing the data type of a value from one type to another. It can be implicit or explicit.   1. Is python case sensitive?   yes   1. What are local variables and global variables in Python?   Local variables exist only in certain parts of the program, whereas global ones are accessible throughout.   1. How to comment multiple lines in python?   Using multiline strings, “”” “””.   1. What is the purpose of is, not and in operators?   Membership operators are used to test for the presence or absence of a value within a sequence   1. Let a =15, what will be the value of the expression 1 <= num <= 10? Is it acceptable? If so why? Is this acceptable in other languages like C?   In Python, this expression is acceptable. It is evaluated as (1 <= num) and (num <= 10). However, in this case num does not exist and has no value and thus evaluates to nothing.  In C and C++, the expression    1≤num≤101 is less than or equal to n u m is less than or equal to 10  1≤𝑛𝑢𝑚≤10  is syntactically valid but is logically incorrect for its intended purpose and will return an unexpected value   1. Which is performed first- arithmetic operators or Boolean operators?   Arithmetic operations   1. not(x and y) is equivalent to not(x) or not(y), which property is this?   De Morgan’s Law |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**#EXPERIMENT 4**

**#Check if number is prime or armstrong without function**

**#do with function later when covered**

**num=int(input("Enter a number : "))**

**# Prime number check**

**if num>1 :**

**for i in range(2, num):**

**if num % i == 0:**

**print(num, "is NOT a prime number")**

**break**

**else:**

**print(num, "is a PRIME number")#not necessary to indent ar same place as if in this case**

**else:**

**print(num, "is NOT a prime number")**

**#Armstrong number**

**l=len(str(num))#getting length of number**

**sum=0 #initalizing sum for armstrong number**

**for i in str(num):#iterating throught string**

**i=int(i)#converting back for calculation**

**sum+=i\*\*l #calculating armstrong number**

**#armstrong number condition and then output**

**if(sum==num):**

**print("Is Armstrong number")**

**else:**

**print("Isn't Armstrong number")**

**EXPERIMENT NO. 5**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Define and use functions in Python * Explain the notion of the side-effects of a function call |
| **Outcome:**  Student will be familiarizing with the math functions in Python. |
| **Problem Statement:**  Program to illustrate use of existing math functions in Python (log, sin, cos, abs etc.) |
| **Background Study:**  Function is a group of statements within a program that perform as specific task. Usually one task of a large program. Functions can be executed in order to perform overall program task |
| **Question Bank:**   1. What are functions in python? 2. What is lamda function? 3. What is range? 4. What is a routine? 5. How is a function defined in Python? 6. Difference between formal and actual parameters? 7. If we pass an integer say num to a function and inside the function the value of num is altered, after returning from the function we print value of num. What is be the output (original value or altered value)? 8. Difference between positional and keyword arguments? 9. What is default argument? 10. What is local scope? What is its lifetime? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**#EXPERIMENT 5**

