**GALGOTIAS UNIVERSITY**

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

MCA 3 SEMESTER

ASSIGNMENT-1

UNIT-1

Tripti Singh

Roll No - 22SCSE2030069

QUESTION-1. Explain the concept of object oriented programming in python language.

ANSWER- Object-oriented programming (OOP) is a programming paradigm that uses objects and classes in programming. It aims to implement real-world entities like inheritance, polymorphisms, encapsulation, etc. in the programming.

In Python, everything is an object, even functions, classes, and modules. Objects are self-contained entities that have data and behaviour. The data is represented by attributes, and the behaviour is represented by methods.

Classes are blueprints for creating objects. They define the attributes and methods that objects will have.

class Person:

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

def greet(self):

print(f"Hello, my name is {self.name} and I am {self.age} years old.")

OOP is a powerful programming paradigm that can help us to write more modular, reusable, and maintainable code. It is a fundamental concept in Python, and it is important for any Python programmer to understand.

QUESTION- 2. Definition of a list. How to create a list in python programming language.

ANSWER-

A list in Python is a mutable, ordered sequence of elements. This means that you can change the elements in a list after it has been created, and that the elements in a list are stored in a specific order.

To create a list in Python, simply place the elements of the list inside square brackets, separated by commas. For example, the following code creates a list with three elements:

my\_list = ["apple", "banana", "cherry"]

QUESTION-3.What is a python web application framework and how many of web application framework to explain.

ANSWER- A Python web application framework is a collection of modules or packages that allow developers to write web applications or services. These frameworks automate the implementation of tasks and provide structure for application development. They also reduce the amount of code a developer needs to write.

Python frameworks can be full-stack, micro, or asynchronous. The most popular Python web frameworks are: Django, Flask, Falcon, Pyramid, Bottle.

Python frameworks are designed to run on HTTP servers such as Apache or Nginx. They can be used on Windows, Unix/Linux, Mac, and Google App Engine.

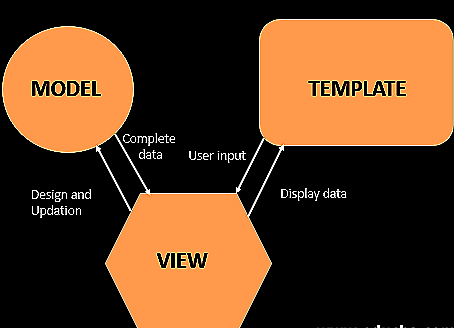
Python frameworks can help developers avoid having to handle low-level details such as protocols, sockets, or process/thread management.

QUESTION-4.Explain the model view template architecture.

ANSWER- The Model-View-Template (MVT) architecture is a software design pattern for developing web applications. The MVT architecture is used by the Django web framework. The MVT architecture is made up of three components:

* Model: The structure of storing data in a database.
* View: The user interface, or what you see in your browser when you render a website. The view is a Python function that handles web requests.
* Template: Contains static content like HTML, CSS, and JavaScript. The template handles the UI and architecture part of an application.

The MVT architecture is responsible for displaying all or a portion of data to the user.



QUESTION-5.Explain the dictionary and dictionary methods with an example.

ANSWER- A dictionary is a collection of key-value pairs. The keys are unique, and the values can be any type of data. Dictionaries are a powerful tool for storing and retrieving data.

**There are many different dictionary methods available in Python. Some of the most commonly used methods are:**

**.get()**

: Returns the value of the specified key, or None if the key does not exist.

**.items()**

: Returns a list containing a tuple for each key-value pair in the dictionary.

**. keys()**

: Returns a list containing all of the keys in the dictionary.

**. pop()**

: Removes the element with the specified key from the dictionary and returns its value.

**.update()**

: Updates the dictionary with elements from another dictionary or an iterable of key-value pairs.

FOR EXAMPLE- dictionary = {"name": "John Smith", "age": 30}  
  
 name = dictionary.get("name")  
  
 print(name)

QUESTION-6.Explain the tuple in python with an example.

ANSWER-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](https://www.w3schools.com/python/python_lists.asp), [Set](https://www.w3schools.com/python/python_sets.asp), and [Dictionary](https://www.w3schools.com/python/python_dictionaries.asp), all with different qualities and usage.

A tuple is a collection which is ordered and **unchangeable**.

Tuples are written with round brackets.

FOR EXAMPLE-

thistuple = ("apple", "banana", "cherry")

print(thistuple)

OUTPUT-('apple', 'banana', 'cherry')

QUESTION-7.What is function and how to declare a function in python programming language.

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

You can pass data, known as parameters, into a function.

A function can return data as a result

Calling a Function

To call a function, use the function name followed by parenthesis:

def my\_function():

print("Hello from a function")

my\_function()

OUTPUT-Hello from a function

QUESTION-8.Explain the following list method with an example. a) append() b) extend() c)insert() d) index() e)sort().

ANSWER- a) **append()**: The **append()** method is used to add an element to the end of a list.

Example: fruits = ["apple", "banana", "cherry"]

fruits.append("orange")

print(fruits)

OUTPUT- ['apple', 'banana', 'cherry', 'orange']

b) **extend()**: The **extend()** method is used to append the elements of an iterable (e.g., a list, tuple, or string) to the end of the list.

Example: fruits = ["apple", "banana", "cherry"]

more\_fruits = ["orange", "grape"]

fruits.extend(more\_fruits)

print(fruits)

OUTPUT- ['apple', 'banana', 'cherry', 'orange', 'grape']

c) **insert()**: The **insert()** method is used to insert an element at a specific index in the list.

Example: fruits = ["apple", "banana", "cherry"]

fruits.insert(1, "orange")

print(fruits)

OUTPUT- ['apple', 'orange', 'banana', 'cherry']

d) **index()**: The **index()** method is used to find the index of the first occurrence of a specified element in the list.

Example: fruits = ["apple", "banana", "cherry"]

index = fruits.index("banana")

print("Index of 'banana':", index)

OUTPUT- Index of 'banana': 1

e) **sort()**: The **sort()** method is used to sort the elements of the list in ascending order. You can also use the **reverse=True** argument to sort in descending order.

Example: numbers = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]

numbers.sort()

print(numbers)

OUTPUT- [1, 1, 2, 3, 3, 4, 5, 5, 5, 6, 9]