

|  |  |
| --- | --- |
| **Programming for Data Science** | |
| Lab Manual | |
| **Department of Computer Science and Engineering**  **The NorthCap University, Gurugram** | |
|  |

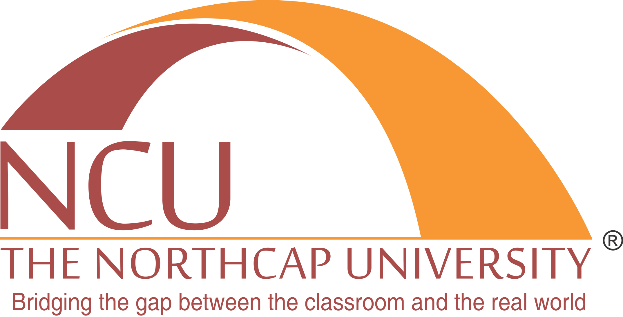
**Programming for Data Science**

**Lab Manual**

**CSL 225**

**Dr. Rita Chhikara**

**Ms. Srishti Vashishtha**



Department of Computer Science and Engineering

NorthCap University, Gurugram- 122001, India

Session 2019-20

*Published by:*

**School of Engineering and Technology**

**Department of Computer Science & Engineering**

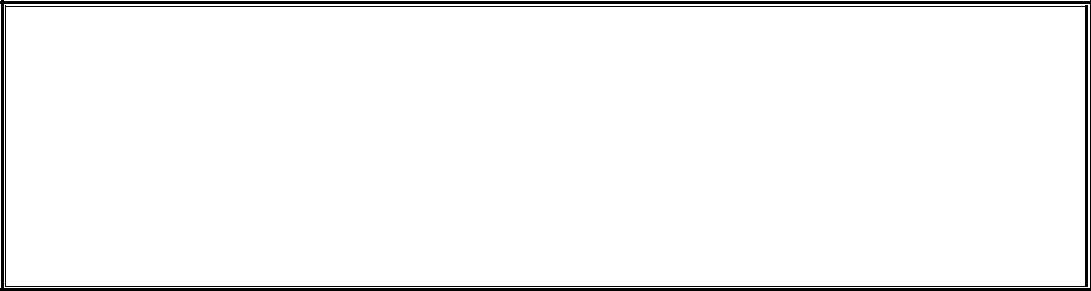
**The NorthCap University Gurugram**

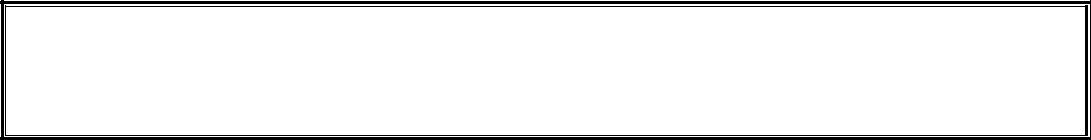
**• Laboratory Manual is for Internal Circulation only**

**© Copyright Reserved**

*No part of this Practical Record Book may be*

*reproduced, used, stored without prior permission of The NorthCap University*



Copying or facilitating copying of lab work comes under cheating and is considered as use of unfair means. Students indulging in copying or facilitating copying shall be awarded zero marks for that particular experiment. Frequent cases of copying may lead to disciplinary action. Attendance in lab classes is mandatory.

Labs are open up to 7 PM upon request. Students are encouraged to make full use of labs beyond normal lab hours.

**PREFACE**

Programming for Data Science Lab Manual is designed to meet the course and program requirements of NCU curriculum for B.Tech II year students of CSE branch. The concept of the lab work is to give brief practical experience for basic lab skills to students. It provides the space and scope for self-study so that students can come up with new and creative ideas.

The Lab manual is written on the basis of “teach yourself pattern” and expected that students who come with proper preparation should be able to perform the experiments without any difficulty. Brief introduction to each experiment with information about self-study material is provided. The laboratory exercises will include installing Jupyter Notebook or Spyder Platform for Python and familiarization with their interface; experiments on strengthening the basics of data visualization. Then, students would be familiarized with different types of operations,lists, tuples, dictionaries in python. The students will apply Object Oriented Programming concepts on real world examples. Students will explore different python packages: NumPy, Pandas, Matplotlib and Seaborn for data pre-processing and data analysis. Finally, the students would require to do guided and unguided project. Students are expected to come thoroughly prepared for the lab.General disciplines, safety guidelines and report writing are also discussed.

The lab manual is a part of curriculum for the TheNorthCap University, Gurugram. Teacher’s copy of the experimental results and answer for the questions are available as sample guidelines.

We hope that lab manual would be useful to students of CSE, IT, ECE and BSc branches and author requests the readers to kindly forward their suggestions / constructive criticism for further improvement of the work book.