**#Program to illustrate use of existing math functions in Python (log, sin, cos, abs etc.)**

**# Program to illustrate essential math functions in Python**

**import math**

**print("Math Functions Demo\n")**

**# numbers to use**

**num1 = -25**

**num2 = 16**

**angle = math.pi / 6 # 30 degrees in radians pi/6=30 deg**

**# absolute value**

**print("abs(", num1, ") =", abs(num1))**

**# square root**

**print("sqrt(", num2, ") =", math.sqrt(num2))**

**# powers and exponentials**

**print("pow(2, 5) =", math.pow(2, 5))**

**print("exp(2) =", math.exp(2))**

**# logarithms**

**print("log(", num2, ") =", math.log(num2)) # natural log**

**print("log10(", num2, ") =", math.log10(num2)) # base 10 log**

**# trigonometric functions**

**print("sin(30°) =", math.sin(angle))**

**print("cos(30°) =", math.cos(angle))**

**print("tan(30°) =", math.tan(angle))**

**# inverse trig functions**

**print("asin(0.5) =", math.degrees(math.asin(0.5)), "degrees")**

**print("acos(0.5) =", math.degrees(math.acos(0.5)), "degrees")**

**print("atan(1) =", math.degrees(math.atan(1)), "degrees")**

**# conversions**

**print("degrees(pi/6) =", math.degrees(angle))**

**print("radians(180) =", math.radians(180))**

**# constants**

**print("Value of pi =", math.pi)**

**print("Value of e =", math.e)**

**# rounding functions**

**print("ceil(4.3) =", math.ceil(4.3))**

**print("floor(4.7) =", math.floor(4.7))**

**# factorial, lcm, and gcd(hcf)**

**print("factorial(5) =", math.factorial(5))**

**print("gcd(24, 36) =", math.gcd(24, 36))**

**print("lcm(22,4,6,8) =",math.lcm(22,4,6,8))**

**EXPERIMENT NO. 6**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Define and use functions in Python * Explain the notion of the side-effects of a function call |
| **Outcome:**  Student will be familiarizing with the string functions in Python. |
| **Problem Statement:**  Program to illustrate use of existing string functions in Python |
| **Background Study:**  Function is a group of statements within a program that perform as specific task. Usually one task of a large program. Functions can be executed in order to perform overall program task |
| **Question Bank:**   1. What are functions in python? 2. What is lamda function? 3. What is range? 4. What is global scope? 5. Are global variables considered good practice? Why? 6. Is it possible to call a function with the use of both positional and keyword arguments? If so, is there any rule which should be followed? 7. Difference between the way we call a value returning function and a non value returning function? 8. In a program where can a function be defined? 9. What is default argument? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 7**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Explain the notion of the side-effects of a function call * Explain the concept of keyword and default arguments in Python |
| **Outcome:**  To familiarize students the recursive functions in Python. |
| **Problem Statement:**  Students will be familiarized with the concepts of recursion functions in Python. |
| **Background Study:**  Function is a group of statements within a program that perform as specific task. It is even possible for the function to call itself. These types of construct are termed as recursive functions. The following image shows the working of a recursive function called recurse. |
| **Question Bank:**   1. What are functions in python? How is recursive function different from normal function? 2. Once a variable has been properly assigned can its value be changed? 3. What are the advantages of recursion? 4. What are the disadvantages of recursion? 5. Only problems that are recursively defined can be solved using recursion. True or False? 6. Recursion and iteration are the same programming approach. True or False? 7. What happens if the base condition isn’t defined in recursive programs? 8. Give few examples of problems which can be solved using recursion. |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 8**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Explain what a list is in programming * Describe the typical operations performed on lists * Explain what is meant by list traversal * Effectively create and use lists in Python |
| **Outcome:**  Students will be familiarized with the use of list collections in Python |
| **Problem Statement:**  Give the commands to print each fruit in a fruit list, add a fruit to the list and remove a fruit from the list. |
| **Background Study:**  Lists are used to store multiple items in a single variable. Lists are one of 4 built-in data types in Python used to store collections of data. Lists use square brackets [] to show where the list starts and ends, and they use commas to separate the items inside. |
| **Question Bank:**   1. What is a list? 2. How indexing of any value is done in lists? 3. What is the difference between Python Arrays and lists? 4. What is the difference between tuples and lists? 5. Common list operations? What is list traversal? 6. In a list are insertion and appending same? If not, what is the difference? 7. How will you create a tuple consisting of only one element? 8. Can a tuple be altered? 9. What is the use of slice operation? What is its syntax? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 9**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Effectively create and use lists in Python * Explain the difference between lists and tuples in Python |
| **Outcome:**  Students will be familiarized with the use of tuple collections in Python |
| **Problem Statement:**  Give the commands to find length of the list and check if a fruit exists in the list, create copy of the list. |
| **Background Study:**  Tuples are used to store multiple items in a single variable. Tuple is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Set, and Dictionary, all with different qualities and usage. A tuple is a collection which is ordered and unchangeable. Tuples are written with round brackets. |
| **Question Bank:**   1. What is a list? 2. How indexing of any value is done in lists? 3. What is the difference between Python Arrays and lists? 4. What is the difference between tuples and lists? 5. String is mutable or immutable? 6. What will be the type of the output of (1) and (1,) 7. Difference between find method and in operator? 8. Why is “+” considered as overloaded operator? 9. When does the comparison operator, ==, return True in case of two lists? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 10**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Effectively create and use lists in Python * Explain the difference between lists and tuples in Python |
