



# **Programming for Data Science**

## **Lab Manual**

**Department of Computer Science and Engineering  
The NorthCap University, Gurugram**

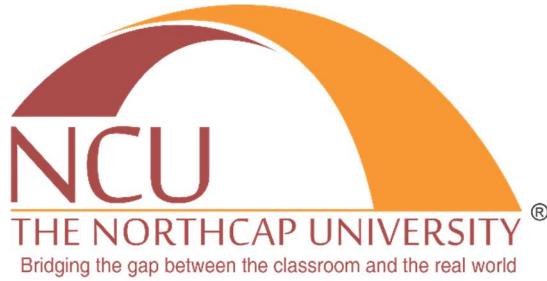
# Programming for Data Science

## 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.	<b>Details</b>	Page No.
	<b>Syllabus</b>	6
1	<b>Introduction</b>	9
2	<b>Lab Requirement</b>	10
3	<b>General Instructions</b>	11
4	<b>List of Experiments</b>	13
5	<b>List of Flip Assignment</b>	16
6	<b>List of Projects</b>	17
7	<b>Rubrics</b>	18
8	<b>Annexure 1 (Format of Lab Report)</b>	19
9	<b>Annexure 2 (Format of Project Report)</b>	20

## SYLLABUS

<b>1. Department:</b>	<b>Department of Computer Science and Engineering</b>		
<b>2. Course Name:</b> Programming in Data Science	<b>3. Course Code</b>	<b>4. L-T-P</b>	<b>5. Credit s</b>
	CSL225	2-0-4	4
<b>6. Type of Course (Check one):</b> Programme Core <input type="checkbox"/> Programme Elective <input checked="" type="checkbox"/> Open Elective			
<b>7. Pre-requisite(s), if any:</b> None			
<b>8. Frequency of offering (check one):</b> Odd <input checked="" type="checkbox"/> Even <input type="checkbox"/> Either semester <input type="checkbox"/> Every semester <input type="checkbox"/>			
<b>9. Brief Syllabus:</b> 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.

		<b>Practice</b>	
<b>Lectures:</b> 30 hours	<b>Tutorials :</b> 0 hours	<b>Lab Work:</b> 60 hours	

**10. Course Outcomes (COs)**

On successful completion of this course students will be able to:

<b>CO 1</b>	Understand and implement the basics of programming in Python.
<b>CO 2</b>	Understand and implement the Collections in Python.
<b>CO 3</b>	Apply Object Oriented Programming concepts on real world examples.
<b>CO 4</b>	Apply the Numpy package for numerical calculations in Python.
<b>CO 5</b>	Apply Pandas package for loading and preprocessing data in Python.
<b>CO 6</b>	Implement various data visualization tools of Python on real world data.

**11. UNIT WISE DETAILS** **No. of Units:** 5

<b>Unit Number: 1</b>	<b>Title: Basics of Python Programming</b>	<b>No. of hours: 4</b>
-----------------------	--	------------------------

**Content Summary:**

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, 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.

<b>Unit Number: 2</b>	<b>Title: Collections</b>	<b>No. of hours: 7</b>
-----------------------	---------------------------	------------------------

**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.

<b>Unit Number: 3</b>	<b>Title: Object Oriented Programming Concepts, Numpy</b>	<b>No. of hours: 6</b>
-----------------------	---	------------------------

**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.

<b>Unit Number: 4</b>	<b>Title: Data processing with Pandas</b>	<b>No. of hours: 6</b>
-----------------------	---	------------------------

**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.

<b>Unit Number: 5</b>	<b>Title: Data Visualization in Python</b>	<b>No. of hours: 7</b>
-----------------------	--	------------------------

**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.

**12. Brief Description of Self-learning components by students (through books/resource material etc.): Topics: Modular Design**

**13. 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/>

## **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

## 2. LAB REQUIREMENTS

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

### **3. GENERAL INSTRUCTIONS**

#### **3.1 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.

#### **3.2 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.

### **3.3 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 the last 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.

## 4. LIST OF EXPERIMENTS

Sr. No.	Title of the Experiment	Software used	Unit covered	CO Covered	Time Required
1.	Write a program to assign different types of variables and perform maths operations.	Python(Jupyter)	1	C01	2 hrs
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.	Python(Jupyter)	1	C01	2 hrs
3.	Write a program to print each letter of a word. Repeat this process for atleast five words.	Python(Jupyter)	1	C01	2 hrs
4.	Check prime and Armstrong number by making functions.	Python(Jupyter)	1	C01	2 hrs
5.	Program to illustrate use of existing math functions in Python (log, sin, cos, abs etc.)	Python(Jupyter)	1	C01	2 hrs
6.	Program to illustrate use of existing string functions in Python	Python(Jupyter)	1	C01	2 hrs
7.	Display Factorial of a number using recursive function.	Python(Jupyter)	1	C01	2hrs
8.	Give the commands to print each fruit in a fruit list, add a fruit to the list and remove a	Python(Jupyter)	2	C02	2 hrs

	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.	Python(Jupyter)	2	C02	2 hrs
10	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	C02	2 hrs
11	Conversion of list to tuple, tuple to list, list to dictionary etc.	Python(Jupyter)	2	C02	2 hrs
12	Give commands to create a dictionary of brand, model and year, access a specific item, change value of any one item.	Python(Jupyter)	2	C02	2 hrs
13	Give the commands to find length of the dictionary, add an item, remove an item, create copy of the dictionary.	Python(Jupyter)	2	C02	2 hrs
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.	Python(Jupyter)	3	C03	2 hrs
15	Create a class “Person” with atleast five attributes. Create	Python(Jupyter)	3	C03	2 hrs

	five objects and display.				
16	Write a program to show the OOPS concepts: Encapsulation, Inheritance and Polymorphism.	Python (Jupyter)	3	C03	2hrs
17	To create and perform operations on Pandas Dataframe using Failed bank Dataset.	Python(Jupyter)	4	C04	2 hrs
18	Import csv, excel, html etc using Pandas and also convert dictionary to dataframe etc.	Python(Jupyter)	4	C04	2 hrs
19	Perform data processing with Pandas and Matplotlib library on Company Sales Dataset.	Python(Jupyter)	4	C04	2 hrs
20	Perform visualization using matplotlib library of Python on Failed bank Dataset.	Python(Jupyter)	5	C05	2 hrs
21	Perform visualization using Seaborn library of Python on Company Sales Dataset.	Python(Jupyter)	5	C05	2 hrs

#### **Value Added Experiments**

1.	Project – Ecommerce dataset	Python(Jupyter)	1,2,3,4,5	C01-C05	4hrs
2.	Project – SF Salaries dataset	Python(Jupyter)	1,2,3,4,5	C01-C05	4 hrs
3.	Project – Titanic Visualization	Python(Jupyter)	1,2,3,4,5	C01-C05	4 hrs

## **5. LIST OF FLIP EXPERIMENTS**

- 1.** To perform various operations like reshape, resize, line, space etc. on NumPy
- 2.** To create dataframe and perform operations on Pandas Series.
- 3.** Perform data processing with Pandas and Matplotlib library on Failed Bank Dataset.
- 4.** To create different pie charts and line graphs using Matplotlib library of Python on Failed bank Dataset.
- 5.** To create different pie charts and boxplots using Seaborn library of Python on SF Salaries Dataset.
- 6.** Perform advanced visualization using Seaborn library of Python on Failed bank Dataset.

## **6. LIST OF PROJECTS**

1. Project for Failed Banks: Import pandas and read in the banklist.csv file into a dataframe called banks and perform different tasks.
2. 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.
3. Project Titanic: Time to practice seaborn package skills to recreate the plots below with a famous titanic data set.
4. 911 Calls Capstone Project: For this capstone project analyze some 911 call data to perform different tasks.
5. Project plotly: Exploratory Data Analysis on Pokemon data set found on Kaggle.
6. Polynomial function project: Implement a polynomial "factory" function of degree 2.
7. Location class project: Design a class named Location for locating a maximal value and its location in a two-dimensional list.
8. 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.
9. 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.

10. 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.
11. 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  $3 \times 3$  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.

## 7. RUBRICS

Marks Distribution	
Continuous Evaluation (50 Marks)	Project Evaluations (20 Marks)
<p>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.</p> <p>Following is the breakup of 10 marks for each</p> <p><b>6 Marks:</b> Observation &amp; conduct of experiment. Teacher may ask questions about experiment.</p> <p><b>2 Marks:</b> For completing questions given at the end of each experiment</p> <p><b>2 Marks:</b> For timely submission.</p>	<p>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.</p>

# **Programming for Data Science**

## **(CSL 225)**

### **Lab Practical Report**



Faculty Name: Dr. Srishti Sharma

Student Name: Piyush Gambhir

Roll No.: 21CSU349

Semester: Semester-III

Group: AIML-B

**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
1.	Experiment - 1		16-08-2022				
2.	Experiment - 2		23-08-2022				
3.	Experiment - 3		26-08-2022				
4.	Experiment - 4		30-08-2022				
5.	Experiment - 5		02-09-2022				
6.	Experiment - 6		06-09-2022				
7.	Experiment - 7		09-09-2022				
8.	Experiment - 8		10-09-2022				
9.	Experiment - 9		13-09-2022				
10.	Experiment - 10		16-09-2022				
11.	Experiment - 11		20-09-2022				
12.	Experiment - 12		03-10-2022				
13.	Experiment - 13		14-10-2022				
14.	Experiment - 14		10-10-2022				
15.	Experiment - 15		21-10-2022				
16.	Experiment - 16		31-10-2022				
17.	Experiment - 17		28-10-2022				
18.	Experiment - 18		01-11-2022				
19.	Experiment -		11-11-2022				

	19						
<b>20.</b>	Experiment - 20		15-11-2022				
<b>21.</b>	Experiment - 21		18-11-2022				
<b>22.</b>	Experiment - 22		17-11-2022				

## EXPERIMENT NO. 1

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### 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:

23. Does Python support both integer and floating-point kinds of numeric values and variables?  
[Yes, python support both integer and floating point-kinds of numeric values and variables.](#)

24. How multiple variables are assigned?  
[Multiple variables are assigned](#)

25. Explain various data types in one line each in Python.  
[Python has the following data types:](#)

26. Do we need to declare variables in Python? How?  
[Python has no command for declaring a variable. A variable is created the moment you first](#)

assign a value to it.

27. What is mixed-type expression? Why is type casting required for it?

If, when executing a certain operator, both operands have different types, then this case is called type mixing.

28. What do you mean by type coercion and type conversion? How are they different?

Type coercion is when the language implicitly convert one object to another type of object in certain circumstances, whereas type conversion is when one object is explicitly converted to another type of object by the user.

29. In the expression  $1+1.5$  which type of conversion will happen? (Implicit or Explicit?)

In the above expression implicit type conversion will happen.

30. What is a data type? Give some examples of built-in data types in Python?

Data type is the type of data a variable can store.

Some built-in data types in python are:

- Int
- Float
- String, etc

31. What do you mean by immutable data types? Give examples.

Immutable data types are basically the data types of which the values cannot be updated.

