VEC ASSIGNMENT 1

1Q: Define Artificial Intelligence (AI) and provide examples of its applications.

Artificial intelligence, or AI, is technology that enables computers and machines to simulate human intelligence and problem-solving capabilities.

On its own or combined with other technologies (e.g., sensors, geolocation, robotics) AI can perform tasks that would otherwise require human intelligence or intervention.

Artificial intelligence (AI), the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings. The term is frequently applied to the project of developing systems endowed with the intellectual processes characteristic of humans, such as the ability to reason, discover meaning, generalize, or learn from past experience.

Applications of AI:

Ai in E-Commerce like personalization, fake review detection

AI in Education Purpose such as Voice Assistant, Gamification, Smart Content creation

AI in Robotics such as NLP,Object Recognition, HRI

GPS and Navigations

Healthcare such as Telehealth, Patient Monitoring, Surgical Assistance

Automobiles such as autonomous driving

Agriculture such as Stock Monitoring, Pest Management, Forecasting, Pest Management etc..

2Q: **Differentiate between supervised and unsupervised learning techniques in ML.**

Supervised Learning	Unsupervised Learning

Supervised learning techniques are trained on a labelled dataset.	*Unsupervised learning allows the model to discover patterns and relationships in unlabeled data.*
Types of Supervised learning are:	*Types of Unsupervised Learning are:*
Regression	*Clustering*
Classification	*Association*
Supervised learning has a feedback mechanism.	*Unsupervised learning has no feedback mechanism*
It contains a known number of classes	*It contains an unknown number of classes*
These techniques are used for prediction	*These techniques are used for Analysis*
	Algorithms include: k-means clustering, hierarchical clustering,

Algorithms include: decision tree, logistic regression, support vector machine *Applications: Risk Evaluation, Forecast sales*	*apriori algorithm* *Applications: Recommondation system, Anamoly detection*

3Q. **What is Python? Discuss its main features and advantages.**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics developed by Guido van Rossum. It was originally released in 1991.

FEATURES OF PYTHON:

1. Free and Open Source

Python language is freely available at the official website, since it is open-source, this means that source code is also available to the public. So you can download it, use it as well as share it.

2. Easy to code

Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in the Python language and anybody can learn Python basics in a few hours or days. It is also a developer-friendly language.

3. Easy to Read

Python's syntax is really straightforward. The code block is defined by the indentations rather than by semicolons or brackets.

4. Object-Oriented Language

One of the key features of Python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, object encapsulation, etc.

5. GUI Programming Support

Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in Python.

6. High-Level Language

Python is a high-level language. When we write programs in Python, we do not need to remember the system architecture, nor do we need to manage the memory.

7. Easy to Debug

Excellent information for mistake tracing. You will be able to quickly identify and correct the majority of your program's issues once you understand how to interpret Python's error traces.

8. Python is a Portable language

Python language is also a portable language. For example, if we have Python code for Windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

9. Python is an Integrated language

Python is also an Integrated language because we can easily integrate Python with other languages like C, C++, etc.

10. Interpreted Language:

Python is an Interpreted Language because Python code is executed line by line at a time. like other languages C, C++, Java, etc. there is no need to compile Python code this makes it easier to debug our code.

11. Large Standard Library

Python has a large standard library that provides a rich set of modules and functions so you do not have to write your own code for every single thing.

ADVANTAGES OF PYTHON :

Easy to Learn and Use. For Beginners, Python is simple to understand and use….

Free and Open-Source. ...

Rapid Development. ...

Interpreted Language. ...

Wide Range of Libraries and Frameworks. ...

Dynamically Typed. ...

Portability. ...

Strong Community Support…

Cross-Platform Compatibility

Strong Community Support

Integration and Extensibility

Scalability and Performance.

4Q. ***What are the advantages of using Python as a programming language for AI and ML?***

The AI and machine learning projects vary significantly. The significant difference is because of the stack technology. There is a demand for deep research in every step. Python AI projects are taking over the world because of their flexibility.

According to Deloitte research, AI companies use technical information to boost productivity. Therefore, there is no doubt that AI can help ease a wide range of tasks, and it helps to meet the strategic goals that play an important role in improving efficiency.

ADVANTAGES OF USING PYTHON FOR AI AND ML:

A great library ecosystem

A low entry barrier

Flexibility

Platform independence

Readability

Good visualization options

Community support

Growing popularity

5Q. ***Discuss the importance of indentation in Python code.***

In the realm of Python programming, indentation is not merely a matter of style but a fundamental aspect of the language's syntax. Python relies on indentation to define the structure of code blocks, making it unique among programming languages. Unlike languages that use explicit symbols or braces to denote code blocks, Python uses whitespace, specifically indentation, to determine the scope of statements.

In Python, indentation refers to the spacing at the beginning of a line of code that determines its grouping and hierarchy within the program's structure. Unlike many programming languages that use braces ({}) or other explicit symbols to denote code blocks, Python uses indentation to signify the beginning and end of blocks of code.
The primary purpose of indentation in Python is to define the scope of statements, such as those within loops, conditionals, functions, and classes. Consistent and proper indentation is crucial for the interpreter to understand the logical structure of the code. Indentation is not just a matter of style or convention in python.

6Q. Define a variable in Python. Provide examples of valid variable names.

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume). Rules for Python variables:

A variable name must start with a letter or the underscore character

A variable name cannot start with a number

A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)

Variable names are case-sensitive (age, Age and AGE are three different variables)

A variable name cannot be any of the Python keywords.