Author expresses deep gratitude to Members, Governing Body-NCU for encouragement and motivation.

**Authors**

**The NorthCap University**

**Gurugram, India**

|  |  |  |  |
| --- | --- | --- | --- |
| **CONTENTS** | | | |
| S.N. | **Details** | Page No. | |
|  | **Syllabus** | 6 | |
| 1 | **Introduction** | 9 | |
| 2 | **Lab Requirement** | 10 |
| 3 | **General Instructions** | 11 |
| 4 | **List of Experiments** | 13 |
| 5 | **List of Flip Assignment** | 16 |
| 6 | **List of Projects** | 17 |
| 7 | **Rubrics** | 18 |
| 8 | **Annexure 1 (Format of Lab Report)** | 19 |
| 9 | **Annexure 2 (Format of Project Report)** | 20 |

**SYLLABUS**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. **Department:** | | | **Department of Computer Science and Engineering** | | | | | | |
| 1. **Course Name:** Programming in Data Science | | | | | 1. **Course Code** | 1. **L-T-P** | | | 1. **Credits** |
| CSL225 | 2-0-4 | | | 4 |
| 1. **Type of Course (Check one):** | | | **✓✓**  Programme Core Programme Elective **✓**Open Elective | | | | | | |
| 1. **Pre-requisite(s), if any:** None | | | | | | | | | |
| 1. **Frequency of offering (check one):**   Odd **✓**Even Either semester Every semester | | | | | | | | | |
| 1. **Brief Syllabus:**   Introduction to Computer Science, Computer Algorithms, Computer Hardware , Operating Systems—Bridging Software and Hardware , Limits of Integrated Circuits Technology: Moore’s , Computer Software , Procedural vs. Object-Oriented Programming, Literals , Variables and Identifiers , Operators, Expressions and Data Types, What Is a Control Structure, Boolean Expressions (Conditions), Relational Operators, Membership Operators, Selection Control, Multi-Way Selection, Iterative Control, While Statement , Infinite loops, Definite vs. Indefinite Loops, Boolean Flags and Indefinite Loops, List Structures, Common List Operations, Tuples , Nested Lists, For Loops , While Loops and Lists (Sequences), Assigning and Copying Lists , Dictionary Type in Python, Set Data Type , Program Routines , Defining Functions, More on Functions , Calling Value-Returning Functions, Calling Non-Value-Returning Functions, Parameter Passing, Arguments in Python Default Arguments in Python, Variable Scope, Recursive Function, Module Specification , Top-Down Design, Developing a Modular Design of the Calendar Year Program, Object-Oriented Programming concepts, Numpy - Creation on Array ,Array generation from Uniform distribution, Random array generation, reshaping, maximum and minimum, reshaping, Arithmetic operations, Mathematical functions, Bracket Indexing and Selection, Broadcasting, Indexing a 2D array (matrices); Pandas - Creating a Series - from lists, arrays and dictionaries, Storing data in series from intrinsic sources, Creating DataFrames, Imputation, Grouping and aggregation, Merging, Joining, Concatenation, Find Null Values or Check for Null Values, Reading data from csv, txt, excel, web,Visualization - Installing and setting up visualization libraries, Canvas and Axes, Subplots, Common plots – scatter, histogram, boxplot, Logarithmic scale, Placement of ticks and custom tick labels, Pandas Viz, Style Sheets, Plot type, Area, Barplots, Histograms, Line Plots, Scatter Plots, BoxPlots, Hexagonal Bin Plot, Kernel Density Estimation plot (KDE), Distribution Plots, Categorical Data Plots, Combining Categorical Plots, Matrix Plots, Regression Plots, Grids. | | | | | | | | | |
| **Total lecture, Tutorial and Practical Hours for this course**  **(Take 15 teaching weeks per semester):** 90 hours  The class size is maximum 30 learners. | | | | | | | | | |
| **Lectures:** 30 hours | | | | **Practice** | | | | | |
| **Tutorials :**0 hours | | | | **Lab Work:** 60 hours | |
| 1. **Course Outcomes (COs)**   On successful completion of this course students will be able to: | | | | | | | | | |
| **CO 1** | Understand and implement the basics of programming in Python. | | | | | | | | |
| **CO 2** | Understand and implement the Collections in Python. | | | | | | | | |
| **CO 3** | Apply Object Oriented Programming concepts on real world examples. | | | | | | | | |
| **CO 4** | Apply the Numpy package for numerical calculations in Python. | | | | | | | | |
| **CO 5** | Apply Pandas package for loading and preprocessing data in Python. | | | | | | | | |
| **CO 6** | Implement various data visualization tools of Python on real world data. | | | | | | | | |
| 1. **UNIT WISE DETAILS No. of Units: 5** | | | | | | | | | |
| **Unit Number: 1** | | **Title: Basics of Python Programming** | | | | | **No. of hours: 4** | | |