| **Outcome:**  Students will be familiarized with the use of tuple collections in Python |
| **Problem Statement:**  Give the commands to create a tuple of computer parts, print the tuple. Create another tuple and join with the existing one. |
| **Background Study:**  Tuples are used to store multiple items in a single variable. Tuple is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Set, and Dictionary, all with different qualities and usage. A tuple is a collection which is ordered and unchangeable. Tuples are written with round brackets. |
| **Question Bank:**   1. What is a list? 2. How indexing of any value is done in lists? 3. What is the difference between Python Arrays and lists? 4. What is the difference between tuples and lists? 5. What is the use of range function? 6. What sequence will be generated for - range(1, 3) and range(1,5,2)? 7. How will a loop variable iterate over the index values of a list? 8. Is there any difference between range(5) and range(0,5) 9. How can we make a copy of a list? Why is it needed? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 11**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Effectively create and use dictionary in Python * Explain the difference between lists and dictionaries in Python |
| **Outcome:**  Students will be familiarized with the use of dictionary collections in Python |
| **Problem Statement:**  Give commands to create a dictionary of brand, model and year, access a specific item, change value of any one item. |
| **Background Study:**  Dictionaries are used to store data values in key: value pairs. A dictionary is a collection which is ordered\*, changeable and does not allow duplicates. Dictionaries are written with curly brackets, and have keys and values. |
| **Question Bank:**   1. What is a dictionary in Python? 2. What are Dict and List comprehensions? 3. Is list allowed in dictionary? List of dictionaries is allowed? List of tuples? Tuples of list? Tuples of dictionary? 4. Difference between indexed linear data structures and associative data structure? How is associative data structure provided in Python? 5. Can Dictionaries be created dynamically? How can you create an empty directory, how can you delete a value from dictionary? 6. What is a set? How do you define it? 7. Operations of set? 8. How will you create an empty set? What will happen if you add a duplicate element to a set? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 12**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Effectively create and use dictionary in Python * Explain the difference between lists and dictionaries in Python |
| **Outcome:**  Students will be familiarized with the use of dictionary collections in Python |
| **Problem Statement:**  Give the commands to find length of the dictionary, add an item, remove an item, create copy of the dictionary. |
| **Background Study:**  Dictionaries are used to store data values in key: value pairs. A dictionary is a collection which is ordered\*, changeable and does not allow duplicates. Dictionaries are written with curly brackets, and have keys and values. |
| **Question Bank:**   1. What is a dictionary in Python? 2. What are Dict and List comprehensions? 3. Is list allowed in dictionary? List of dictionaries is allowed? List of tuples? Tuples of list? Tuples of dictionary? 4. Types of set in python? Difference between them? 5. Differences between LIST, TUPLE, DICTIONARY, SET? 6. Can Dictionaries be created dynamically? How can you create an empty directory, how can you delete a value from dictionary? 7. What is dictionary in python and how can we access and element of it? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 13**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Effectively create and use NumPy lists in Python * Generate arrays from existing data, generated data or random data |
| **Outcome:**  Students will be familiarized with the use of NumPy package in Python |
| **Problem Statement:**  Use NumPy to generate an array of 25 random numbers sampled from a standard normal distribution. Further compute the min and max, values of the array; and their index locations. |
| **Background Study:**  NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.  NumPy stands for Numerical Python. |
| **Question Bank:**   1. What is NumPy? 2. Why NumPy is used in python? 3. Where is NumPy used? 4. How to install NumPy in Windows? 5. Why are Numpy arrays better than lists? 6. Types of Numpy arrays and difference between them? 7. What is arange? 8. What is zeros and ones? 9. What is linspace? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 14**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Effectively create and use NumPy lists in Python * To effectively modify and manipulate NumPy arrays |
| **Outcome:**  Students will be familiarized with the use of NumPy package in Python |
| **Problem Statement:**  To perform various operations like reshape, resize, linspace, eye and ones on NumPy. |
| **Background Study:**  NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.  NumPy stands for Numerical Python. |
| **Question Bank:**   1. What are the different operations in NumPy? 2. How is NumPy array different from normal array? 3. Why NumPy is used in python? 4. Why is NumPy Faster Than Lists? 5. What is eye? 6. How can you generate random numbers in Numpy? 7. What is reshape? 8. What are max,min,argmax,argmin functions? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 15**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Explain the concept of an object * Effectively use objects in Python |
| **Outcome:**  Students are familiarized how to create objects and classes. |
| **Problem Statement:**  Create a class “Person” with at least five attributes. Create five objects and display. |
| **Background Study:**  Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior. |
| **Question Bank:**  1. What is the difference between a class and an object?  2. What is the difference between OOP and SOP?  3. What is the difference between a class and a structure?  4. What is class?  5. What is an object?  6. What is the use of period/dot operator in Python objects?  7. How are objects represented in python?  8. What is Inheritance?  9. What is Overloading? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample OutputsEXPERIMENT NO. 16**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Effectively use objects in Python * Effectively use objects and explore the OOPS features |