32. In Python, a variable must be declared before it is assigned a value. Is this statement True or False?

The statement is false, as python allows us to declare the variables at the time of assigning a value.

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

```
a = 14
b = 4.2

print("Number a is", a)
print("Number b is", b)

print()

print("a + b =", a + b)
print("a - b =", a - b)
print("a * b =", a * b)
print("a / b =", a / b)
```

```
Number a is 14
Number b is 4.2
a + b = 18.2
a - b = 9.8
```

## EXPERIMENT NO. 2

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### 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:

- Once a variable has been properly assigned can its value be changed?  
*Yes, value of the variable can be changed after being properly declared.*
- How is the value  $2.45 \times 10^{-5}$  expressed as a Python literal?  
*The value of  $2.45 \times 10^{-5}$  is expressed as  $2.4e-5$  as a python literal.*
- Can a Python programmer do anything to ensure that a variable's value can never be changed after its initial assignment?  
*If it's a class variable that you want to treat as a constant, do what standard library does and PEP8 recommends. Make the variable name all upper-case with underscores between words.*
  - If it's an instance attribute, prefix it with an underscore
  - If you want "read-only" access to the instance attribute, create a property method
  - If it's a local variable, add a comment stating that it should never change
- Sort the following binary operators in order of high to low precedence: +, -, \*, //, /, %, =.

Sorted Binary Operators (high to low precedence): %, //, /, \*, +, -, =.

5. What symbol signifies the beginning of a comment in Python?

'#' has symbol signifies the beginning of a comment in Python.

6. What is Control Flow, Control Statement and Control Structure?

Control flow in programming languages is the order in which individual statement, instructions or function calls will be executed.

7. Do  $a=1$  and  $a==1$  denoted the same thing? If not, then what is the difference?

No, both have different meanings.  $a = 1$  is used to assign variable a value 1 whereas,  $a == 1$  returns a Boolean expression of whether a is equal to 1 or not.

8. What is short-circuit (lazy) evaluation?

Short-circuiting is a programming concept in which the compiler skips the execution or evaluation of some sub-expressions in a logical expression. The compiler stops evaluating the further sub-expressions as soon as the value of the expression is determined.

9. What is the difference between definite loop and an indefinite loop?

Definite loop is a loop which runs finitely whereas, an indefinite loop is a loop which runs infinitely.

10. What are the two means of constructing multi-way selection in Python?

The case control structure is a multi-way selection. Case control structures compare a given value with specified constants and take action according to the first expression to match. Python does not support a case control structure.

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

```
n = int(input("Enter the Number of Number You Wish to Enter: "))
while(n>0):
    number = int(input("Enter a Number: "))

    print("")

    if(number>0):
        print("The Entered Number is Positive.")
    elif(number<0):
        print("The Entered Number is Negative.")
    else:
        print("The Entered Number is Zero.")

    if(number%2 == 0):
        print("The Entered Number is Even.")
    else:
        print("The Entered Number is Odd.")

    n = n - 1
```

Enter the Number of Number You Wish to Enter: 5  
Enter a Number: 23

The Entered Number is Positive.  
The Entered Number is Odd.  
Enter a Number: -8

The Entered Number is Negative.  
The Entered Number is Even.  
Enter a Number: 84

The Entered Number is Positive.  
The Entered Number is Even.  
Enter a Number: 6

The Entered Number is Positive.  
The Entered Number is Even.  
Enter a Number: 21

The Entered Number is Positive.  
The Entered Number is Odd.

## EXPERIMENT NO. 3

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### Objective:

- Effectively use loop statements in Python for selection control.
- Effectively implement multi-way selection in Python.

### Outcome:

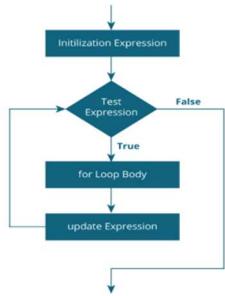
Students will be familiarized with the use of loop statements in Python

### 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:

- Is “1 a string literal or variable?  
[“1 is a literal as a variable cannot start with “.](#)
- What is the difference between the following two strings? ‘n’ and ‘\n’?  
[‘n’ prints n, whereas ‘\n’ print a new line.](#)
- What symbol signifies the beginning of a comment in Python?  
[‘#’ hash symbol signifies the beginning of a comment in Python.](#)
- [What is header, suite and clause in Python?](#)

A clause consists of a header and a 'suite'. The clause headers of a particular compound statement are all at the same indentation level. Each clause header begins with a uniquely identifying keyword and ends with a colon. A suite is a group of statements controlled by a clause. A suite can be one or more semicolon-separated simple statements on the same line as the header, following the header's colon, or it can be one or more indented statements on subsequent lines.

5. Difference between use of indentation in Python and other languages like C?

As indentation in other programming languages like C is used only to make the code or script look neat and better, but in Python, indentation must execute any block of the code as the interpreter notices these 4 spaces which are used for indenting the code and in another language, they use flower brackets or braces.

6. How does break, continue and pass work?

Break, continue and pass statements are also known as control statements.

**Break Statement** - The break statement is used to terminate the loop or statement in which it is present.

**Continue Statement** - Continue is also a loop control statement just like the break statement. continue statement is opposite to that of break statement, instead of terminating the loop, it forces to execute the next iteration of the loop.

**Pass Statement** - The pass statement in Python is used when a statement is required syntactically but you do not want any command or code to execute.

7. What are python iterators?

An iterator is an object that contains a countable number of values. An iterator is an object that can be iterated upon, meaning that you can traverse through all the values.

8. What do you mean by range and what is its use?

The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

```
n = 5
while(n>0):
    word = input("Enter a Word: ")
    for letter in word:
        print(letter)
    n = n - 1
```

Enter a Word: Hello

H

e

l

l

o

Enter a Word: World

W

o

r

l

d

Enter a Word: Python

P

y

t

h

o

n

Enter a Word: Data Science

D

a

t

a

S

c

i

e

n

c

e

Enter a Word: Google

G  
o  
o  
g  
l  
e

## EXPERIMENT NO. 4

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### 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:

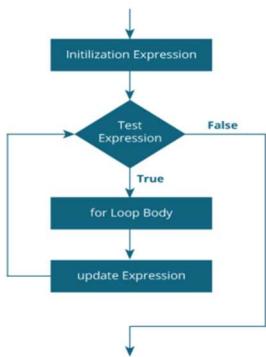
Check prime and Armstrong number by making functions.

### 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:

- What is type conversion in Python?

Type conversion in python is the process of explicitly converting a data type to another data type.

2. Is python case sensitive?

Yes, python is a case sensitive language.

3. What are local variables and global variables in Python?

Variables that are defined inside a function body have a local scope, and those defined outside have a global scope. This means that local variables can be accessed only inside the function in which they are declared, whereas global variables can be accessed throughout the program body by all functions.

4. How to comment multiple lines in python?

To comment out multiple lines in Python, you can prepend each line with a hash ( # ).

5. What is the purpose of is, not and in operators?

is: returns the true value when both the operands are true (Example: "x" is 'x')

not: returns the inverse of the boolean value based upon the operands (example:"1" returns "0" and vice-versa.

in: helps to check if the element is present in a given Sequence or not.

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?

If expressions contain operators from more than one category, arithmetic operators are evaluated first, comparison operators next, and logical operators last. Comparison operators all have equal precedence; they are evaluated in the left-to-right order in which they appear

8. not(x and y) is equivalent to not(x) or not(y), which property is this?

This property is exclusive OR.

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

```
def isArmstrong(n):
    sum = 0
    temp = n
    while temp > 0:
        digit = temp % 10
        sum += digit ** 3
        temp //= 10
    if n == sum:
        return True
    else:
        return False

def isPrime(n):
    if (n == 1):
        return False
    if (n == 2):
        return True
    for i in range(2, n):
        if (n % i == 0):
            return False
    return True

n = int(input("Enter a Number: "))

if isArmstrong(n):
    print(n, "is an Armstrong Number")
else:
    print(n, "is not an Armstrong Number")

if isPrime(n):
    print(n, "is a Prime Number")
else:
    print(n, "is not a Prime Number")
```

131 is not an Armstrong Number

131 is a Prime Number

## EXPERIMENT NO. 5

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### 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?

Functions in python are blocks of code that perform certain tasks and only run when called.

2. What is lambda function?

In Python, a lambda function is a single-line function declared with no name, which can have any number of arguments, but it can only have one expression.

3. What is range?

The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.

4. What is a routine?

Routines is a simple package that provides a single decorator for you to mark functions as tasks (routines) that should be executed on a regular schedule

5. How is a function defined in Python?

A function in python is defined using def keyword.

Syntax:

```
def <function name>():  
    # function body
```

6. Difference between formal and actual parameters?

The major difference between actual and formal parameters is that the actual parameter is used to invoke or pass the information while executing the program whereas formal parameters are declared while calling the function.

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?

Positional arguments must be included in the correct order. Keyword arguments are included with a keyword and equals sign. Positional Arguments An argument is a variable, value or object passed to a function or method as input. Positional arguments are arguments that need to be included in the proper position or order

9. What is default argument?

A default argument is a function argument that has a default value provided to it

10. What is local scope? What is its lifetime?

All variables which are not instance or class variables are known as local variables. Scope: Within the block it is declared. Lifetime: Until control leaves the block in which it is declared

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

```
import math

n = int(input("Enter a Number: "))
print("The log of {} is {}".format(n, math.log(n)))
print("The sin of {} is {}".format(n, math.sin(n)))
print("The cos of {} is {}".format(n, math.cos(n)))
print("The tan of {} is {}".format(n, math.tan(n)))
print("The absoulte value of {} is {}".format(n, abs(n)))
```

The log of 14 is 2.6390573296152584  
The sin of 14 is 0.9906073556948704  
The cos of 14 is 0.1367372182078336  
The tan of 14 is 7.2446066160948055  
The absoulte value of 14 is 14

## EXPERIMENT NO. 6

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### 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?

A function is a block of code which only runs when it is called.

2. What is lambda function?

A lambda function is a small anonymous function. A lambda function can take any number of arguments but can only have one expression.

3. What is range?

The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.

4. What is global scope?

Global scope means the variables can be accessed globally in the entire program.

5. Are global variables considered good practice? Why?

While in many or most other programming languages variables are treated as global if not declared otherwise, Python deals with variables the other way around. They are local, if not otherwise declared. The driving reason behind this approach is that global variables are generally bad practice and should be avoided.

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?

A positional argument means its position matters in a function call. A keyword argument is a

function argument with a name label. Passing arguments as keyword arguments means order does not matter.

7. Difference between the way we call a value returning function and a non-value returning function?

Void functions are created and used just like value-returning functions except they do not return a value after the function executes. In lieu of a data type, void functions use the keyword "void." A void function performs a task, and then control returns back to the caller--but, it does not return a value.

8. In a program where can a function be defined?

A function can be defined where a chunk of code is to be used multiple times.

