***Course Code: - OEC CSM701A***
***Course Name: - AI with Robotics***

**Question 1: Describe basics of Robot Programming Language. (CO5) [2 Marks]**

**Answer:**

A **Robot Programming Language (RPL)** is a specialized computer language developed to control and automate robotic systems. It enables users to instruct robots on how to perform specific tasks such as movement, manipulation, sensing, and communication with other devices or systems.

**1. Key Characteristics:**

- **Motion Commands:**
  Used to define movements of the robot's joints or end-effectors, such as *MOVE*, *ROTATE*, *GRIP*, etc.
  Example:
  MOVEJ P1 – moves the robot to position P1 using joint motion.
- **Task Sequencing:**
  Allows defining the sequence of actions a robot should follow to perform a job efficiently.
- **Sensor Integration:**
  RPLs can read and interpret sensor data (like proximity, force, or vision sensors) to make dynamic decisions.
- **Control Structures:**
  Like other programming languages, they use **loops, conditionals, and functions** to allow logical control and decision-making.
- **Communication and Synchronization:**
  Robots can exchange data with other systems (like PLCs or other robots) for coordinated tasks.

**2. Levels of Robot Programming:**

- **Teach Programming:** Manually guiding the robot to positions (e.g., using a teach pendant).
- **Offline Programming:** Writing programs on a computer and uploading them to the robot.
- **High-Level Programming:** Using task-oriented languages integrated with AI and machine vision.

**3. Examples of RPLs:**

- **RAPID** – used by ABB Robots.
- **KRL (KUKA Robot Language)** – used for industrial automation.
- **VAL** – one of the first robot languages developed by Unimation.
- **AML (A Manufacturing Language)** – used by IBM for flexible automation.

**4. Applications:**

Used in **welding, painting, assembly lines, packaging, and inspection** where precision and repeatability are required.

---

**Question 2: Justify that CNN is used for Image-Related Analysis. (CO4) [3 Marks]**

**Answer:**

A **Convolutional Neural Network (CNN)** is a deep learning model specifically designed for analyzing and interpreting visual data such as images and videos. CNNs mimic the human visual system by automatically learning hierarchical patterns and spatial features from raw pixel data.

**1. Architecture and Functionality:**

CNNs consist of multiple layers that process an image step by step:

- **Convolutional Layers:**
  Apply filters (kernels) that detect features like edges, corners, colors, or textures.
- **Activation Function (ReLU):**
  Introduces non-linearity to model complex relationships.
- **Pooling Layers:**
  Reduce the spatial dimensions, making the model more efficient and translation-invariant.
- **Fully Connected Layers:**
  Combine extracted features to make final predictions or classifications.

**2. Why CNNs are Suitable for Image Analysis:**

- **Automatic Feature Extraction:**
  Unlike traditional methods, CNNs do not require manual feature engineering — they learn relevant visual features automatically.
- **Spatial Hierarchies:**
  Early layers learn simple patterns, while deeper layers learn complex features like shapes or objects.
- **Invariance to Distortion and Position:**
  Pooling operations allow CNNs to recognize objects even when they are scaled, rotated, or partially occluded.
- **Parameter Sharing:**
  The same filter is used across the image, reducing the number of parameters and improving generalization.

**3. Common Applications:**

- **Image Classification:** e.g., recognizing objects in photos (dogs, cars, faces).
- **Object Detection:** identifying and locating multiple objects within an image (used in autonomous vehicles).
- **Facial Recognition:** used in security and social media applications.
- **Medical Image Analysis:** detecting tumors or abnormalities in scans.

**4. Example:**

In facial recognition, CNNs can automatically extract unique facial features such as eye distance, shape, and contours to identify individuals accurately — a task that traditional algorithms struggle with.

***Course Code: - PCC-CSM701***
***Course Name: - Software Engineering***

***Question 1: Describe the Top-Down Approach to Integration Testing. [2 Marks]***

**Answer:**

The **Top-Down Approach** to integration testing is a method where testing begins from the **top-level modules** of the software hierarchy and progressively moves downward to the lower-level modules.