| **Content Summary:**  Introdcution to Computer Science, Computer Algorithms, Computer Hardware , Operating Systems—Bridging Software and Hardware , Limits of Integrated Circuits Technology: Moore’s , Computer Software , Procedural vs. Object-Oriented Programming, Literals , Variables and Identifiers , Operators, Expressions and Data Types, Operator Precedence, Operator Associativity, What Is a Control Structure, Boolean Expressions (Conditions), Relational Operators, Membership Operators, Selection Control, Indentation in Python, Multi-Way Selection, Iterative Control, While Statement , Infinite loops, Definite vs. Indefinite Loops, Boolean Flags and Indefinite Loops. | | | | | | | | | |
| **Unit Number: 2** | | **Title: Collections** | | | | | **No. of hours: 7** | | |
| **Content Summary:**  List Structures, Common List Operations, List Traversal, Lists (Sequences) in Python , Python List Type , Tuples , Sequences, Nested Lists, Iterating Over Lists (Sequences) in Python, For Loops , While Loops and Lists (Sequences), Assigning and Copying Lists , Dictionary Type in Python, Set Data Type , Program Routines , What Is a Function Routine? , Defining Functions, More on Functions , Calling Value-Returning Functions, Calling Non-Value-Returning Functions, Parameter Passing, Arguments in Python Default Arguments in Python, Variable Scope, What Is a Recursive Function, The Factorial Function, Recursive Problem Solving ,1 Thinking Recursively , MergeSort Recursive Algorithm, Iteration vs. Recursion, Tower of Hanoi , Module Specification , Top-Down Design, Developing a Modular Design of the Calendar Year Program, Specification of the Calendar Year Program Modules. | | | | | | | | | |
| **Unit Number: 3** | | **Title: Object Oriented Programming Concepts, Numpy** | | | | | **No. of hours: 6** | | |
| **Content Summary:**  What Is an Object?, Object References, What Is Object-Oriented Programming? , What Is a Class? , Three Fundamental Features of Object-Oriented Programming , Encapsulation , Defining Classes in Python , Inheritance , Defining Subclasses in Python , Polymorphism , The Use of Polymorphism; Numpy - Array generation from Uniform distribution, Random array generation, reshaping, maximum and minimum, reshaping, Arithmetic operations, Mathematical functions, Bracket Indexing and Selection, Broadcasting, Indexing a 2D array (matrices), Selection. | | | | | | | | | |
| **Unit Number: 4** | | **Title: Data processing with Pandas** | | | | | **No. of hours: 6** | | |
| **Content Summary:**  Creating a Series - from lists, arrays and dictionaries, Storing data in series from intrinsic sources, Differences from numpy array, Using an Index, Creating DataFrames, Selection and Indexing, Creating a new column, Removing Columns, Selecting Rows, Selecting subset of rows and columns, Filtering,, Manipulating Index, Multi-Index and Index Hierarchy, Imputation, Grouping and aggregation, Merging, Joining, Concatenation, Info on Unique Values, Selecting Data, Applying Functions, Get column and index names, Sorting and Ordering a DataFrame, Find Null Values or Check for Null Values, Reading data from csv, txt, excel, web. | | | | | | | | | |
| **Unit Number: 5** | | **Title: Data Visualization in Python** | | | | | **No. of hours: 7** | | |
| **Content Summary:**  Installing and setting up visualization libraries, Procedural methodology of visualization creation ,Oops methodology of visualization creation, Canvas and Axes, Subplots, Figure size, aspect ratio and DPI, Saving figures, Legends, labels and titles, Setting colors, linewidths, linetypes, Line and marker styles, Plot ranges, Common plots – scatter, histogram, boxplot, Logarithmic scale, Placement of ticks and custom tick labels, Pandas Viz, Style Sheets, Plot type, Area, Barplots, Histograms, Line Plots, Scatter Plots, BoxPlots, Hexagonal Bin Plot, Kernel Density Estimation plot (KDE), Distribution Plots, jointplot, pairplot, rugplot, kdeplot, Categorical Data Plots, barplot and countplot, countplot, boxplot VS violinplot, stripplot and swarmplot, Combining Categorical Plots, factorplot, Matrix Plots, Heatmap, clustermap, Regression Plots, Grids, PairGrid, pairplot, Facet Grid, JointGrid, Color. | | | | | | | | | |
| 1. **Brief Description of Self-learning components by students (through books/resource material etc.): Topics: Modular Design** | | | | | | | | | |
| 1. **Books Recommended :**   **Textbooks:**   1. Charles Dierbach., *Introduction to Computer Science using Python*, Wiley Publications, Second Edition, 2015 <https://www.pdfdrive.com/introduction-to-computer-science-using-python-e34771850.html> 2. Mark Lutz ,*Learning Python*, O’Reilly publications , Fifth Edition, 2015 <https://www.pdfdrive.com/learning-python-e18760999.html>   **Reference Books:**   1. Paul Barry, *Head First Python*, Orielly Publications, Second Edition, 2010   **Reference Websites:**  **(nptel, swayam, coursera, edx, udemy, lms, official documentation weblink)**   * [**www.lms.ncuindia.edu/lms**](http://www.lms.ncuindia.edu/lms) * [**https://swayam.gov.in/nd1\_noc19\_cs59/preview**](https://swayam.gov.in/nd1_noc19_cs59/preview) * [**https://www.python.org/**](https://www.python.org/) | | | | | | | | | |