9. What is default argument?

Default arguments in Python functions are those arguments that take default values if no explicit values are passed to these arguments from the function call.

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

```
str = "Hello World"
print("Original String", str) # Prints complete string
print()

# Prints string in upper case
print("String in Upper Case:", str.upper())
# Prints string in lower case
print("String in Lower Case:", str.lower())
print()

# Prints True if string is in upper case
print("String is in Upper Case:", str.isupper())
# Prints True if string is in lower case
print("String is in Lower Case:", str.islower())
# Prints True if string is in title case
print("String is in Title Case:", str.istitle())
print()

# Prints True if string is in alphabets
print("String is in Alphabets:", str.isalpha())
# Prints True if string is in digits
print("String is in Digits:", str.isdigit())
# Prints True if string is in alphanumeric
print("String is Alphanumeric:", str.isalnum())
# Prints True if string is in space
print("String is in Space:", str.isspace())
# Prints True if string ends with d
print("String Ends With 'd':", str.endswith('d'))
print()

# Prints first character of the string
print(str[0])
# Prints characters starting from 3rd to 5th
print(str[2:5])
# Prints string starting from 3rd character
print(str[2:])
print()

# Prints string two times
print(str * 2)
print()

# Prints concatenated string
print(str + "TEST")
print()

# Prints string with step size 2
print(str[0:5:2])
# Prints string in reverse order
print(str[::-1])
print()
```

```
Original String Hello World
String in Upper Case: HELLO WORLD
String in Lower Case: hello world
String is in Upper Case: False

String is in Lower Case: False
String is in Title Case: True
String is in Alphabets: False
String is in Digits: False
String is Alphanumeric: False
String is in Space: False
String Ends With 'd': True
H
llo
llo World
Hello WorldHello World
Hello WorldTEST
Hlo
dlrow olleH
```

## EXPERIMENT NO. 7

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### Objective:

- Explain the notion of the side-effects of a function call
- Explain the concept of keyword and default arguments in Python

### Outcome:

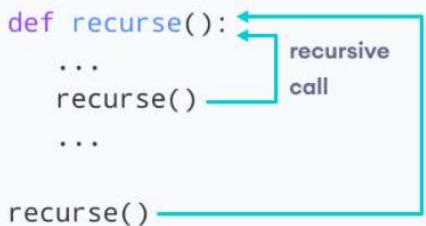
To familiarize students the recursive functions in Python.

### 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 constructs 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?

Functions in python are blocks of code that perform a certain task and are only executed when called.

2. Once a variable has been properly assigned can its value be changed?

Yes, after a variable is properly assigned in python its value can be changed.

3. What are the advantages of recursion?

Advantages of recursion are:

- Reduced the size of code
- Recursive code is easier to debug.

4. What are the disadvantages of recursion?

Disadvantages of recursion:

- Slower than non-recursive function.
- Requires a lot of memory.
- Not effective in terms of space and time

5. Only problems that are recursively defined can be solved using recursion. True or False?

False

6. Recursion and iteration are the same programming approach. True or False?

False

7. What happens if the base condition isn't defined in recursive programs?

If the base condition isn't defined the recursive functions will keep running until the stack memory is full.

8. Give few examples of problems which can be solved using recursion.

Few examples which can be solved using recursion are:

- Fibonacci Series
- Sum of N Natural Numbers, etc.

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

```
def print_fibonacci_series(n):
    if n < 1:
        return []
    if n == 1:
        return [1]
    if n == 2:
        return [1, 1]
    fib = print_fibonacci_series(n-1)
    return fib + [fib[-1] + fib[-2]]


def print_n_natural_numbers(n):
    if n < 1:
        return []
    return print_n_natural_numbers(n-1) + [n]


n = int(input("Enter the Value of n: "))
print("Fibonacci Series: ", end=" ")
for i in print_fibonacci_series(n):
    print(i, end=" ")

print()

n = int(input("Enter the Value of n: "))
print("First n Natural Numbers: ", end=" ")
for i in print_n_natural_numbers(n):
    print(i, end=" ")
```

Fibonacci Series: 1 1 2 3 5 8 13 21 34 55  
First n Natural Numbers: 1 2 3 4 5 6 7 8 9 10

## EXPERIMENT NO. 8

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### 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?

A list is a data structure in Python that is a mutable, or changeable, ordered sequence of elements. Each element or value that is inside of a list is called an item.

2. How is indexing of any value done in lists?

Indexing in a list from left start from 0 and from right starts from -1.

3. What is the difference between Python Arrays and lists?

A list in Python is a collection of items which can contain elements of multiple data types, which may be either numeric, character logical values, etc. It is an ordered collection supporting

negative indexing. A list can be created using [] containing data values.

An array is a vector that contains homogenous items, that is, elements of the same data type. Elements are assigned contiguous memory addresses, allowing for easy change, i.e., addition, deletion, and access to elements.

4. What is the difference between tuples and lists?

Tuples	Lists
Tuples are immutable.	Lists are mutable.
Tuple data type is appropriate for accessing the elements.	The list is better for performing operations, such as insertion and deletion.
Tuple consumes less memory as compared to the list.	Lists consume more memory.

5. Common list operations? What is list traversal?

List operations are the operations that can be performed on the data in the list data structure. A few of the basic list operations used in Python programming are extend(), insert(), append(), remove(), pop(), slice, reverse(), min() & max(), concatenate(), count(), multiply(), sort(), index(), clear(), etc.

Lists are basically equal to arrays in other languages. However, they do have the extra benefit of being dynamic in size. In Python, the list is a kind of container in Data structures. It comes in use for storing numerous data at the same time.

6. In a list are insertion and appending same? If not, what is the difference?

The difference is that with append, you just add a new entry at the end of the list. With insert(position, new\_entry) you can create a new entry exactly in the position you want. The append method adds a new item to the end of a list.

7. How will you create a tuple consisting of only one element?

To create a tuple with only one item, you have add a comma after the item, otherwise Python will not recognize the variable as a tuple.

8. Can a tuple be altered?

No, a tuple cannot be altered as tuple is an immutable data type.

9. What is the use of slice operation? What is its syntax?

The slice() method returns a portion of an iterable as an object of the slice class based on the specified range.

**Syntax:**

`slice(stop)`

`slice(start, stop, step)`

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

```
# Fruit List
fruit_list = ["apple", "banana", "cherry", "durian", "elderberry",
    "fig", "grape", "honeydew", "jackfruit", "kiwi",
    "lemon", "mango", "nectarine", "orange", "papaya",
    "quince", "raspberry", "strawberry", "tomato",
    "ugli fruit", "watermelon"]

# Printing the fruit List
print("Original Fruit List: ", end = " ")
for fruit in fruit_list:
    print(fruit, end = ", ")
print("\n")

# Adding a fruit to the end of the list
fruit_list.append("pineapple")

# Printing the updated fruit list
print("Updated Fruit List: ", end = " ")
for fruit in fruit_list:
    print(fruit, end = ", ")
print("\n")

# Removing the first fruit from the list
fruit_list.pop(0)

# Printing the updated fruit list
print("Updated Fruit List: ", end = " ")
for fruit in fruit_list:
    print(fruit, end = ", ")
print("\n")
```

Original Fruit List: apple, banana, cherry, durian, elderberry, fig, grape, honeydew, jackfruit, kiwi, lemon, mango, nectarine, orange, papaya, quince, raspberry, strawberry, tomato, ugli fruit, watermelon,

Updated Fruit List: apple, banana, cherry, durian, elderberry, fig, grape, honeydew, jackfruit, kiwi, lemon, mango, nectarine, orange, papaya, quince, raspberry, strawberry, tomato, ugli fruit, watermelon, pineapple,

Updated Fruit List: banana, cherry, durian, elderberry, fig, grape, honeydew, jackfruit, kiwi, lemon, mango, nectarine, orange, papaya, quince, raspberry, strawberry, tomato, ugli fruit, watermelon, pineapple,

## EXPERIMENT NO. 9

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### 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?

A list is a data structure in Python that is a mutable, or changeable, ordered sequence of elements.

2. How is indexing of any value done in lists?

The indexing in a list starts from 0 from the left and -1 from the right.

3. What is the difference between Python Arrays and lists?

Arrays	Lists
Only consists of elements belonging to the same data type	Can consist of elements belonging to different data types
Need to explicitly import a module for declaration	No need to explicitly import a module for declaration
Can directly handle arithmetic operations	Cannot directly handle arithmetic operations

4. What is the difference between tuples and lists?

Tuples	Lists
Tuples are immutable.	Lists are mutable.
Tuple data type is appropriate for accessing the elements.	The list is better for performing operations, such as insertion and deletion.
Tuple consumes less memory as compared to the list.	Lists consume more memory.

5. String is mutable or immutable?

Strings in python are immutable.

6. What will be the type of the output of (1) and (1,)

(1,) will give output as tuple and (1) will give output as integer.

7. Difference between find method and in operator?

The find() method of Matcher Class attempts to find the next subsequence of the input sequence that find the pattern. It returns a boolean value showing the same. Whereas, The in operator works with iterable types, such as lists or strings, in Python. It is used to check if an element is found in the iterable. The in operator returns True if an element is found. It returns False if not.

8. Why is “+” considered as overloaded operator?

We use the "+" operator for adding two integers as well as joining two strings or merging two lists. We can achieve this as the "+" operator is overloaded by the "int" class and "str" class.

9. When does the comparison operator, ==, return True in case of two lists?

Comparison operator, ==, return True in case when all the elements of the two lists are equal.

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

```
fruit_list = ['apple', 'banana', 'cherry', 'durian', 'elderberry', 'fig',
              'grape', 'honeydew', 'jackfruit', 'kiwi', 'lemon', 'mango',
              'nectarine', 'orange', 'papaya', 'quince', 'raspberry',
              'strawberry', 'tomato', 'ugli fruit', 'watermelon',
              'xigua', 'yuzu', 'zucchini']

print("Original Fruit List: ", end=" ")
for fruit in fruit_list:
    print(fruit, end=' ')

print("\n\nLength of original fruit list: ", len(fruit_list))

fruit_list_copy = fruit_list.copy()

print("Copied Fruit List: ", end=" ")
for fruit in fruit_list:
    print(fruit, end=' ')

print("\n\nLength of copied fruit listlist: ", len(fruit_list))
```

Original Fruit List: apple banana cherry durian elderberry fig grape honeydew  
jackfruit kiwi lemon mango nectarine orange papaya quince raspberry strawberry  
tomato ugli fruit watermelon xigua yuzu zucchini

Length of original fruit list: 24

'apple' in Fruit List: False

Copied Fruit List: apple banana cherry durian elderberry fig grape honeydew ja  
ckfruit kiwi lemon mango nectarine orange papaya quince raspberry strawberry to  
mato ugli fruit watermelon xigua yuzu zucchini

Length of copied fruit listlist: 24

## EXPERIMENT NO. 10

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### 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?

A list is a data structure in Python that is a mutable, or changeable, ordered sequence of elements.

2. How is indexing of any value done in lists?

The indexing in a list starts from 0 from the left and -1 from the right.