**Key Features:**

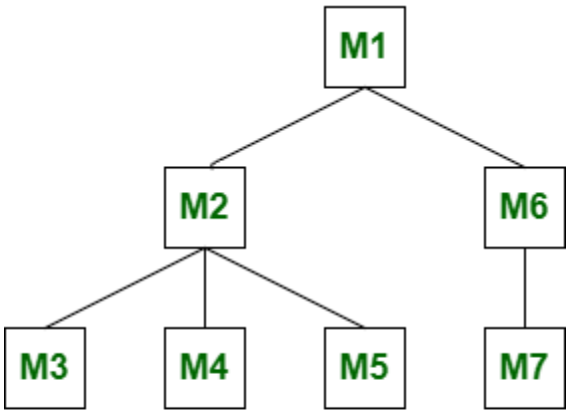
- **Start from Main Control Module:** Testing begins with the main (top) module, which is integrated and tested first.
- **Use of Stubs:** Since lower-level modules may not yet be developed, temporary programs called **stubs** are used to simulate their behavior.
- **Progressive Integration:** As lower modules are developed, stubs are replaced with the actual components, and testing continues.

**Advantages:**

- Early detection of design and control flow errors.
- Major functionalities are tested early in development.

**Example of Top Down Integration Testing :**

In the top-down integration testing, if the depth-first approach is adopted then we will start integration from module M1. Then we will integrate M2, then M3, M4, M5, M6, and at last M7.




Program Structure

***Question 2: Explain the purpose and components of a Class Diagram.***

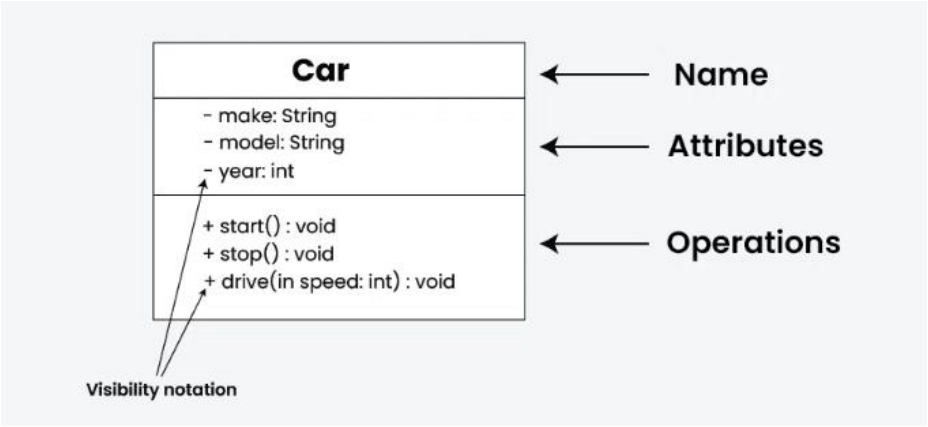
**Answer:**

A **Class Diagram** is one of the most important **UML (Unified Modeling Language)** diagrams used in **Object-Oriented Design (OOD)**. It visually represents the **static structure** of a system by showing **classes**, their **attributes**, **methods (operations)**, and the **relationships** among them.

**1. Purpose of a Class Diagram:**

- **System Blueprint:**
  It provides a **blueprint of the system architecture**, showing how different classes interact.
- **Understanding Relationships:**
  It helps developers understand **inheritance, associations, and dependencies** among objects.
- **Documentation:**
  Acts as a **documentation tool** for explaining the design to team members and stakeholders.
- **Basis for Code Generation:**
  Many software design tools can automatically generate source code from class diagrams.
- **Helps in Analysis and Design:**
  Used during **object-oriented analysis** and **system design** phases to model real-world entities.

**2. Main Components of a Class Diagram:**




***Class Name:*** The name of the class is typically written in the top compartment of the class box and is centered and bold.

***Attributes:*** Attributes, also known as properties or fields, represent the data members of the class. They are listed in the second compartment of the class box and often include the visibility (e.g., public, private) and the data type of each attribute.