1. **INTRODUCTION**

That ‘learning is a continuous process’ cannot be over emphasized. The theoretical knowledge gained during lecture sessions need to be strengthened through practical experimentation. Thus, practical makes an integral part of a learning process.

The purpose of conducting experiments can be stated as follows:

* To familiarize the students with fundamentals of writing Python scripts.
* Learning and understanding the different collections in python.
* Applying the Object-Oriented Programming concepts on real world examples.
* Applying the NumPy package for numerical calculations in Python.
* Applying the Pandas package for loading and pre-processing data in Python.
* Implementing various data visualization tools of Python on real world data.
* Write Python functions to facilitate code reuse
* Use Python to read and write files
* Make the code robust by handling errors and exceptions properly

1. **LAB REQUIREMENTS**

|  |  |
| --- | --- |
| **Requirements** | **Details** |
| **Software Requirements** | Anaconda Navigator: Jupyter Notebook and Spyder |
| **Operating System** | Any Operating System |
| **Hardware Requirements** | Windows and Linux:  8 GB RAM (Recommended)  80 GB hard disk space |
| **Required Bandwidth** | NA |

1. **GENERAL INSTRUCTIONS** 
   1. **General discipline in the lab**
   * Students must turn up in time and contact concerned faculty for the experiment they are supposed to perform.
   * Students will not be allowed to enter late in the lab.
   * Students will not leave the class till the period is over.
   * Students should come prepared for their experiment.
   * Experimental results should be entered in the lab report format and certified/signed by concerned faculty/ lab Instructor.
   * Students must get the connection of the hardware setup verified before switching on the power supply.
   * Students should maintain silence while performing the experiments. If any necessity arises for discussion amongst them, they should discuss with a very low pitch without disturbing the adjacent groups.
   * Violating the above code of conduct may attract disciplinary action.
   * Damaging lab equipment or removing any component from the lab may invite penalties and strict disciplinary action.
   1. **Attendance**

* Attendance in the lab class is compulsory.
* Students should not attend a different lab group/section other than the one assigned at the beginning of the session.
* On account of illness or some family problems, if a student misses his/her lab classes, he/she may be assigned a different group to make up the losses in consultation with the concerned faculty / lab instructor. Or he/she may work in the lab during spare/extra hours to complete the experiment. No attendance will be granted for such case**.**
  1. **Preparation and Performance**
* Students should come to the lab thoroughly prepared on the experiments they are assigned to perform on that day. Brief introduction to each experiment with information about self-study reference is provided on LMS.
* Students must bring the lab report during each practical class with written records of thelast experiments performed complete in all respect.
* Each student is required to write a complete report of the experiment he has performed and bring to lab class for evaluation in the next working lab. Sufficient space in work book is provided for independent writing of theory, observation, calculation and conclusion.
* Students should follow the Zero tolerance policy for copying / plagiarism. Zero marks will be awarded if found copied. If caught further, it will lead to disciplinary action.
* Refer **Annexure 1** for Lab Report Format.