3. What is the difference between Python Arrays and lists?

Arrays	Lists
Only consists of elements belonging to the same data type	Can consist of elements belonging to different data types
Need to explicitly import a module for declaration	No need to explicitly import a module for declaration
Can directly handle arithmetic operations	Cannot directly handle arithmetic operations

4. What is the difference between tuples and lists?

Tuples	Lists
Tuples are immutable.	Lists are mutable.
Tuple data type is appropriate for accessing the elements.	The list is better for performing operations, such as insertion and deletion.
Tuple consumes less memory as compared to the list.	Lists consume more memory.

5. What is the use of range function?

The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.

6. What sequence will be generated for - range(1, 3) and range(1,5,2)?

range(1,3) – 1 2

range(1,5,2) – 1 3

7. How will a loop variable iterate over the index values of a list?

I = []

for i in range(len(I)):

I[i]

8. Is there any difference between range(5) and range(0,5)?

No, both will give the same output in the above case.

9. How can we make a copy of a list? Why is it needed?

We can make a copy of a list using the copy() function. Sometimes, there is a need to reuse any object, hence copy methods are always of great utility.

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

```
computer_parts_1 = ("computer", "monitor")
print("Tuple computer_parts_1:", computer_parts_1)
computer_parts_2 = ("keyboard", "mouse")
print("Tuple computer_parts_2:", computer_parts_2)

computer_parts = computer_parts_1 + computer_parts_2
print("\nTuple computer_parts (Joined Tuple):", computer_parts)
```

```
Tuple computer_parts_1: ('computer', 'monitor')
Tuple computer_parts_2: ('keyboard', 'mouse')

Tuple computer_parts (Joined Tuple): ('computer', 'monitor', 'keyboard', 'mous
e')
```

## EXPERIMENT NO. 11

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### 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?

Dictionaries are used to store data values in key: value pairs. A dictionary is a collection which is ordered\*, changeable and do not allow duplicates.

2. What are Dict and List comprehensions?

Python comprehensions are syntactic sugar constructs that provide a way to build a list, dictionary or set from a starting list, dictionary or set whilst altering or filtering elements.

3. Is list allowed in dictionary? List of dictionaries is allowed. List of tuples? Tuples of list? Tuples of dictionary?

Yes, list is allowed in dictionary but only as values and not key.

4. Difference between indexed linear data structures and associative data structures? How is associative data structure provided in Python?

An indexed linear data structures is a data structure that improves the speed of data retrieval operations the cost of additional writes and storage space to maintain the index data structure. Whereas an Associative Array Data Structure is a collection data structure that facilitates the storage, update, and retrieval of key-value pairs with the same key.

5. Can Dictionaries be created dynamically? How can you create an empty directory, how can you delete a value from dictionary?

Yes, dictionaries in python can be created dynamically. An empty dictionary without any items is written with just two curly braces, like this: {}. The del keyword deletes a key: value pair from a dictionary. Simply access the value that needs to be deleted

6. What is a set? How do you define it?

A Set is an unordered collection data type that is iterable, mutable and has no duplicate elements.

Set is defined as {}.

7. Operations of set?

[Python Set Operations](#). Sets can be used to carry out mathematical set operations like union, intersection, difference, and symmetric difference. We can do this with operators or methods.

8. How will you create an empty set? What will happen if you add a duplicate element to a set?

An empty set in python can be created using the set() function. Sets cannot contain duplicates. Duplicates are discarded when initializing a set. If adding an element to a set, and that element is already contained in the set, then the set will not change

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

```
car_dict = {'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
print("Original Dictionary: ", car_dict)
```

```
car_dict['year'] = 2018
print("Updated Dictionary: ", car_dict)
```

```
Original Dictionary:  {'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
Updated Dictionary:  {'brand': 'Ford', 'model': 'Mustang', 'year': 2018}
```

## EXPERIMENT NO. 12

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### 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?

Dictionaries are used to store data values in key: value pairs. A dictionary is a collection which is ordered\*, changeable and do not allow duplicates.

2. What are Dict and List comprehensions?

Python comprehensions are syntactic sugar constructs that provide a way to build a list, dictionary or set from a starting list, dictionary or set whilst altering or filtering elements.

3. Is list allowed in dictionary? List of dictionaries is allowed. List of tuples? Tuples of list? Tuples of dictionary?

Yes, list is allowed in dictionary but only as values and not key.

4. Types of set-in python? Difference between them?

In python there are basically two types of sets. The set and the frozenset. The set type is mutable, whether the frozenset is immutable.

5. Differences between LIST, TUPLE, DICTIONARY, SET?

List	Tuple	Set	Dictionary
List is a non-homogeneous data structure that stores the elements in single row and multiple rows and columns	Tuple is also a non-homogeneous data structure that stores single row and multiple rows and columns	Set data structure is also non-homogeneous data structure but stores in single row	Dictionary is also a non-homogeneous data structure which stores key value pairs
List can be represented by [ ]	Tuple can be represented by ( )	Set can be represented by { }	Dictionary can be represented by { }
List allows duplicate elements	Tuple allows duplicate elements	Set will not allow duplicate elements	Set will not allow duplicate elements and dictionary doesn't allow duplicate keys.
List can use nested among all	Tuple can use nested among all	Set can use nested among all	Dictionary can use nested among all
Example: [1, 2, 3, 4, 5]	Example: (1, 2, 3, 4, 5)	Example: {1, 2, 3, 4, 5}	Example: {1, 2, 3, 4, 5}

6. Can Dictionaries be created dynamically? How can you create an empty directory, how can you delete a value from dictionary?

Yes, dictionaries in python can be created dynamically. An empty dictionary without any items is written with just two curly braces, like this: {}. The del keyword deletes a key: value pair from a dictionary. Simply access the value that needs to be deleted

7. What is dictionary in python and how can we access and element of it?

An empty dictionary without any items is written with just two curly braces, like this: {}. Keys are unique within a dictionary while values may not be. The values of a dictionary can be of any type, but the keys must be of an immutable data type such as strings, numbers, or tuples.

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

```
car_dict = {'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
print("Original Dictionary: ", car_dict)
print("Length of Original Dictionary: ", len(car_dict))

print()
car_dict_copy = car_dict.copy()
print("Copy of Dictionary: ", car_dict_copy)

print()
# Adding a new key-value pair
car_dict['color'] = 'Red'
print("Updated Original Dictionary: ", car_dict)

# Removing a key-value pair
del car_dict['year']
print("Updated Original Dictionary: ", car_dict)

Original Dictionary: {'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
Length of Original Dictionary: 3

Copy of Dictionary: {'brand': 'Ford', 'model': 'Mustang', 'year': 1964}

Updated Original Dictionary: {'brand': 'Ford', 'model': 'Mustang', 'year': 1964, 'color': 'Red'}
Updated Original Dictionary: {'brand': 'Ford', 'model': 'Mustang', 'color': 'Red'}
```

## EXPERIMENT NO. 13

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### 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?

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

#### 2. Why NumPy is used in python?

Numpy is one of the most commonly used packages for scientific computing in Python. It provides a multidimensional array object, as well as variations such as masks and matrices, which can be used for various math operations.

#### 3. Where is NumPy used?

Numpy is one of the most commonly used packages for scientific computing in Python. It provides a multidimensional array object, as well as variations such as masks and matrices, which can be used for various math operations.

#### 4. How to install NumPy in Windows?

Python is not installed by default in windows operating system. You can download the required version of python from [python.org](http://python.org). Once python is installed successfully, open command prompt and use pip to install numpy.

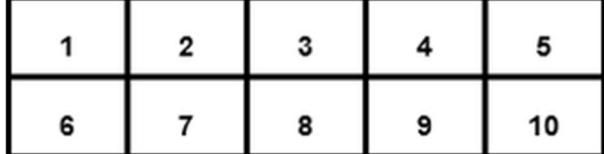
#### 5. Why are Numpy arrays better than lists?

NumPy uses much less memory to store data. The NumPy arrays takes significantly less amount of memory as compared to python lists. It also provides a mechanism of specifying the data types of the contents, which allows further optimisation of the code.

#### 6. Types of Numpy arrays and difference between them?

Types of NumPy Arrays:

1. One-Dimensional Arrays
2. Multi-Dimensional Arrays

One-Dimensional Arrays	Multi-Dimensional Arrays
A one-dimensional array is a type of linear array.	Data in multidimensional arrays are stored in tabular form.
Example: One-Dimensional Arrays: 	Example: Two-Dimensional Array: 

#### 7. What is arange?

The arange() function is used to get evenly spaced values within a given interval. Values are generated within the half-open interval [start, stop]. For integer arguments the function is equivalent to the Python built-in range function, but returns an ndarray rather than a list.

#### 8. What is zeros and ones?

zeros() and ones() are the NumPy library functions to create two different arrays. zeros() function is used to create an array based on the particular shape and type. All array elements are initialized to 0, which is created by the zeros() function. ones() function works like the zeros() function.

#### 9. What is linspace?

linspace is an in-built function in Python's NumPy library. It is used to create an evenly spaced sequence in a specified interval.

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

```
import numpy as np
arr = np.random.randint(1, 100, 24)
print("Generated Array:", arr)
print()

max = np.max(arr)
min = np.min(arr)

print("Maximum Value:", max)
print("Maximum Value Index:", np.where(arr == max))
print()
print("Minimum Value:", min)
print("Minimum Value Index:", np.where(arr == min))
```

Generated Array: [75 23 88 56 46 91 5 73 48 64 62 79 81 92 41 13 71 79 4 92 4  
5 61 37 58]

Maximum Value: 92  
Maximum Value Index: (array([13, 19], dtype=int64),)

Minimum Value: 4  
Minimum Value Index: (array([18], dtype=int64),)

## EXPERIMENT NO. 14

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### 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?

Input arrays for performing arithmetic operations such as add(), subtract(), multiply(), and divide() must be either of the same shape or should conform to array broadcasting rules.

2. How is NumPy array different from normal array?

Numpy arrays facilitate advanced mathematical and other types of operations on large numbers of data.

3. Why NumPy is used in python?

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray , it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important.

4. Why is NumPy Faster Than Lists?

NumPy Arrays are faster than Python Lists because of the following reasons: An array is a collection of homogeneous data-types that are stored in contiguous memory locations. On the

other hand, a list in Python is a collection of heterogeneous data types stored in non-contiguous memory locations.

5. What is `eye`?

`eye()` function in Python is used to return a two-dimensional array with ones (1) on the diagonal and zeros (0) elsewhere.

6. How can you generate random numbers in Numpy?

`choice()` method allows you to generate a random value based on an array of values. The `choice()` method takes an array as a parameter and randomly returns one of the values.

7. What is `reshape`?

Reshaping means changing the shape of an array. The shape of an array is the number of elements in each dimension. By reshaping we can add or remove dimensions or change number of elements in each dimension.