***Methods:*** Methods, also known as functions or operations, represent the behavior or functionality of the class. They are listed in the third compartment of the class box and include the visibility (e.g., public, private), return type, and parameters of each method.

***Visibility Notation:*** Visibility notations indicate the access level of attributes and methods. Common visibility notations include:

- \+ for public (visible to all classes)
- \- for private (visible only within the class)
- # for protected (visible to subclasses)
- ~ for package or default visibility (visible to classes in the same package)

**4. Advantages of Using Class Diagrams:**

- Helps visualize **object-oriented structure**.
- Supports **better communication** among developers.
- Aids in **code implementation and system maintenance**.
- Provides a **clear model** for system analysis and design.

***Course Code: - OEC-CSM702C***
***Course Name: - Cyber Law and Ethics***

***Q.1.A : What are Certifying Authorities (CAs)? Explain their role and regulation under the IT Act. [1 ]***

**Answer:**

**Certifying Authorities (CAs)** are trusted entities licensed by the **Controller of Certifying Authorities (CCA)** under the **Information Technology Act, 2000** to issue **Digital Signature Certificates (DSCs)**.

**Role and Regulation:**

- CAs verify the identity of individuals or organizations before issuing DSCs used for **secure electronic communication, authentication, and digital transactions**.
- They operate under the supervision of the **Controller of Certifying Authorities (CCA)** as defined in **Sections 17–24 of the IT Act, 2000**.
- The CCA ensures that CAs maintain **security standards, data integrity, and confidentiality** in digital certification processes.
- Example: eMudhra, Sify, and NIC are licensed Certifying Authorities in India.

---

***Q.1.B : Identify the duties of the Controller of Certifying Authorities (CCA). [1]***

**Answer:**

The **Controller of Certifying Authorities (CCA)** is the apex regulatory body that oversees and licenses all CAs in India under the IT Act.

**Major Duties include:**

1. **Licensing and Supervision:** Grant, renew, and revoke licenses of Certifying Authorities.
2. **Standards Enforcement:** Specify standards for digital signature creation and verification.
3. **Compliance Monitoring:** Ensure CAs follow prescribed security procedures.
4. **Repository Maintenance:** Maintain a **national repository** of issued digital certificates.
5. **Public Key Infrastructure (PKI) Management:** Promote trust and legal validity in electronic communication.

---

***Q.2.A : Evaluate the importance of jurisdiction in handling cybercrime. [1]***

**Answer:**

**Jurisdiction** refers to the **legal authority** of a court or law enforcement agency to investigate, prosecute, and adjudicate a crime.

In cybercrime, **jurisdiction is crucial** because offences often cross **national and international boundaries**. Determining where the crime occurred (origin, server location, or victim's location) is essential for:

- Ensuring **proper investigation and prosecution**,
- Avoiding **conflicts between countries' laws**, and
- Enabling **international cooperation** through treaties like the **Budapest Convention on Cybercrime**.

---

***Q.2.B : Analyze the pros and cons of regulating online content through government intervention. [2]***

**Answer:**

**Pros (Advantages):**

1. **Protection from Harmful Content:** Helps prevent the spread of hate speech, misinformation, terrorism, and child exploitation online.
2. **National Security:** Enables monitoring of online activities that threaten public order or security.
3. **Accountability:** Ensures platforms are responsible for the content they host.

**Cons (Disadvantages):**

1. **Risk to Freedom of Speech:** Excessive regulation may lead to censorship or suppression of dissent.
2. **Political Misuse:** Governments may misuse laws to silence critics or control narratives.
3. **Global Disparity:** Different countries have different standards, leading to inconsistencies in enforcement.

**Conclusion:**

While **government regulation** of online content can enhance **safety and accountability**, it must be **balanced with the protection of individual rights and freedom of expression** to maintain democratic values.

***Course Code: - PEC-CSM701A***
***Course Name: - DEEP LEARNING***

***Question 1: Analyze the concept of Lemmatization. [2 Marks]***

**Answer:**

**Lemmatization** is a **text preprocessing technique** in **Natural Language Processing (NLP)** that reduces words to their **base or root form (lemma)** while considering the **context and grammatical meaning** of the word.