.

Examples of Valid Variable Names in Python:

Valid Variable Names:

Variable names must start with a letter or an underscore (_).

They can only contain letters, numbers, and underscores.

Variable names cannot contain spaces or special characters.

Variables are case-sensitive.

Examples:

Valid:

user_age

website

password1

answer

weight

message

is_authenticated

fruits

point

Naming Conventions:

Variable names should be descriptive, not too short or too long.

Use lowercase letters and underscores to separate words (snake_case).

Follow PEP8 naming conventions for consistency.

In Python, variables play a crucial role in storing and manipulating data. By following the naming rules and conventions, developers can create meaningful and readable variable names that enhance code clarity and maintainability.

7Q. *Explain the difference between a keyword and an identifier in Python*.

S.NO	KEYWORD	IDENTIFIER
1	A keyword refers to a predefined word that python reserves for working programs that have a specific meaning, You can’t use a keyword anywhere else.	Python Identifiers are the different values that a programmer can use to define various variables, integers, functions, and classes.
2	A keyword can specify the type of entity.	An identifier can identify a single entity (a variable, a class, or a function).
3	All the keywords except ‘True’, ‘False’, and ‘None’ start in lowercase letters.	The first character can be a lowercase letter or an uppercase letter. However, an identifier can’t start with a digit.
4	Keywords are generally in lower case.	A variable can be in uppercase or lowercase letters.
5	Python Keywords comprise alphabetical characters.	An identifier can comprise alphabets, numbers, and underscore.
6	There is no use of special characters.	No special character is used except underscore (‘_’).

7	*A few examples of Python keywords are: True, False, else, import, finally, is, and global*	*A few examples of identifiers are testing, sq4 sides, area_square, etc.*

8Q. List the basic data types available in Python.

Python Data types are the classification or categorization of data items. It represents the kind of value that tells what operations can be performed on a particular data. Since everything is an object in Python programming, Python data types are classes and variables are instances (objects) of these classes. The following are the standard or built-in data types in Python:

Numeric

Sequence Type

Boolean

Set

Dictionary

Binary Types

1. Numeric Data Types in Python

The numeric data type in Python represents the data that has a numeric value. A numeric value can be an integer, a floating number, or even a complex number. These values are defined as Python int, Python float, and Python complex classes in Python.

2. Sequence Data Types in Python

The sequence Data Type in Python is the ordered collection of similar or different Python data types. Sequences allow storing of multiple values in an organized and efficient fashion. There are several sequence data types of Python:

Python String

Python List

Python Tuple

3. Boolean Data Type in Python

Python Data type with one of the two built-in values, True or False. Boolean objects that are equal to True are truthy (true), and those equal to False are falsy (false)

4. Set Data Type in Python

In Python Data Types, a Set is an unordered collection of data types that is iterable, mutable, and has no duplicate elements. The order of elements in a set is undefined though it may consist of various elements.

5. Dictionary Data Type in Python

A dictionary in Python is an unordered collection of data values, used to store data values like a map, unlike other Python Data Types that hold only a single value as an element, a Dictionary holds a key: value pair. Key-value is provided in the dictionary to make it more optimized. Each key-value pair in a Dictionary is separated by a colon : , whereas each key is separated by a ‘comma’.

9Q. Describe the syntax for an if statement in Python

if Statement

In Python, if statements are a starting point to implement a condition. Let’s look at the simplest example:

```
If <condition> :

    <expression>
```

When <condition> is evaluated by Python, it’ll become
either True or False (Booleans). Thus, if the condition is True (i.e, it is met),
the <expression> will be executed, but if <condition> is False (i.e., it is not met),
the <expression> won’t be executed.

We are pretty free to decide what conditions and expressions can be because Python is very flexible.

Let’s look at a concrete example.

```
X=3;

Y=10;

if x<y:

   print(“ x is smaller than y”);
```

output:

```
x is smaller than y.
```

First of all, we define two variables, x and y. Then we say that if variable x is smaller than variable y, print out x is smaller than y). Indeed, if we execute this code, we’ll print out this output because 3 is smaller than 10.

Output: x is smaller than y.

10Q. Explain the purpose of the elif statement in Python

The “elif” keyword in Python, stands for “else if”. It can be used in conditional statements to check for multiple conditions. For example, if the first condition is false, it moves on to the next “elif” statement to check if that condition is true. If none of the conditions are true, the code will execute the else statement.

Once understood, the "elif" keyword can become a powerful tool in Python that lets kids begin to write more complex conditional statements. It can be used to check multiple conditions and execute different blocks of code depending on the outcome. The "elif" statement is often used in conjunction with the if statement to create more complex logic.

Examples of using "elif" in Python

Elif Syntax

"Elif" stands for "else if" in Python. Here's a very simple example of where is can be used to check multiple conditions and execute a block of code if the conditions are true. The syntax is as follows:

```
if condition1:

    # execute code if condition1 is true

elif condition2:

    # execute code if condition2 is true

else:

    # execute code if all conditions are false
```

Here's another simple example where "elif" is used in Python.

```
x = 10
```

```
if x > 10:
    print("x is greater than 10")
elif x < 10:
    print("x is less than 10")
else:
    print("x is equal to 10")
```

In the example above, Python checks the value of the variable x against multiple conditions. If x is greater than 10, it prints “x is greater than 10”. If x is less than 10, it prints “x is less than 10”. Otherwise, if neither of the previous conditions is met, it prints “x is equal to 10”.