1. **LIST OF EXPERIMENTS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Title of the Experiment** | **Software used** | **Unit covered** | **CO**  **Covered** | **Time Required** |
|  | Write a program to assign different types of variables and perform maths operations. | Python(Jupyter) | 1 | CO1 | 2 hrs |
|  | 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. | Python(Jupyter) | 1 | CO1 | 2 hrs |
|  | Write a program to print each letter of a word. Repeat this process for atleast five words. | Python(Jupyter) | 1 | CO1 | 2 hrs |
|  | Check prime and Armstrong number by making functions. | Python(Jupyter) | 1 | CO1 | 2 hrs |
|  | Program to illustrate use of existing math functions in Python (log, sin, cos, abs etc.) | Python(Jupyter) | 1 | CO1 | 2 hrs |
|  | Program to illustrate use of existing string functions in Python | Python(Jupyter) | 1 | CO1 | 2 hrs |
|  | Display Factorial of a number using recursive function. | Python(Jupyter) | 1 | CO1 | 2hrs |
|  | Give the commands to print each fruit in a fruit list, add a fruit to the list and remove a fruit from the list. | Python(Jupyter) | 2 | CO2 | 2 hrs |
|  | Give the commands to find length of the list and check if a fruit exists in the list, create copy of the list. | Python(Jupyter) | 2 | CO2 | 2 hrs |
|  | Give the commands to create a tuple of computer parts, print the tuple. Create another tuple and join with the existing one. | Python(Jupyter) | 2 | CO2 | 2 hrs |
|  | Conversion of list to tuple, tuple to list, list to dictionary etc. | Python(Jupyter) | 2 | CO2 | 2 hrs |
|  | Give commands to create a dictionary of brand, model and year, access a specific item, change value of any one item. | Python(Jupyter) | 2 | CO2 | 2 hrs |
|  | Give the commands to find length of the dictionary, add an item, remove an item, create copy of the dictionary. | Python(Jupyter) | 2 | CO2 | 2 hrs |
|  | 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. | Python(Jupyter) | 3 | CO3 | 2 hrs |
|  | Create a class “Person” with atleast five attributes. Create five objects and display. | Python(Jupyter) | 3 | CO3 | 2 hrs |
|  | Write a program to show the OOPS concepts: Encapsulation, Inheritance and Polymorphism. | Python (Jupyter) | 3 | CO3 | 2hrs |
|  | To create and perform operations on Pandas Dataframe using Failed bank Dataset. | Python(Jupyter) | 4 | CO4 | 2 hrs |
|  | Import csv, excel, html etc using Pandas and also convert dictionary to dataframe etc. | Python(Jupyter) | 4 | CO4 | 2 hrs |
|  | Perform data processing with Pandas and Matplotlib library on Company Sales Dataset. | Python(Jupyter) | 4 | CO4 | 2 hrs |
|  | Perform visualization using matplotlib library of Python on Failed bank Dataset. | Python(Jupyter) | 5 | CO5 | 2 hrs |
|  | Perform visualization using Seaborn library of Python on Company Sales Dataset. | Python(Jupyter) | 5 | CO5 | 2 hrs |
| **Value Added Experiments** | | | | | |
|  | Project – Ecommerce dataset | Python(Jupyter) | 1,2,3,4,5 | CO1-CO5 | 4hrs |
|  | Project – SF Salaries dataset | Python(Jupyter) | 1,2,3,4,5 | CO1-CO5 | 4 hrs |
|  | Project – Titanic Visualization | Python(Jupyter) | 1,2,3,4,5 | CO1-CO5 | 4 hrs |

1. **LIST OF FLIP EXPERIMENTS**
2. To perform various operations like reshape, resize, line, space etc. on NumPy
3. To create dataframe and perform operations on Pandas Series.
4. Perform data processing with Pandas and Matplotlib library on Failed Bank Dataset.
5. To create different pie charts and line graphs using Matplotlib library of Python on Failed bank Dataset.
6. To create different pie charts and boxplots using Seaborn library of Python on SF Salaries Dataset.
7. Perform advanced visualization using Seaborn library of Python on Failed bank Dataset.
8. **LIST OF PROJECTS**
9. Project for Failed Banks: Import pandas and read in the banklist.csv file into a dataframe called banks and perform different tasks.
10. Project for SF Salaries: Import pandas and read in the SF Salaries.csv file into a dataframe called sal and perform different tasks to Explore San Francisco city employee salary data.
11. Project Titanic: Time to practice seaborn package skills to recreate the plots below with a famous titanic data set.
12. 911 Calls Capstone Project: For this capstone project analyze some 911 call data to perform different tasks.
13. Project plotly: Exploratory Data Analysis on Pokemon data set found on Kaggle.
14. Polynomial function project: Implement a polynomial "factory" function of degree 2.
15. Location class project: Design a class named Location for locating a maximal value and its location in a two-dimensional list.
16. Making your objects truthy or falsey using bool(): The bool() built-in can be used to obtain the truth value of an object. To define its behavior, you can use the bool() (nonzero() in Python 2.x) special method.
17. Indexing and slicing your objects using []: The [] operator is called the indexing operator and is used in various contexts in Python such as getting the value at an index in sequences, getting the value associated with a key in dictionaries, or obtaining a part of a sequence through slicing. You can change its behavior using the getitem() special method.
18. Reverse Operators: making your classes mathematically correct: While defining the add(), sub(), mul(), and similar special methods allows you to use the operators when your class instance is the left-hand side operand, the operator will not work if the class instance is the right-hand side operand.
19. Tictactoe Project: Tic-tac-toe is a paper-and-pencil game for two players, X and O, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game. Write a code for tictactoe.
20. **RUBRICS**