| **Outcome:**  Students are familiarized how to create objects, classes and implement OOPS. |
| **Problem Statement:**  Write a program to show the OOPS concepts: Encapsulation, Inheritance and Polymorphism. |
| **Background Study:**  Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior. |
| **Question Bank:**   1. What is the difference between a class and an object? 2. What is the difference between OOP and SOP? 3. What is the difference between a class and a structure? 4. What is dereferenced value? How will you get it? 5. How can you verify whether two variables refer to the same object instance or not? What are the main features of OOPs? 6. What are the different types of inheritance? 7. What is polymorphism? 8. What do you mean by overloading> 9. Difference between Overloading and Overriding? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 17**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Effectively create panadas object in Python * Effectively use pandas series |
| **Outcome:**  Students will be familiarized with the use of Pandas package in Python. |
| **Problem Statement:**  To create and perform operations on Pandas Series. |
| **Background Study:**  Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc. |
| **Question Bank:**   1. What is a dataframe? 2. How to create dataframe? 3. What is a pandas series? 4. Name different pandas operations? 5. Difference between Pandas Series and Numpy Arrays? 6. What is the type of a dataframe column? 7. How will you pass a list of column names to dataframe? 8. How will you display a particular row? 9. How will you drop a column? 10. What is the use of groupby function? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample OutputsEXPERIMENT NO. 18**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Effectively create panadas object in Python * Effectively use pandas operations |
| **Outcome:**  Students will be familiarized with the use of Pandas package in Python. |
| **Problem Statement:**  To create and perform operations on Pandas Dataframe using Failed bank Dataset. |
| **Background Study:**  Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc. |
| **Question Bank:**   1. What is the type of a dataframe column? 2. How will you pass a list of column names to dataframe? 3. How will you display a particular row? 4. How will you drop a column? 5. How will you drop rows and columns containing missing values? 6. What is the use of groupby function? 7. How will you use concatenation? 8. What is the use of merge function? 9. How to read from csv file and save data to it? 10. What is joining? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample OutputsEXPERIMENT NO. 19**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Effectively create panadas object in Python * Effectively use pandas operations |
| **Outcome:**  Students will be familiarized with the use of Pandas package in Python. |
| **Problem Statement:**  Import csv, excel, html etc using Pandas and also convert dictionary to dataframe etc. |
| **Background Study:**  Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.  The pandas read\_html() function is a quick and convenient way to turn an HTML table into a pandas DataFrame. This function can be useful for quickly incorporating tables from various websites without figuring out how to scrape the site’s HTML. |
| **Question Bank:**   1. Which are the 3 main ways of combining DataFrames together? 2. How will you use concatenation? 3. What is the use of merge function? 4. What is joining? 5. How to read from csv file and save data to it? 6. Explain Reindexing in pandas? 7. Define the different ways a DataFrame can be created in pandas? 8. Explain Categorical data in Pandas? 9. How will you create a series from dict in Pandas? 10. How can we create a copy of the series in Pandas? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample OutputsEXPERIMENT NO. 20**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Effectively use pandas operations * Effectively use matplotlib functions |
| **Outcome:**  Students will be familiarized with the use of Matplotlib package in Python |
| **Problem Statement:**  Perform data processing with Pandas and Matplotlib library on Failed Bank Dataset. |
| **Background Study:**  Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.  Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. It provides an object-oriented API that helps in embedding plots in applications using Python GUI. |
| **Question Bank:**   1. What is Matplotlib? 2. What is subplot()? 3. What is the Matplotlib Object Oriented Method? 4. What does add\_axes([a,b,c,d]) (where a,b,c,d lie between 0 to 1) mean? 5. Which functions will you use to add x label, y label and title to a axis in a figure? 6. What will subplots(nrows=1, ncols=2) give? 7. How can we configure the range of axes? 8. While giving the value of color, what does 'b.-' mean? What is the other way to define a color? What does alpha indicate? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 21**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Effectively use matplotlib functions * Learn to effectively present data graphically |
| **Outcome:**  Students will be familiarized with the use of Matplotlib package in Python |
| **Problem Statement:**  Perform data processing with Pandas and Matplotlib library on Failed Bank Dataset. |
| **Background Study:**  Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.  Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. It provides an object-oriented API that helps in embedding plots in applications using Python GUI. |
| **Question Bank:**   1. What does add\_axes([a,b,c,d]) (where a,b,c,d lie between 0 to 1) mean? 2. Which functions will you use to add x label, y label and title to a axis in a figure? 3. What will subplots(nrows=1, ncols=2) give? 4. What is bar plots? 5. What is a histogram? 6. What are the different pie charts in matplotlib? 7. What does legend(loc=3) mean? 8. How can you display an legend? 9. How can you save a figure? 10. How can DPI and figure size be specified when the Figure object is created? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**EXPERIMENT NO. 22**

