ASSIGNMENT-1

NAME : NIKHILSAI
ROLL NUMBER : 21UK1A6794
MAIL:nikhilsai@0803gmail.com
BRANCH:CSE(DS)

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

- It is a technology that allows machines to simulate human intelligence and problem-solving capabilities.
- AI encompasses various subfields,including machine learning, natural language processing,computer vision,robotics,and expert system.

  # EXAMPLE OF AI APPLICATION:
  - -> Customer Service
  - -> Energy Management
  - -> Supply Chain Management
  - -> Agriculture
  - -> Content Creation.

2.Differentiate between supervised and unsupervised learning techniques in ML.

- Supervised learning:

  It Uses labeled training data to train algorithms to predict outcomes and recognize patterns.

  Classification and regression are common tasks in supervised learning. In classification, the model predicts discrete class labels (e.g., spam or not spam, cat or dog), while in regression, the model predicts continuous numerical values (e.g., house prices, temperature).

- Unsupervised learning:

  It does not use labeled training data. Unsupervised learning algorithms work independently to learn the data's inherent patterns, differences, similarities, and structure.

  Clustering and dimensionality reduction are common tasks in unsupervised learning. In clustering, the goal is to group similar data points into clusters based on their feature similarities .

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

- Python is high level language which is easy to learn and use.

- Python is also an object-oriented language, which means that it supports the use of objects and classes.

  # MAIN FEATURES AND ADVANTAGES OF PYTHON
     -> Standard library
     -> Dynamic language
-> Obeject Oriented
-> Portable
-> Open source

4.What are the advantages of using Python as a programming language for AI and ML?
     -> Scalability and Performance
     -> Integration Capabilities
     -> Community Support and Documentation
     -> Ease of Prototyping and Experimentation
     -> Vast Ecosystem of Libraries and Frameworks.

5.Discuss the importance of indentation in Python code.
- Readability: Indentation enhances the readability of Python code by visually representing the hierarchical structure of the program.
- Enforcement of Syntax: In Python, indentation is not just a matter of style,it is enforced by the interpreter as part of the language syntax.
- Code Blocks: Indentation delineates code blocks such as loops, conditional statements, function definitions, and class definitions in Python.
- Consistency and Conventions: Python's official style guide, PEP 8, recommends using four spaces for each level of indentation, promoting consistency and uniformity across Python codebases.

6.Define a variable in Python. Provide examples of valid variable names.
i. A variable is a named reference to a value stored in memory.
ii. variables are used to store the data that can be manipulated,accessesed and refrenced throuthout a program
iii. When defining a variable in Python, you assign a value to a name using the assignment operator (=).

     # EXAMPLES OF VALID VARIABLES:
SYNTAX FOR VARIABLE
-> variable_name = value
# EXAMPLE
x = 5
name = "John"

```
age = 30
is_student = True
my_list = [1, 2, 3]
```

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

- In Python, keywords and identifiers are both fundamental concepts, but they serve different purposes in the language:

<u>Keywords:</u>

- Keywords, also known as reserved words, are predefined words that have special meanings and purposes in Python.
- These words are reserved by the Python language and cannot be used as identifiers (variable names, function names, etc.)
- Examples of Python keywords include if, else, for, while, def, class, return, try, except, True, False, None, etc.

<u>Identifiers:</u>

- Identifiers are names given to various programming elements in Python, such as variables, functions, classes, modules, and objects.
- Identifiers must adhere to certain rules and conventions:
- They can contain letters (both uppercase and lowercase), digits, and underscores.
- They cannot begin with a digit. They cannot be a keyword.
- They should follow Python's naming conventions, such as using lowercase letters for variable and function names (snake_case) and using CamelCase for class names.
- Examples of identifiers in Python include variable names (x, name, age), function names (calculate_sum, print_message).

8.List the basic data types available in Python.

- In Python, there are several basic data types:
- 1. Integers (`int`): Whole numbers, like 5, -3, 1000, etc.
- 2. Floating-point numbers (`float`): Decimal numbers, like 3.14, -0.001, 2.71828, etc.
- 3. Complex numbers (`complex`): Numbers with a real and imaginary part, written as `a + bj`, where `a` is the real part and `b` is the imaginary part.
- 4. Strings (`str`): Ordered sequence of characters, like "hello", 'python', "123", etc. They are immutable.
- 5. Booleans (`bool`): Represents True or False.

- 6. Lists (`list`): Ordered and mutable collections of items, enclosed in square brackets, like [1, 2, 3], ['apple', 'banana', 'orange'], etc.
- 7. Tuples (`tuple`): Ordered and immutable collections of items, enclosed in parentheses, like (1, 2, 3), ('a', 'b', 'c'), etc.
- 8. Sets (`set`): Unordered collections of unique items, enclosed in curly braces, like {1, 2, 3}, {'apple', 'banana', 'orange'}, etc.
- 9. Dictionaries (`dict`): Unordered collections of key-value pairs, enclosed in curly braces, like {'name': 'Alice', 'age': 30}, {'a': 1, 'b': 2, 'c': 3}, etc.

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

- In Python, an if statement is used to conditionally execute a block of code based on whether a certain condition is true.

  ```
  # Syntax for an if statement:
  if condition:
      # code block to execute if condition is True
  ```

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

- The elif statement in Python stands for "else if". It's used to check additional conditions after an initial if statement.

  ```
  # Syntax for an elif statement:
  if condition1:
      # code block to execute if condition1 is True
  elif condition2:
      # code block to execute if condition2 is True
  else:
      # code block to execute if all conditions are False
  ```

  Example for elif statement:

  ```
  x = 5
  if x > 10:
      print("x is greater than 10")
  elif x > 5:
      print("x is greater than 5 but not greater than 10")
  else:
      print("x is 5 or less")
  ```