8. What are `max,min,argmax,argmin` functions?

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

In [2]:

```
import numpy as np
```

resize

In [3]:

```
a = np.arange(2, 11).reshape(3, 3)
print(a)
```

```
[[ 2  3  4]
 [ 5  6  7]
 [ 8  9 10]]
```

linspace

In [4]:

```
c = np.linspace(1, 10, 10)
print(c)
```

```
[ 1.  2.  3.  4.  5.  6.  7.  8.  9. 10.]
```

eye

In [5]:

```
d = np.eye(3)
print(d)
```

```
[[1.  0.  0.]
 [0.  1.  0.]
 [0.  0.  1.]]
```

ones

In [6]:

```
e = np.ones((3, 3))
print(e)
```

```
[[1.  1.  1.]
 [1.  1.  1.]
 [1.  1.  1.]]
```

## EXPERIMENT NO. 15

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### 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?

A class is a blueprint for declaring and creating objects. An object is a class instance that allows programmers to use variables and methods from inside the class.

2. What is the difference between OOP and SOP?

The main difference between structured and object oriented programming is that structured programming allows developing a program using a set of modules or functions, while object oriented programming allows constructing a program using a set of objects and their interactions.

3. What is the difference between a class and a structure?

Structures and classes differ in the following particulars: Structures are value types; classes are reference types. A variable of a structure type contains the structure's data, rather than containing a reference to the data as a class type does.

4. What is class?

A Class is like an object constructor, or a "blueprint" for creating objects.

5. What is an object?

Almost everything in Python is an object. Every object has certain attributes and methods. The connection between the attributes or the method with the object is indicated by a “dot” (“.”) written between them.

6. What is the use of period/dot operator in Python objects?

The connection between the attributes or the methods with the object is indicated by a “dot” (“.”) written between them.

7. How are objects represented in python?

Objects are represented as classes which make up a general classification structure including type, identity, and mutability.

8. What is Inheritance?

Inheritance is the procedure in which one class inherits the attributes and methods of another class.

9. What is Overloading?

Overloading is the ability of a function or an operator to behave in different ways based on the parameters that are passed to the function, or the operands that the operator acts on.

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

In [13]:

```
import datetime

class Person:

    def __init__(self, name, surname, birthdate, address, telephone, email):
        self.name = name
        self.surname = surname
        self.birthdate = birthdate
        self.address = address
        self.telephone = telephone
        self.email = email

    def __str__(self):
        return "Name: " + self.name + " " + self.surname + "\n" + "Birthdate: " \
               + str(self.birthdate) + "\n" + "Address: " + self.address + "\n" + \
               "Telephone: " + self.telephone + "\n" + "Email: " + self.email + "\n"

john = Person("John", "Doe", datetime.date(1990, 1, 1),
              "Dwarka", "9895321458", "john@mail.com")
print(john)

dave = Person("Dave", "Noah", datetime.date(1992, 1, 6),
              "Palam", "989532143", "dave@mail.com")
print(dave)
piyush = Person("Piyush", "Gambhir", datetime.date(1991, 6, 2),
                "Gurugram", "9565321458", "piyush@mail.com")
print(piyush)
riten = Person("Riten", "Dhawan", datetime.date(1996, 4, 1),
               "Haryana", "9895644458", "riten@mail.com")
print(riten)
pridhi = Person("Pridhi", "Singh", datetime.date(1996, 11, 3),
                 "Chandigarh", "3433321458", "pridhi@mail.com")
print(pridhi)
```

Name: John Doe  
Birthdate: 1990-01-01  
Address: Dwarka  
Telephone: 9895321458  
Email: john@mail.com

Name: Dave Noah  
Birthdate: 1992-01-06  
Address: Palam  
Telephone: 989532143  
Email: dave@mail.com

Name: Piyush Gambhir  
Birthdate: 1991-06-02  
Address: Gurugram  
Telephone: 9565321458  
Email: piyush@mail.com

Name: Riten Dhawan  
Birthdate: 1996-04-01  
Address: Haryana  
Telephone: 9895644458  
Email: riten@mail.com

Name: Pridhi Singh  
Birthdate: 1996-11-03  
Address: Chandigarh  
Telephone: 3433321458  
Email: pridhi@mail.com

## EXPERIMENT NO. 16

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

<b>Objective:</b> <ul style="list-style-type: none"><li>• Effectively use objects in Python</li><li>• Effectively use objects and explore the OOPS features</li></ul>
<b>Outcome:</b> <p>Students are familiarized how to create objects, classes and implement OOPS.</p>
<b>Problem Statement:</b> <p>Write a program to show the OOPS concepts: Encapsulation, Inheritance and Polymorphism.</p>
<b>Background Study:</b> <p>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.</p>
<b>Question Bank:</b> <ol style="list-style-type: none"><li>1. What is the difference between a class and an object? <i>A class is a blueprint for declaring and creating objects. An object is a class instance that allows programmers to use variables and methods from inside the class.</i></li><li>2. What is the difference between OOP and SOP?</li><li>3. What is the difference between a class and a structure? <i>Structures and classes differ in the following particulars: Structures are value types; classes are reference types. A variable of a structure type contains the structure's data, rather than containing a reference to the data as a class type does.</i></li><li>4. What is dereferenced value? How will you get it?</li><li>5. How can you verify whether two variables refer to the same object instance or not? What are the main features of OOPs? <i>The <code>is</code> keyword is used to test if two variables refer to the same object. The test returns True if</i></li></ol>

the two objects are the same object. The test returns False if they are not the same object, even if the two objects are 100% equal.

Main features of python's object-oriented programming are encapsulation, abstraction, inheritance and polymorphism

6. What are the different types of inheritance?

Types of inheritance in python are:

- Single Inheritance.
- Multiple Inheritance.
- Multilevel Inheritance.
- Hierarchical Inheritance.
- Hybrid Inheritance.

7. What is polymorphism?

Polymorphism in python defines methods in the child class that have the same name as the methods in the parent class.

8. What do you mean by overloading?

Overloading is the ability of a function or an operator to behave in different ways based on the parameters that are passed to the function, or the operands that the operator acts on.

9. Difference between Overloading and Overriding?

Difference Between Method Overloading and Method Overriding in Python. In the process of method overloading, all the functions or methods must contain the same name with varied signatures. In the process of method overriding, all the functions or methods must possess the very same name along with similar signatures.

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

In [1]:

```
# parent class
class Person(object):

    # __init__ is known as the constructor
    def __init__(self, name, idnumber):
        self.name = name
        self.idnumber = idnumber

    def display(self):
        print(self.name)
        print(self.idnumber)

    def details(self):
        print("My name is {}".format(self.name))
        print("IdNumber: {}".format(self.idnumber))

# child class
class Employee(Person):
    def __init__(self, name, idnumber, salary, post):
        self.salary = salary
        self.post = post

        # invoking the __init__ of the parent class
        Person.__init__(self, name, idnumber)

    def details(self):
        print("My name is {}".format(self.name))
        print("IdNumber: {}".format(self.idnumber))
        print("Post: {}".format(self.post))

# creation of an object variable or an instance
a = Employee('Rahul', 886012, 200000, "Intern")

# calling a function of the class Person using
# its instance
a.display()
a.details()
```

```
Rahul
886012
My name is Rahul
IdNumber: 886012
Post: Intern
```

## EXPERIMENT NO. 17

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### Objective:

- Effectively create pandas 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?

A DataFrame is a data structure that organizes data into a 2-dimensional table of rows and columns, much like a spreadsheet.

2. How to create dataframe?

```
# import pandas as pd  
import pandas as pd
```

```
# Calling DataFrame constructor  
df = pd.DataFrame()
```

```
print(df)
```

3. What is a panda's series?

A Pandas Series is like a column in a table. It is a one-dimensional array holding data of any

type.

4. Name different pandas' operations?

There are 4 options here: at, iat, loc and iloc. Among these 'iat' and 'iloc' are similar in the sense they provide integer-based indexing while 'loc' and 'at' provide name-based indexing.

5. Difference between Pandas Series and Numpy Arrays?

Pandas	NumPy
Pandas module works with the tabular data	NumPy module works with numerical data.
Pandas has powerful tools like Series, DataFrame etc.	NumPy has a powerful tool like Arrays.
Pandas consume large memory as compared to NumPy.	NumPy consumes less memory as compared to Pandas.
Pandas provides 2d table object called DataFrame.	NumPy provides a multi-dimensional array.

6. How will you display a particular row?

In the Pandas DataFrame we can find the specified row value with the using function iloc(). In this function we pass the row number as parameter.

7. How will you drop a column?

The most common way to remove a column is using df. drop() . Sometimes, del command in Python is also used.

8. What is the use of groupby function?

Pandas groupby is used for grouping the data according to the categories and apply a function to the categories.

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

In [1]:

```
import pandas as pd
import numpy as np

series = pd.Series([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
print(series)
print("\n")

print(series[:5])
print("\n")

print(series[5:])
print("\n")

print(series[5:8])
print("\n")

print(series[5:8].values)
print("\n")

print(series[5:8].index)
print("\n")

print(series.loc[5:8])
print("\n")

print(series.iloc[5:8])
print("\n")
```

```
0    1
1    2
2    3
3    4
4    5
5    6
6    7
7    8
8    9
9   10
dtype: int64
```

```
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

```
5    6
6    7
7    8
8    9
9   10
```

```
dtype: int64
```

```
5    6
6    7
7    8
dtype: int64
```

```
[6 7 8]
```

```
RangeIndex(start=5, stop=8, step=1)
```

```
5    6
6    7
7    8
8    9
dtype: int64
```

## EXPERIMENT NO. 18

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### Objective:

- Effectively create pandas 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:

What is the type of a dataframe column?

If any column has mixed data types are stored then the data type of the entire column is indicated as object dtype.

How will you pass a list of column names to dataframe?

We can add columns to an existing DataFrame using its columns attribute.

How will you display a particular row?

In the Pandas DataFrame we can print the specified row value with the using function iloc().

How will you drop a column?

The columns can be removed by specifying label names and corresponding axis, or by specifying index or column names directly.

How will you drop rows and columns containing missing values?

The dropna() function is used to remove missing values. Determine if rows or columns which contain missing values are removed. 0, or 'index' : Drop rows which contain missing values. 1, or 'columns' : Drop columns which contain missing value.

What is the use of groupby function?

Pandas groupby is used for grouping the data according to the categories and apply a function to the categories.

What will you use concatenation?

We'll pass two dataframes to pd.concat() method in the form of a list and mention in which axis you want to concat, i.e. axis=0 to concat along rows, axis=1 to concat along columns.

What is the use of merge function?

The merge() method updates the content of two DataFrame by merging them together, using the specified method(s). Use the parameters to control which values to keep and which to replace.

How to read from csv file and save data to it?

To read the CSV file using pandas, we can use the read\_csv() function. Here, the program reads people.csv from the current directory. To write to a CSV file, we need to call the to\_csv() function of a DataFrame.

What is joining?

Inner join is the most common type of join you'll be working with. It returns a dataframe with only those rows that have common characteristics.

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

In [1]:

```
import pandas as pd
```

In [2]:

