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

Ans: **Artificial Intelligence (AI)** is a branch of computer science that aims to create systems capable of performing tasks that typically require human intelligence. These tasks include reasoning, learning, problem-solving, perception, understanding natural language, and more. Examples of AI applications include:

**. Natural Language Processing (NLP)**

**. Computer Vision**

**. Machine Learning (ML)**

1. Differentiate between supervised and unsupervised learning techniques in ML

Ans:

* **Supervised Learning:** In supervised learning, the algorithm learns from labeled data, where each input is paired with the correct output. The algorithm aims to learn a mapping from inputs to outputs, making predictions or decisions when new data is encountered. Examples include classification and regression tasks.
* **Unsupervised Learning:** In unsupervised learning, the algorithm learns from unlabeled data without explicit guidance. The goal is to discover hidden patterns, structures, or relationships within the data. Examples include clustering, dimensionality reduction, and association rule learning.

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

Ans:

**Python** is a high-level, interpreted programming language now for its simplicity, readability, and versatility. Its main features and advantages include:

* + **Readability:** Python uses simple and intuitive syntax, making it easy to write and understand code, even for beginners.
  + **Versatility:** Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming, making it suitable for various applications.
  + **Large Standard Library:** Python comes with a vast collection of modules and libraries for tasks ranging from web development and data analysis to machine learning and artificial intelligence.
  + **Dynamic Typing:** Python dynamically determines the data type of variables at runtime, simplifying development and allowing for faster prototyping.
  + **Platform Independence:** Python code can run on various operating systems without modification, enhancing its portability and interoperability.

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

Ans: Python's advantages for AI and ML stem from its simplicity, readability, and extensive libraries such as TensorFlow and scikit-learn. It offers ease of prototyping, scalability, and interoperability with other languages and tools.

1. Discuss the importance of indentation in Python code.

Ans: **Importance of Indentation in Python Code:**

In Python, indentation is crucial for defining the structure and scope of code blocks, such as loops, conditional statements, and function definitions. Unlike other programming languages that use curly braces or keywords to denote code blocks, Python relies on indentation for readability and clarity. Consistent indentation ensures that code is organized logically and makes it easier to understand, debug, and maintain

1. Define a variable in Python. Provide examples of valid variable names.

Ans: **Defining a Variable in Python:**

In Python, a variable is a named reference to a value stored in memory. Variables are created using the assignment operator "=" and can hold different types of data, such as numbers, strings, lists, or objects. Examples of valid variable names in Python include:

x = 10 # Integer variable

name = "John" # String variable

is\_valid = True # Boolean variable

my\_list = [1, 2, 3] # List variable

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

Ans: **Difference between a Keyword and an Identifier in Python:**

. **Keyword:** Keywords are reserved words in Python that have predefined meanings and cannot be used as identifiers (variable names, function names, etc.). Examples of keywords include "if", "else", "for", "while", "def", "class", etc.

**. Identifier:** Identifiers are names given to entities in Python, such as variables, functions, classes, modules, etc. Identifiers must follow certain rules, such as starting with a letter or underscore, and can contain letters, digits, and underscores. Identifiers cannot be keywords or contain special characters (except underscores).

1. List the basic data types available in Python.
2. Ans: **Basic Data Types available in Python:**

Python supports several built-in data types, including:

* + **Integer (int):** Represents whole numbers without any fractional part.
  + **Float (float):** Represents numbers with a fractional part.
  + **Boolean (bool):** Represents truth values, either True or False.
  + **String (str):** Represents sequences of characters enclosed in single, double, or triple quotes.
  + **List:** Represents ordered collections of items, which can be of different data types and mutable.
  + **Tuple:** Represents ordered collections of items, similar to lists but immutable.
  + **Dictionary (dict):** Represents key-value pairs enclosed in curly braces, allowing for fast lookups.

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

Ans: The syntax for an if statement in Python is:

if condition:

# code block to execute if condition is true

* **if**: This keyword starts the if statement.
* **condition**: This is an expression that evaluates to either True or False. If the condition is True, the code block following the if statement is executed.
* **‘:**:’ : A colon marks the end of the if statement's condition and the beginning of the indented code block.
* Indented code block: This block contains the code that is executed if the condition is true. It can consist of one or more statements. The indentation level determines which statements are part of this block.

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

Ans: **Purpose of the elseif statement in Python:**

The elseif statement (short for "else if") is used in conjunction with the if statement to specify alternative conditions to be checked if the preceding if condition is False. It allows for more complex conditional branching in Python programs. The syntax is as follows:

if condition1:

# Code block executed if condition1 is True

# Statements here

elseif condition2:

# Code block executed if condition2 is True and condition1 is False

# Statements here

else:

# Code block executed if all preceding conditions are False

# Statements here

If condition1 is True, its corresponding code block is executed. If condition1 is False but condition2 is True, the code block under elseif condition2 is executed. If all conditions are False, the code block under else is executed.