**Language Basics:**

1. Print a greeting message saying “Welcome to Python – It is a great language!!! “
2. Define a variable and display the type of that variable
3. Write a program that asks the user for a first name and then asks the user for a last name. The program should print the first and last names together on the same line.
4. Write a program that prompts for a string and a number and prints out the string the number of times indicated by the number.
5. Enter an email address from the user and check if the address is valid (hint – check for the presence of @ sign)
6. Input your age into a variable called age. Have the computer printout how old you'll be eight years from now, ten years from now, twenty-four years from now.
7. Write a program that asks a user for five flowers, and then store them in a list called flowerlist and print them out in alphabetical order.
8. You are a shopkeeper who sells fruits. All the fruits that you sell have been stored in a dictionary as name vs price. You read through the dictionary and let your customers know about the details.
9. Ask the user to input a number between 1 to 10.Give the user 3 chances to enter the number correctly. If the user enters correctly then print the number entered by the user. Else quit with an error message saying “Sorry, your chances are over!!!”

**Modularization:**

1. How will you find the contents in a module?
2. Create a function that takes a person's name and prints "Hello, <person's name>."
3. Create a function that takes in a number and returns the square of that number.
4. Create a function that takes name of students and their scores as arguments. The number of scores entered for a student might vary. Inside the function calculate the total score of the student and print the message “Hi <name>, Your total score is <totalscore>”
5. Create a python module called as calculator and keep the following functions in that module:

Add – for adding numbers ( any number of arguments) and return the result

Subtract – for subtracting two numbers and return the result

Multiply – for multiplying two numbers and return the result

Divide – for dividing two numbers and return the result

Create a python script file and call the above functions in the file.

1. How do you see which are the “builtins” available?
2. How do you check the entries in your local symbol table?

**Object Oriented Concepts:**

1. Create a class called Student which has the following specifications :

A class level variable 🡪 StudentCount

Instance variables 🡪 name, age, stream

Constructor (with arguments) 🡪 for initializing the instance variables

Method displayStudent() 🡪 to display a student object printing its name, age, and stream

Method displayStudentCount() 🡪 to display the current count of students

Create 3 student objects

Call the displayStudent() method for each object

Print the total count of students

1. Create a class Rectangle which has following members.

Instance variables 🡪 length, breadth

Constructor to initialize the instance variables – length and breadth

Method calArea( ) which should return area of the rectangle.

Create 5 objects of Rectangle. Get the data from user through keyboard and set in all objects. Print the area of every object

1. Create a class Shape having following method:

draw( ) – Printing “This is generic shape”

rotate( ) -- Printing “Rotating generically”

Create a class Circle extending Shape class.

draw( ) should display message “Circle Drawn”.

rotate( ) should display message “Circle Rotated”.

Create a class Rectangle extending Shape class.

draw( ) should display message “Rectangle Drawn”.

rotate( ) should display message “Rectangle Rotated”.

Create an object of Circle and Rectangle. Call draw and rotate methods to display appropriate messages.

1. Create a class Parent with following members:

Instance variables 🡪 name, age

Constructor to initialize the instance variables – name and age

Method 🡪 parentDetails() --- printing name and age

Create a class Employee with following members:

Instance variables 🡪 eid, designation

Constructor to initialize the instance variables – eid and designation

Method 🡪 employeeDetails() --- printing eid and designation

Create a class Person which inherits from both Parent and Employee

Constructor to initialize the instance variables

Method 🡪 personDetails() 🡪 printing name, age, eid and designation

Create an object of Person class and call the personDetails(), ParentDetails() and employeeDetails() methods to see the result.

**Exception Handling:**

1. Create a class Average having following methods:

printAverage (totalSum, totalNumber ) – method to print average based on the arguments passed

computeAverage(sum, number ) – method to actually calculate and return the average

Call computeAverage method from printAverage method

Ask user to enter value of total sum and total number.

Call printAverage to calculate and print average.

Handle probable Exceptions.

1. Create a class Customer having following members:

custNo

custName

custPhone

Constructor to initialize all instance variables

Method – displayCustomer() 🡪 printing customer details

Ask user to enter customer details.

Check if custPhone is a 10 digit number.

If yes then create the Customer object and call the displayCustomer() method.

If not then raise user defined exception called InvalidPhoneException.

**File Handling:**

1. Write a program to read each line of the file one by one and display output with a line number in the beginning.
2. Create a file names.txt and store 10 different user names in that file, one user name in one line.

Now write a script that accepts user name as command line argument, checks whether a similar name exists in names.txt, if yes, it asks you to provide the Age, Salary and Phone no for that user and store all these details in a file called userdata.txt in the current directory.

1. Provide source file and destination file names as command line arguments. Perform following functionality:

Program should copy contents of source file to destination file

If source file does not exist, display appropriate error message

If destination file does not exist, it should be created.

If destination file already exist, program should ask Want to Overwrite? (yes/no). If user selects Yes then overwrite otherwise append

1. Create a class Employee having members as follows:

empNo

empName

empBasic

Constructor to initialize members

Ask user to enter details of an employee and set them in an Employee object.

Store details of this object in a file emp.txt

Read employee details from the file and display those details