```
# Read the data
data = pd.read_csv('failed_bank_dataset.csv')
print("Data read into dataframe!")
data
```

Data read into dataframe!

Out[2]:

	Bank Name	City	State	Cert	Acquiring Institution	Closing Date	Fund
0	Almena State Bank	Almena	KS	15426	Equity Bank	23-Oct-20	10538
1	First City Bank of Florida	Fort Walton Beach	FL	16748	United Fidelity Bank, fsb	16-Oct-20	10537
2	The First State Bank	Barboursville	WV	14361	MVB Bank, Inc.	03-Apr-20	10536
3	Ericson State Bank	Ericson	NE	18265	Farmers and Merchants Bank	14-Feb-20	10535
4	City National Bank of New Jersey	Newark	NJ	21111	Industrial Bank	01-Nov-19	10534
...	...	...	...	...	...	...	...
558	Superior Bank, FSB	Hinsdale	IL	32646	Superior Federal, FSB	27-Jul-01	6004
559	Malta National Bank	Malta	OH	6629	North Valley Bank	03-May-01	4648
560	First Alliance Bank & Trust Co.	Manchester	NH	34264	Southern New Hampshire Bank & Trust	02-Feb-01	4647
561	National State Bank of Metropolis	Metropolis	IL	3815	Banterra Bank of Marion	14-Dec-00	4646
562	Bank of Honolulu	Honolulu	HI	21029	Bank of the Orient	13-Oct-00	4645

563 rows × 7 columns

In [3]:

```
# Print the first five rows of the data
print("First five rows of the data:")
data.head(5)
```

First five rows of the data:

Out[3]:

	Bank Name	City	State	Cert	Acquiring Institution	Closing Date	Fund
0	Almena State Bank	Almena	KS	15426	Equity Bank	23-Oct-20	10538
1	First City Bank of Florida	Fort Walton Beach	FL	16748	United Fidelity Bank, fsb	16-Oct-20	10537
2	The First State Bank	Barboursville	WV	14361	MVB Bank, Inc.	03-Apr-20	10536
3	Ericson State Bank	Ericson	NE	18265	Farmers and Merchants Bank	14-Feb-20	10535
4	City National Bank of New Jersey	Newark	NJ	21111	Industrial Bank	01-Nov-19	10534

In [4]:

```
# Print the Last five rows of the data
print("Last five rows of the data:")
data.tail(5)
```

Last five rows of the data:

Out[4]:

	Bank Name	City	State	Cert	Acquiring Institution	Closing Date	Fund
558	Superior Bank, FSB	Hinsdale	IL	32646	Superior Federal, FSB	27-Jul-01	6004
559	Malta National Bank	Malta	OH	6629	North Valley Bank	03-May-01	4648
560	First Alliance Bank & Trust Co.	Manchester	NH	34264	Southern New Hampshire Bank & Trust	02-Feb-01	4647
561	National State Bank of Metropolis	Metropolis	IL	3815	Banterra Bank of Marion	14-Dec-00	4646
562	Bank of Honolulu	Honolulu	HI	21029	Bank of the Orient	13-Oct-00	4645

In [5]:

```
# Print the number of rows and columns of the data
print("Number of Columns in the data:", len(data.columns))
print("Number of Rows in the data:", len(data))
```

Number of Columns in the data: 7  
Number of Rows in the data: 563

In [6]:

```
# Print the column names of the data
print("Column names of the data:")
data.columns
```

Column names of the data:

Out[6]:

```
Index(['Bank Name ', 'City ', 'State ', 'Cert ', 'Acquiring Institution ',
       'Closing Date ', 'Fund'],
      dtype='object')
```

In [7]:

```
# Print the data types of the data
print("Data types of the data:")
data.dtypes
```

Data types of the data:

Out[7]:

Bank Name	object
City	object
State	object
Cert	int64
Acquiring Institution	object
Closing Date	object
Fund	int64
dtype: object	

In [8]:

```
# Print the summary statistics of the data
print("Summary statistics of the data:")
data.describe()
```

Summary statistics of the data:

Out[8]:

	Cert	Fund
<b>count</b>	563.000000	563.000000
<b>mean</b>	31630.271758	10036.875666
<b>std</b>	16443.670177	1115.338358
<b>min</b>	91.000000	4645.000000
<b>25%</b>	20096.500000	10116.500000
<b>50%</b>	32165.000000	10257.000000
<b>75%</b>	35364.000000	10397.500000
<b>max</b>	58701.000000	10538.000000

In [9]:

```
# Print the number of unique values for each column
print("Number of unique values for each column:")
data.unique()
```

Number of unique values for each column:

Out[9]:

Bank Name	546
City	433
State	44
Cert	563
Acquiring Institution	297
Closing Date	258
Fund	563
dtype: int64	

## EXPERIMENT NO. 19

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### Objective:

- Effectively create pandas 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?  
*merge() for combining data on common columns or indices  
.join() for combining data on a key column or an index  
concat() for combining DataFrames across rows or columns*
2. How will you use concatenation?  
*We'll pass two dataframes to pd.concat() method in the form of a list and mention in which axis you want to concat, i.e. axis=0 to concat along rows, axis=1 to concat along columns.*
3. What is the use of merge function?  
*The merge() method updates the content of two DataFrame by merging them together, using the specified method(s). Use the parameters to control which values to keep and which to*

replace.

4. What is joining?

Inner join is the most common type of join you'll be working with. It returns a dataframe with only those rows that have common characteristics.

5. How to read from csv file and save data to it?

To read the CSV file using pandas, we can use the `read_csv()` function. Here, the program reads `people.csv` from the current directory. To write to a CSV file, we need to call the `to_csv()` function of a DataFrame.

6. Explain Reindexing in pandas?

The `reindex()` method allows you to change the row indexes, and the columns labels. Note: The values are set to NaN if the new index is not the same as the old.

7. Define the different ways a DataFrame can be created in pandas?

Different ways to create a DataFrame are:

- Using CSV or Excel Files
- Using NumPy Arrays
- Using Dictionary
- Using List

8. Explain Categorical data in Pandas?

Categoricals are a pandas data type corresponding to categorical variables in statistics. A categorical variable takes on a limited, and usually fixed, number of possible values.

9. How will you create a series from dict in Pandas?

To make a series from a dictionary, simply pass the dictionary to the command `pandas.Series` method. The keys of the dictionary form the index values of the series and the values of the dictionary form the values of the series.

10. How can we create a copy of the series in Pandas?

`copy()` method is used to create a copy of a series object's indices and its data (values). And it returns a copied series object as a result.

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

In [1]:

```
import pandas as pd
```

Reading Data from CSV Files in Python with Pandas

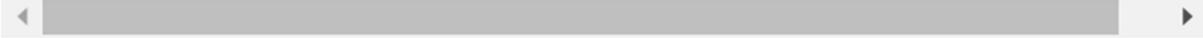
In [2]:

```
df = pd.read_csv('heart.csv')  
df
```

Out[2]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	

1025 rows × 14 columns



Reading excel files in python with pandas

In [3]:

```
import pandas as pd
df = pd.read_excel('heart.xlsx')
print(df)
```

	age	sex	cp	trestbps	chol	fb	restecg	thalach	exang	oldpeak	slo
0	52	1	0	125	212	0	1	168	0	1.0	
1	53	1	0	140	203	1	0	155	1	3.1	
2	70	1	0	145	174	0	1	125	1	2.6	
3	61	1	0	148	203	0	1	161	0	0.0	
4	62	0	0	138	294	1	1	106	0	1.9	
5	58	0	0	100	248	0	0	122	0	1.0	
6	58	1	0	114	318	0	2	140	0	4.4	
7	55	1	0	160	289	0	0	145	1	0.8	
8	46	1	0	120	249	0	0	144	0	0.8	
9	54	1	0	122	286	0	0	116	1	3.2	
10	71	0	0	112	149	0	1	125	0	1.6	

Reading HTML content in Python with pandas

In [4]:

```
df = pd.read_html('https://trends.builtwith.com/websitelist/Responsive-Tables')
df[0]
```

Out[4]:

	Unnamed: 0	Website	Location	Sales Revenue	Tech Spend	Social	Emplk
0	NaN	spotrac.com	United States	NaN	\$250+	20,000+	
1	NaN	newsletter.someecards.com	United States	NaN	\$250+	1,000,000+	
2	NaN	sendy.sketch.io	United States	NaN	\$500+	50+	
3	NaN	supersoluce.com	France	NaN	\$100+	250+	
4	NaN	live.infoq.com	Canada	NaN	\$500+	10,000+	

5	NaN	visual-paradigm.com	Hong Kong	NaN	\$250+	100+
6	NaN	southeastern.edu	United States	NaN	\$1000+	5,000+
7	NaN	stifel.com	United States	NaN	\$5000+	1,000+ 1
8	NaN	sendy.icons8.com	United States	NaN	\$1000+	5,000+
9	NaN	mail2.3dissue.com	India	NaN	\$2000+	500+
10	NaN	me.snacksafely.com	United States	NaN	\$250+	2,000+
11	NaN	shop.sailboatowners.com	United States	NaN	\$500+	500+
12	NaN	nl.infomoney.com.br	Brazil	NaN	\$5000+	200,000+
13	NaN	sendy.smoreapp.co	United States	NaN	\$250+	NaN
14	NaN	newsletter.ilovefreesoftware.com	NaN	NaN	\$100+	1,000+
15	NaN	openbook.illinoisccontroller.gov	United States	NaN	\$2000+	NaN
16	NaN	email.shopsavvy.com	United States	NaN	\$250+	2,000+
17	NaN	subscribe.vegan.com	NaN	NaN	\$250+	150,000+
18	NaN	sendy.silkworth.net	United States	NaN	\$250+	100+
19	NaN	iapp.org	United States	NaN	\$5000+	1,000+
20	NaN	sendy.swarajyamag.com	NaN	NaN	\$500+	10,000+
21	NaN	scan2cad.com	United Kingdom	\$14k+	\$250+	10+
22	NaN	infinityagents.com	United States	NaN	\$250+	NaN

## Converting dictionary to Pandas DataFrame

In [5]:

```
dict = {'Name':['Tom', 'Jack', 'Steve', 'Ricky'], 'Age':[28,34,29,42]}
df = pd.DataFrame(dict)
df
```

▶

Out[5]:

	Name	Age
0	Tom	28
1	Jack	34
2	Steve	29
3	Ricky	42

## EXPERIMENT NO. 20

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### 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?

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

2. What is subplot()?

subplots method provides a way to plot multiple plots on a single figure. Given the number of rows and columns , it returns a tuple ( fig , ax ), giving a single figure fig with an array of axes ax.

3. What is the Matplotlib Object Oriented Method?

Introduction to the Object Oriented Method. The main idea in using the more formal Object

Oriented method is to create figure objects and then just call methods or attributes off of that object. This approach is nicer when dealing with a canvas that has multiple plots on it.

4. What does `add_axes([a,b,c,d])` (where a,b,c,d lie between 0 to 1) mean?

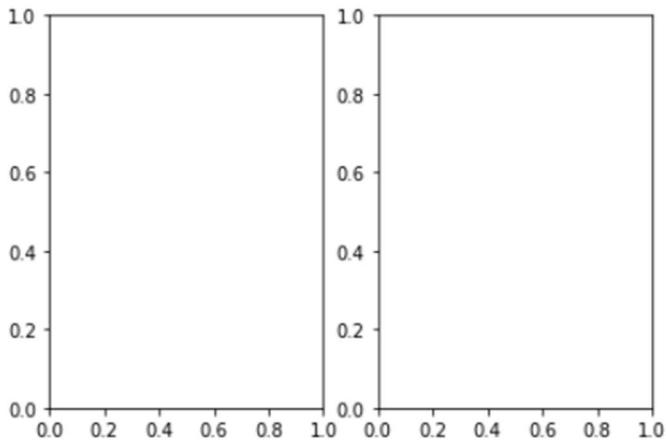
The calling signature of `add_axes` is `add_axes(rect)`, where `rect` is a list `[x0, y0, width, height]` denoting the lower left point of the new axes in figure coordinates `(x0,y0)` and its width and height.

5. Which functions will you use to add x label, y label and title to a axis in a figure?

Add axis labels to the chart by using the `xlabel` and `ylabel` functions.

6. What will `subplots(nrows=1, ncols=2)` give?