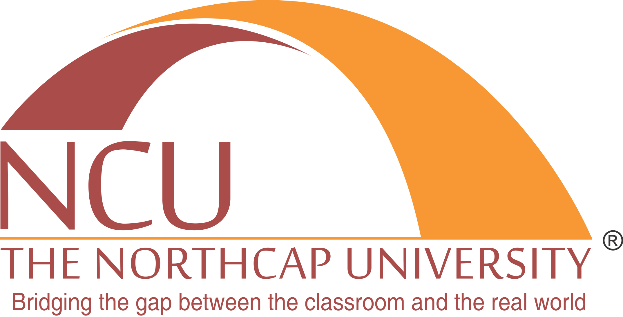
|  |  |
| --- | --- |
| **Marks Distribution** | |
| **Continuous Evaluation (50 Marks)** | **Project Evaluations (20 Marks)** |
| Each experiment shall be evaluated for 10 marks and viva at the end of the semester proportional marks shall be awarded out of total 50. | Both the projects shall be evaluated for 10 marks each and at the end of the semester viva will be conducted related to the projects as well as concepts learned in labs and this component carries 20 marks. |
| Following is the breakup of 10 marks for each  **6 Marks**: Observation & conduct of experiment. Teacher may ask questions about experiment.  **2 Marks:** For completing questions given at the end of each experiment  **2 Marks:** For timely submission. |

**Annexure 1**

**Programming for Data Science**

**(CSL 225)**

**Lab Practical Report**



Faculty name: Student name:

Roll No.:

Semester:

Group:

**Department of Computer Science and Engineering**

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

**Session 2019-20**

**INDEX**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No** | **Experiment** | **Page No.** | **Date of Experiment** | **Date of Submission** | **Marks** | **CO Covered** | **Sign** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**EXPERIMENT NO. 1**

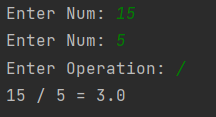
|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **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?  2. How multiple variables are are assigned?  3. Explain various data types in one line each in Python.  4. Do we need to declare variables in Python? How?  5. What is mixed-type expression? Why is type casting required for it?  6. What do you mean by type coercion and type conversion? How are they different?  7. In the expression 1+1.5 which type of conversion will happen?(Implicit or Explicit?)  8. What is a data type? Give some examples of built-in data types in Python?  9. What do you mean by immutable data types? Give examples.  10. In Python, a variable must be declared before it is assigned a value. Is this statement True or False? |

**Student Work Area**

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

num1 = int(input("Enter Num: "))  
num2 = int(input("Enter Num: "))  
oper = input("Enter Operation: ")  
if oper == '+':  
 print(f"{num1} + {num2} = {num1 + num2}")  
elif oper == '-':  
 print(f"{num1} - {num2} = {num1 - num2}")  
elif oper == '\*':  
 print(f"{num1} \* {num2} = {num1 \* num2}")  
elif oper == '/':  
 print(f"{num1} / {num2} = {num1 / num2}")

****

**EXPERIMENT NO. 2**

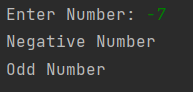
|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **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? 2. How is the value 2.45x10—-5 expressed as a Python literal? 3. Can a Python programmer do anything to ensure that a variable’s value can never be changedafter its initial assignment? 4. Sort the following binary operators in order of high to low precedence: +, -, \*, //, /, %, =. 5. What symbol signifies the beginning of a comment in Python? 6. What is Control Flow, Control Statement and Control Structure? 7. Do a=1 and a==1 denoted the same thing? If not, then what is the difference? 8. What is short-circuit (lazy) evaluation? 9. What is the difference between definite loop and an indefinite loop? 10. What are the two means of constructing multi-way selection in Python? |

**Student Work Area**

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

num = int(input("Enter Number: "))  
if num < 0:  
 print("Negative Number")  
elif num >= 0:  
 print("Positive Number")  
  
if num % 2 == 0:  
 print("Even Number")  
else:  
 print("Odd Number")

****

**EXPERIMENT NO. 3**

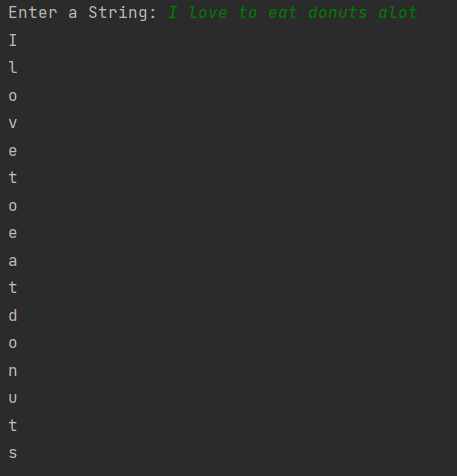
|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **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? 2. What is the difference between the following two strings? ’n’ and ’\n’? 3. What symbol signifies the beginning of a comment in Python? 4. What is header, suite and clause in Python? 5. Difference between use of indentation in Python and other languages like C? 6. How does break, continue and pass work? 7. What are python iterators? 8. What do you mean by range and what is its use? |