|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **Objective:**   * Effectively use seaborn functions * Learn which type of plot is to be used which with type of data * Learn the correct use of colors in data visualization |
| **Outcome:**  Students will be familiarized with the use of Seaborn package in Python |
| **Problem Statement:**  Perform data processing with Pandas and Seaborn library on Company Sales Bank Dataset. |
| **Background Study:**  Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.  Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. |
| **Question Bank:**   1. What is seaborn? 2. Mention few features of seaborn? 3. What is the function to give color to plot ? 4. How to classify the different ways for using color\_palette() ? 5. How can we view all the available data sets in the Seaborn library ? 6. What is KDE ? How can we plot it? 7. On top of which library Seaborn is built ? 8. How can you install Seaborn? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

**Annexure 2**

**Programming for Data Science**

**CSL225**

Project Report



Faculty name: Student name:

Roll No.:

Semester:

Group:

**Department of Computer Science and Engineering**

**The NorthCap University, Gurugram- 122001, India**

**Session 2025-26**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S.No** |  | **Page No.** |
| **1.** | **Project Description** |  |
| **2.** | **Problem Statement** |  |
| **3.** | **Analysis**  **3.1 Hardware Requirements**  **3.2 Software Requirements** |  |
| **4.** | **Design**  **4.1 Data/Input Output Description:**  **4.2 Algorithmic Approach / Algorithm / DFD / ER diagram/Program Steps** |  |
| **5.** | **Implementation and Testing (stage/module wise)** |  |
| **6.** | **Output (Screenshots)** |  |
| **7.** | **Conclusion and Future Scope** |  |