```
ax = plt.subplots(nrows=1,ncols=2)
```



7. How can we configure the range of axes?

Matplotlib sets the default range of the axis by finding extreme values (i.e.minimum and maximum) on that axis.

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

```
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
```

```
# Read the data
data = pd.read_csv('banklist.csv')
data
```

	Bank Name	City	State	Cert	Acquiring Institution	Closing Date	Fund
0	Almena State Bank	Almena	KS	15426	Equity Bank	23-Oct-20	10538
1	First City Bank of Florida	Fort Walton Beach	FL	16748	United Fidelity Bank, fsb	16-Oct-20	10537
2	The First State Bank	Barboursville	WV	14361	MVB Bank, Inc.	3-Apr-20	10536
3	Ericson State Bank	Ericson	NE	18265	Farmers and Merchants Bank	14-Feb-20	10535
4	City National Bank of New Jersey	Newark	NJ	21111	Industrial Bank	1-Nov-19	10534
...	...	...	...	...	...	...	...
558	Superior Bank, FSB	Hinsdale	IL	32646	Superior Federal, FSB	27-Jul-01	6004
559	Malta National Bank	Malta	OH	6629	North Valley Bank	3-May-01	4648
560	First Alliance Bank & Trust Co.	Manchester	NH	34264	Southern New Hampshire Bank & Trust	2-Feb-01	4647
561	National State Bank of Metropolis	Metropolis	IL	3815	Banterra Bank of Marion	14-Dec-00	4646
562	Bank of Honolulu	Honolulu	HI	21029	Bank of the Orient	13-Oct-00	4645

563 rows × 7 columns

```
# Print Number of Unique Values in each column
for col in data.columns:
    print(col, data[col].nunique())
```

Bank Name 546  
City 433  
State 44  
Cert 563  
Acquiring Institution 297  
Closing Date 258  
Fund 563

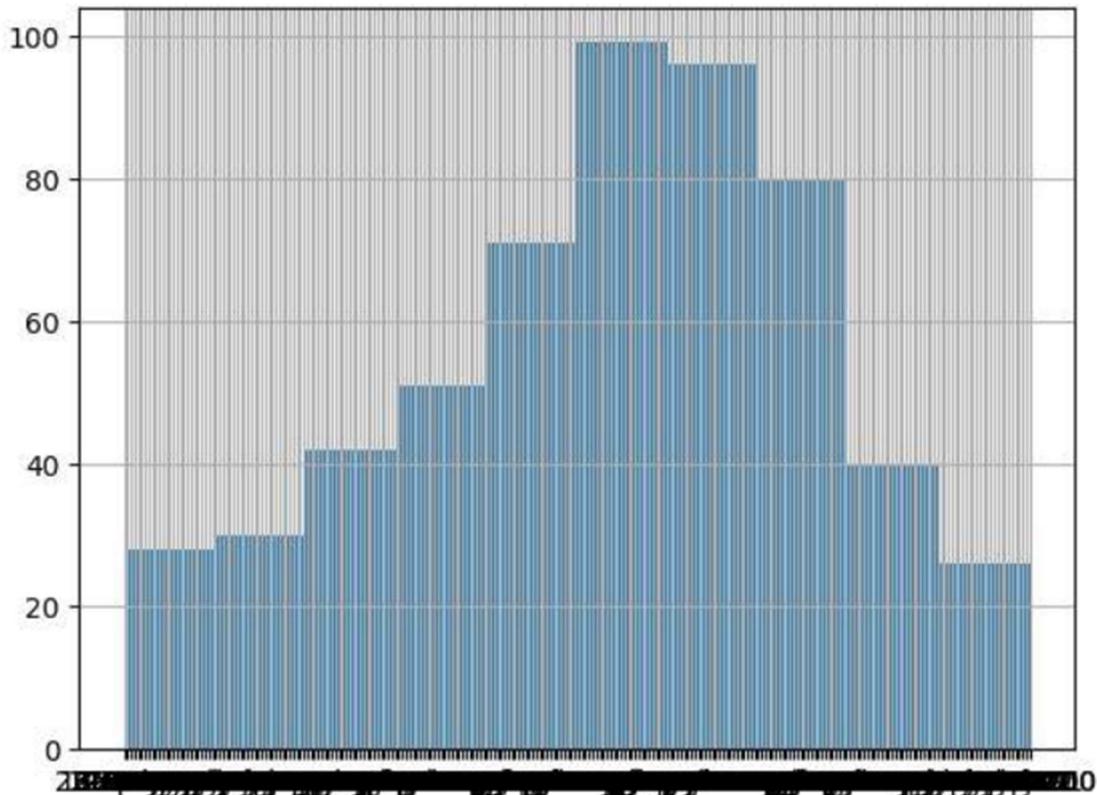
```
# Print Number of Missing Values in each column
for col in data.columns:
    print(col, data[col].isnull().sum())
```

Bank Name 0  
City 0  
State 0  
Cert 0  
Acquiring Institution 0  
Closing Date 0  
Fund 0

```
# Print Number of Unique Values in each column
for col in data.columns:
    print(col, data[col].nunique())
```

Bank Name 546  
City 433  
State 44  
Cert 563  
Acquiring Institution 297  
Closing Date 258  
Fund 563

```
# Plot the histogram of the column 'Closing Date'  
data[ 'Closing Date' ].hist()  
plt.show()
```



## EXPERIMENT NO. 21

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### 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:

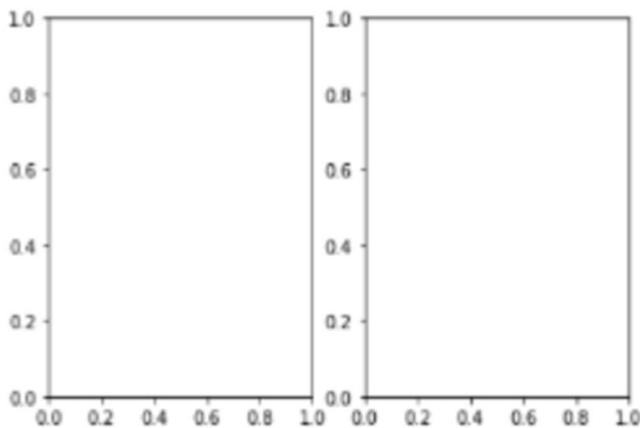
Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, 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?  
The calling signature of `add_axes` is `add_axes(rect)`, where `rect` is a list `[x0, y0, width, height]` denoting the lower left point of the new axes in figure coordinates `(x0,y0)` and its width and height.
2. Which functions will you use to add x label, y label and title to a axis in a figure?  
Add axis labels to the chart by using the `xlabel` and `ylabel` functions.
3. What will `subplots(nrows=1, ncols=2)` give?

```
ax = plt.subplots(nrows=1, ncols=2)
```



#### 4. What is bar plots?

A barplot (or barchart) is one of the most common types of graphic. It shows the relationship between a numeric and a categoric variable.

#### 5. What is a histogram?

A histogram is a chart that uses bars to represent frequencies which helps visualize distributions of data.

#### 6. What are the different pie charts in matplotlib?

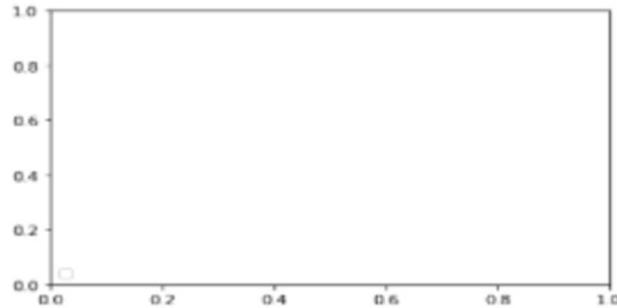
- Start angle
- Explode
- Shadow
- Color
- Legend

#### 7. What does legend(loc=3) mean?

```
from matplotlib import pyplot as plt
from matplotlib import style
plt.legend(loc=3)

No artists with labels found to put in legend. Not
() is called with no arguments.

<matplotlib.legend.Legend at 0x25c091a6a30>
```



8. How can you display a legend?

`plt.legend()`

9. How can you save a figure?

`Matplotlib plots can be saved as image files using the plt. savefig() function.`

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

```
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
```

```
# Read the data
data = pd.read_csv('banklist.csv')
data
```

	Bank Name	City	State	Cert	Acquiring Institution	Closing Date	Fund
0	Almena State Bank	Almena	KS	15426	Equity Bank	23-Oct-20	10538
1	First City Bank of Florida	Fort Walton Beach	FL	16748	United Fidelity Bank, fsb	16-Oct-20	10537
2	The First State Bank	Barboursville	WV	14361	MVB Bank, Inc.	3-Apr-20	10536
3	Ericson State Bank	Ericson	NE	18265	Farmers and Merchants Bank	14-Feb-20	10535
4	City National Bank of New Jersey	Newark	NJ	21111	Industrial Bank	1-Nov-19	10534
...	...	...	...	...	...	...	...
558	Superior Bank, FSB	Hinsdale	IL	32646	Superior Federal, FSB	27-Jul-01	6004
559	Malta National Bank	Malta	OH	6629	North Valley Bank	3-May-01	4648
560	First Alliance Bank & Trust Co.	Manchester	NH	34264	Southern New Hampshire Bank & Trust	2-Feb-01	4647
561	National State Bank of Metropolis	Metropolis	IL	3815	Banterra Bank of Marion	14-Dec-00	4646
562	Bank of Honolulu	Honolulu	HI	21029	Bank of the Orient	13-Oct-00	4645

563 rows × 7 columns