**Student Work Area**

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

string = input("Enter a String: ")  
string += " "  
temp = ""  
list1 = []  
for i in string:  
 if i != " ":  
 temp += i  
 elif i == " ":  
 list1.append(temp)  
 temp = ""  
for j in range(0, len(list1)):  
 if j == 5:  
 break  
 else:  
 for k in list1[j]:  
 print(k)

****

**EXPERIMENT NO. 4**

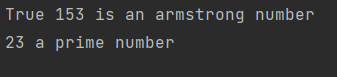
|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **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? 2. Is python case sensitive? 3. What are local variables and global variables in Python? 4. How to comment multiple lines in python? 5. What is the purpose of is, not and in operators? 6. 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? 7. Which is performed first- arithmetic operators or Boolean operators? 8. not(x and y) is equivalent to not(x) or not(y), which property is this? |

**Student Work Area**

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

def armstrong(n):  
 num\_check = n  
 armstrongval = 0  
 while n != 0:  
 temp = n % 10  
 n = n // 10  
 armstrongval += temp \*\* 3  
 if num\_check == armstrongval:  
 print(f"True {num\_check} is an armstrong number")  
 else:  
 print(False)  
  
  
def prime(n):  
 count = 0  
 if n > 1:  
 for i in range(1, n + 1):  
 if (n % i) == 0:  
 count += 1  
 continue  
 else:  
 continue  
 if count > 2:  
 print(f"{n} not a prime number")  
 else:  
 print(f"{n} a prime number")  
  
  
armstrong(153)  
prime(23)

****

**EXPERIMENT NO. 5**

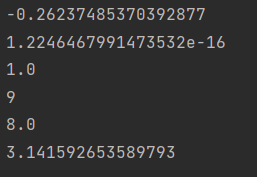
|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **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**

import math  
  
print(math.sin(50))  
print(math.sin(math.pi))  
print(math.log(10, 10))  
print(abs(-9))  
print(math.pow(2, 3))  
print(math.pi)

****

**EXPERIMENT NO. 6**

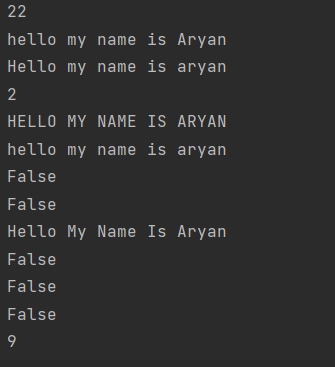
|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **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**

str1 = "hello my name is Aryan"  
print(len(str1))  
print(str1[:])  
print(str1.capitalize())  
print(str1.count("a"))  
print(str1.upper())  
print(str1.lower())  
print(str1.isalpha())  
print(str1.isalnum())  
print(str1.title())  
print(str1.isdigit())  
print(str1.isupper())  
print(str1.islower())  
print(str1.find("name"))

****

**EXPERIMENT NO. 7**

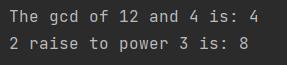
|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **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**

# gcd recursive function  
def gcd(a, b):  
 if b == 0:  
 return abs(a)  
 else:  
 return gcd(b, a % b)  
  
  
a = 12  
b = 4  
print(f"The gcd of 12 and 4 is: {gcd(a, b)}")  
  
  
# power recursive program  
def power(a, b):  
 if b == 0:  
 return 1  
 elif b == 1:  
 return a \* 1  
 else:  
 return a \* power(a, b - 1)  
  
  
print(f"2 raise to power 3 is: {power(2, 3)}")

**EXPERIMENT NO. 8**

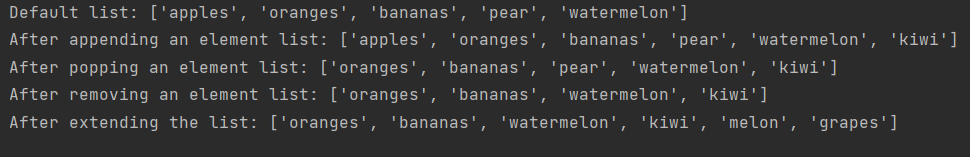
|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **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**

fruitlist = ["apples", "oranges", "bananas", "pear", "watermelon"]  
print(f"Default list: {fruitlist}")  
fruitlist.append("kiwi")  
print(f"After appending an element list: {fruitlist}")  
fruitlist.pop(0)  
print(f"After popping an element list: {fruitlist}")  
fruitlist.remove("pear")  
print(f"After removing an element list: {fruitlist}")  
fruitlist.extend(["melon", "grapes"])  
print(f"After extending the list: {fruitlist}")

****

**EXPERIMENT NO. 9**

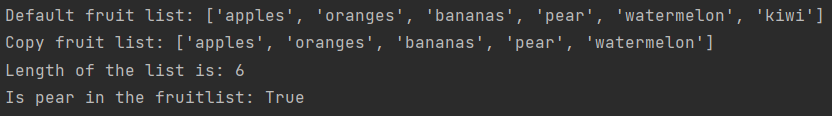
|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **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**

fruitlist = ["apples", "oranges", "bananas", "pear", "watermelon"]  
fruitlistcopy = fruitlist.copy()  
fruitlist.append("kiwi")  
print(f"Default fruit list: {fruitlist}")  
print(f"Copy fruit list: {fruitlistcopy}")  
print(f"Length of the list is: {len(fruitlist)}")  
boolval = "pear" in fruitlist  
print(f"Is pear in the fruitlist?: {boolval}")

****

**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**

tup1 = ("Moniter", "Mouse", "Keyboard", "RAM", "CPU")  
print(f"Tuple of computer parts: {tup1}")  
tup2 = ("Speaker", "Printer", "Joystick")  
tup2 = tup2 + tup1  
print(f"Joined tuple: {tup2}")

****

**EXPERIMENT NO. 11**

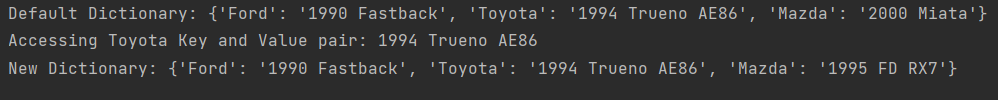
|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **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**

dictcars ={  
 "Ford": "1990 Fastback",  
 "Toyota": "1994 Trueno AE86",  
 "Mazda": "2000 Miata",  
 "Honda": "1997 NSX",  
 "Mitsubishi": "1995 Lan EvoII"  
}  
print(f"Default Dictionary: {dictcars}")  
access = dictcars.get("Toyota")  
print(f"Accessing Toyota Key and Value pair: {access}")  
dictcars["Mazda"] = "1995 FD RX7"  
print(f"New Dictionary: {dictcars}")

****

**EXPERIMENT NO. 12**

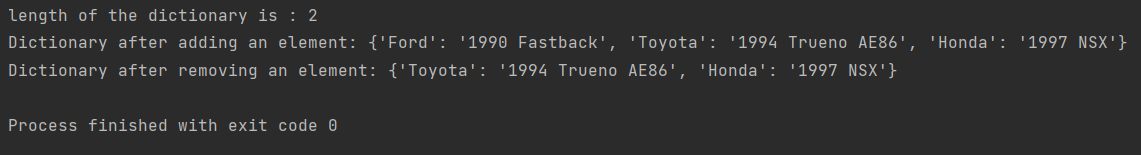
|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **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**

dictcars ={  
 "Ford": "1990 Fastback",  
 "Toyota": "1994 Trueno AE86",  
 "Mazda": "1995 FD RX7"  
}  
print(f"length of the dictionary is : {len(dictcars)}")  
dictcars.update({"Honda": "1997 NSX"})  
print(f"Dictionary after adding an element: {dictcars}")  
dictcars.pop("Ford")  
print(f"Dictionary after removing an element: {dictcars}")

****

**EXPERIMENT NO. 13**

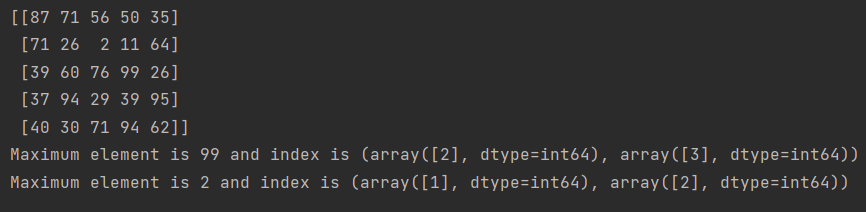
|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks:** |

|  |
| --- |
| **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**

import numpy as np  
x = np.random.randint(100, size=(5,5))  
print(x)  
maxelement = x.max()  
maxindex = np.where(x==maxelement)  
minelement = x.min()  
minindex = np.where(x==minelement)  
print(f"Maximum element is {maxelement} and index is {maxindex}")  
print(f"Maximum element is {minelement} and index is {minindex}")

****

**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 manipulateNumPy 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 2019-20**

**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** |  |