```
# Print Number of Unique Values in each column
for col in data.columns:
    print(col, data[col].nunique())
```

Bank Name 546  
City 433  
State 44  
Cert 563  
Acquiring Institution 297  
Closing Date 258  
Fund 563

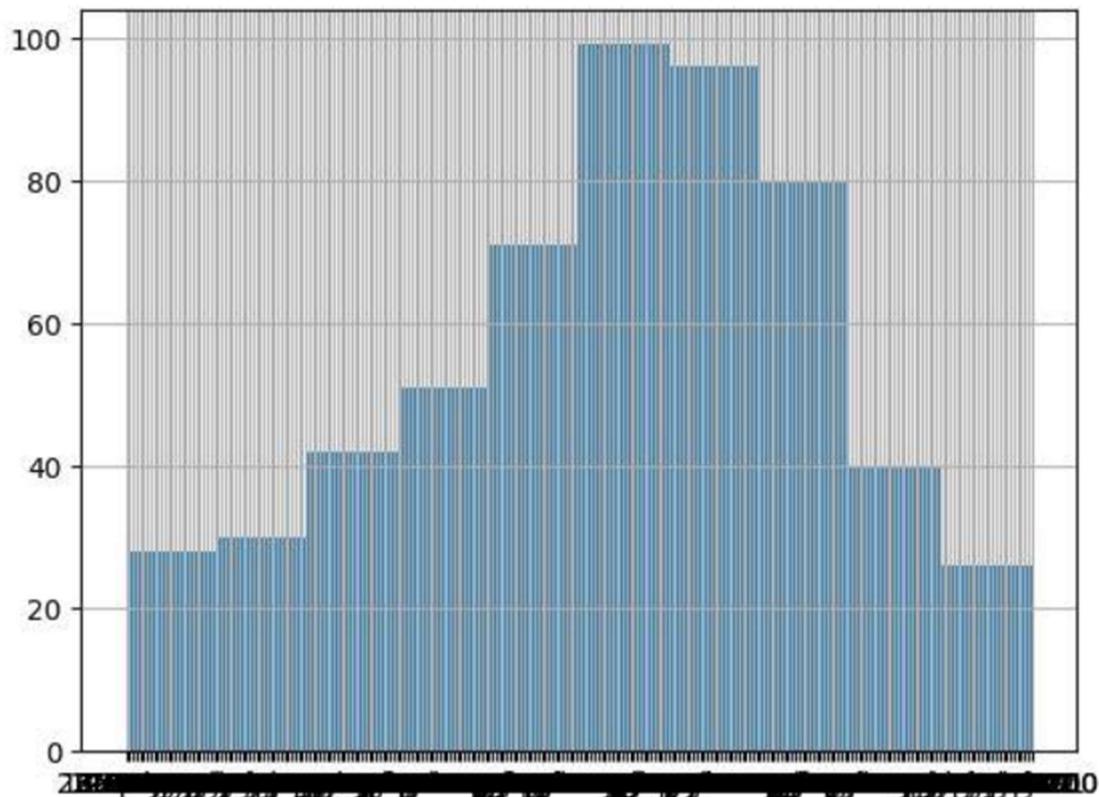
```
# Print Number of Missing Values in each column
for col in data.columns:
    print(col, data[col].isnull().sum())
```

Bank Name 0  
City 0  
State 0  
Cert 0  
Acquiring Institution 0  
Closing Date 0  
Fund 0

```
# Print Number of Unique Values in each column
for col in data.columns:
    print(col, data[col].nunique())
```

Bank Name 546  
City 433  
State 44  
Cert 563  
Acquiring Institution 297  
Closing Date 258  
Fund 563

```
# Plot the histogram of the column 'Closing Date'  
data[ 'Closing Date' ].hist()  
plt.show()
```



## EXPERIMENT NO. 22

<b>Student Name and Roll Number:</b> Piyush Gambhir – 21CSU349
<b>Semester /Section:</b> CS-III - AIML-B
<b>Link to Code:</b> <a href="https://github.com/Piyush-Gambhir/PFDS-Lab-Manual-Semester-III">Piyush-Gambhir/PFDS-Lab-Manual-Semester-III (github.com)</a>
<b>Date:</b>
<b>Faculty Signature:</b>
<b>Marks:</b>

### Objective:

- Effectively use seaborn functions
- Learn which type of plot is to be used with which 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?

Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions.

2. Mention few features of seaborn?

- Built in themes for styling matplotlib graphics.
- Visualizing univariate and bivariate data.
- Fitting in and visualizing linear regression models.
- Plotting statistical time series data.

- Seaborn works well with NumPy and Pandas data structures.
3. What is the function to give color to plot?
- Seaborn provides a function called `color_palette()`, which can be used to give colors to plots and adding more aesthetic value to it.
4. How to classify the different ways for using `color_palette()` ?
- We will classify the different ways for using `color_palette()` types
- Qualitative
  - Sequential
  - Diverging
5. How can we view all the available data sets in the Seaborn library?
- Use `get_dataset_names()` to see a list of available datasets.
6. What is KDE ? How can we plot it?
- KDE Plot described as Kernel Density Estimate is used for visualizing the Probability Density of a continuous variable.
- We can plot KDE using syntax:  
`seaborn.kdeplot(data)`
7. On top of which library Seaborn is built?
- Seaborn is a library for making statistical graphics in Python. It builds on top of `matplotlib` and integrates closely with `pandas` data structures.
8. How can you install Seaborn?
- We can install Seaborn by writing the following command in command prompt:
- ```
pip install seaborn
```

## Student Work Area

### Algorithm/Flowchart/Code/Sample Outputs

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Read the data
df = pd.read_csv('Project_Dataset_SF Salaries.csv')
df
```

|        |        |                   |                                                | Id        | EmployeeName | JobTitle | BasePay   | OvertimePay | OtherPay | Benefits |
|--------|--------|-------------------|------------------------------------------------|-----------|--------------|----------|-----------|-------------|----------|----------|
| 0      | 1      | NATHANIEL FORD    | GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY | 167411.18 |              |          | 0.00      | 400184.25   |          | NaN      |
| 1      | 2      | GARY JIMENEZ      | CAPTAIN III (POLICE DEPARTMENT)                | 155966.02 |              |          | 245131.88 | 137811.38   |          | NaN      |
| 2      | 3      | ALBERT PARDINI    | CAPTAIN III (POLICE DEPARTMENT)                | 212739.13 |              |          | 106088.18 | 16452.60    |          | NaN      |
| 3      | 4      | CHRISTOPHER CHONG | WIRE ROPE CABLE MAINTENANCE MECHANIC           | 77916.00  |              |          | 56120.71  | 198306.90   |          | NaN      |
| 4      | 5      | PATRICK GARDNER   | DEPUTY CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)  | 134401.60 |              |          | 9737.00   | 182234.59   |          | NaN      |
| ...    | ...    | ...               | ...                                            | ...       | ...          | ...      | ...       | ...         | ...      | ...      |
| 148649 | 148650 | Roy I Tillery     | Custodian                                      | 0.00      |              |          | 0.00      | 0.00        |          | 0.0      |
| 148650 | 148651 | Not provided      | Not provided                                   | NaN       |              |          | NaN       | NaN         |          | NaN      |
| 148651 | 148652 | Not provided      | Not provided                                   | NaN       |              |          | NaN       | NaN         |          | NaN      |
| 148652 | 148653 | Not provided      | Not provided                                   | NaN       |              |          | NaN       | NaN         |          | NaN      |
| 148653 | 148654 | Joe Lopez         | Counselor, Log Cabin Ranch                     | 0.00      |              |          | 0.00      | -618.13     |          | 0.0      |

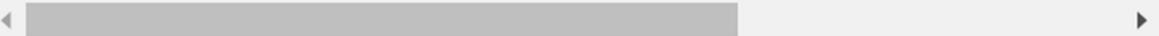
148654 rows × 13 columns



0

```
| # Check the head of the DataFrame.  
| df.head()
```

| 0 | 1 | NATHANIEL FORD    | GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY | 167411.18 | 0.00      | 400184.25 | NaN | 567595.4 |  |
|---|---|-------------------|------------------------------------------------|-----------|-----------|-----------|-----|----------|--|
| 1 | 2 | GARY JIMENEZ      | CAPTAIN III (POLICE DEPARTMENT)                | 155966.02 | 245131.88 | 137811.38 | NaN | 538909.2 |  |
| 2 | 3 | ALBERT PARDINI    | CAPTAIN III (POLICE DEPARTMENT)                | 212739.13 | 106088.18 | 16452.60  | NaN | 335279.9 |  |
| 3 | 4 | CHRISTOPHER CHONG | WIRE ROPE CABLE MAINTENANCE MECHANIC           | 77916.00  | 56120.71  | 198306.90 | NaN | 332343.6 |  |
| 4 | 5 | PATRICK GARDNER   | DEPUTY CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)  | 134401.60 | 9737.00   | 182234.59 | NaN | 326373.1 |  |



```
| # Use the .info() method to find out how many entries there are.  
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 148654 entries, 0 to 148653  
Data columns (total 13 columns):  
 #   Column           Non-Null Count   Dtype     
---  --  
 0   Id               148654 non-null    int64  
 1   EmployeeName     148654 non-null    object  
 2   JobTitle         148654 non-null    object  
 3   BasePay          148045 non-null    float64  
 4   OvertimePay      148650 non-null    float64  
 5   OtherPay         148650 non-null    float64  
 6   Benefits          112491 non-null    float64  
 7   TotalPay         148654 non-null    float64  
 8   TotalPayBenefits 148654 non-null    float64  
 9   Year              148654 non-null    int64  
 10  Notes             0 non-null        float64  
 11  Agency            148654 non-null    object  
 12  Status            0 non-null        float64  
dtypes: float64(8), int64(2), object(3)  
memory usage: 14.7+ MB
```

```
| # Print Number of unique values in each column
| for col in df.columns:
|     print(col, df[col].nunique())
```

```
Id 148654
EmployeeName 110811
JobTitle 2159
BasePay 109489
OvertimePay 65998
OtherPay 83225
Benefits 98465
TotalPay 138486
TotalPayBenefits 142098
Year 4
Notes 0
Agency 1
Status 0
```

```
| # What is the average BasePay
| df['BasePay'].mean()
```

```
66325.4488404877
```

```
| # What is the highest amount of OvertimePay in the dataset
| df['OvertimePay'].max()
```

```
245131.88
```

# **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.   | <b>Project Description</b>                                                                                                                         |          |
| 2.   | <b>Problem Statement</b>                                                                                                                           |          |
| 3.   | <b>Analysis</b><br><br><b>3.1 Hardware Requirements</b><br><br><b>3.2 Software Requirements</b>                                                    |          |
| 4.   | <b>Design</b><br><br><b>4.1 Data/Input Output Description:</b><br><br><b>4.2 Algorithmic Approach / Algorithm / DFD / ER diagram/Program Steps</b> |          |
| 5.   | <b>Implementation and Testing (stage/module wise)</b>                                                                                              |          |
| 6.   | <b>Output (Screenshots)</b>                                                                                                                        |          |
| 7.   | <b>Conclusion and Future Scope</b>                                                                                                                 |          |