***1. Concept:***

- Lemmatization uses a **vocabulary and morphological analysis** of words to return the dictionary form.
- Unlike **stemming**, which simply chops off word endings, lemmatization ensures that the resulting word is a **valid and meaningful word** in the language.
- It takes into account the **part of speech (POS)** — for example, whether a word is used as a noun or verb.

***2. Example:***

| Word | Lemmatized Form |
|---|---|
| running | run |
| studies | study |
| better | good |

Here, "running" is converted to "run" and "studies" to "study," based on grammar rules and meaning.

***3. Importance:***

- Helps in **text normalization** for machine learning and NLP tasks.
- Improves **accuracy in search engines, text mining, and sentiment analysis** by treating similar words as one.

***4. Tools Used:***

- Popular libraries like **NLTK (WordNetLemmatizer)** and **spaCy** provide built-in lemmatization functions.

***Question 2: Estimate the output after applying unshared convolution on the input digital image [3]***

**Estimate the output after applying unshared convolution on the input digital image** $\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}$

**using the following filters:**

**Filter1:** $\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$

**Filter2:** $\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$

**Filter3:** $\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}$

**Filter4:** $\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$

***ANSWER:***

Input (3x3): $\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}$

Filter1: $\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$

Filter2: $\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$

Filter3: $\begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}$

Filter4: $\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$

Convolution Process (with no stride and no padding):

$$\begin{bmatrix}1 & 2\\4 & 5\end{bmatrix} * \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$$

$\Rightarrow (1 * 1) + (2 * 0) + (4 * 0) + (5 * 1) = 1 + 0 + 0 + 5 = 6$

$$\begin{bmatrix}2 & 3\\5 & 6\end{bmatrix} * \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$$

$\Rightarrow (2 * 0) + (3 * 1) + (5 * 1) + (6 * 0) = 0 + 3 + 5 + 0 = 8$

$$\begin{bmatrix}4 & 5\\7 & 8\end{bmatrix} * \begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}$$

$\Rightarrow (4 * 1) + (5 * 1) + (7 * 1) + (8 * 1) = 4 + 5 + 7 + 8 = 24$

$$\begin{bmatrix}5 & 6\\8 & 9\end{bmatrix} * \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$$

$\Rightarrow (5 * 1) + (6 * 0) + (8 * 0) + (9 * 1) = 5 + 0 + 0 + 9 = 14$

Final Output:

$$\begin{bmatrix}6 & 8\\24 & 14\end{bmatrix}$$

***Course Code: - PEC-CSM701A***
***Course Name: - DATA VISUALISATION***

***Question 1: Classify the Planar Graph [2]***

***Answer:***

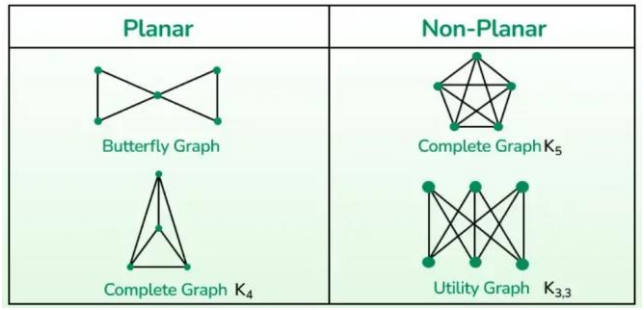
A planar graph is a graph that can be embedded in the plane such that no edges intersect except at their endpoints. In other words, it can be drawn on a flat surface without any edges crossing.

**Properties of Planar Graphs**

- **Euler's Formula**: For any **connected planar graph**, the relationship between the number of:
  - **Vertices** (V)
  - **Edges** (E)
  - **Faces** (F) is given by : V − E + F = 2
- **Kuratowski's Theorem** : A finite graph is planar (i.e., it can be drawn on a plane without any edges crossing) if and only if it does not contain a subgraph that is a subdivision of either:
  - K5 : The complete graph on 5 vertices, or
  - K3,3: The complete bipartite graph with partitions of 3 vertices each (also called the utility graph).

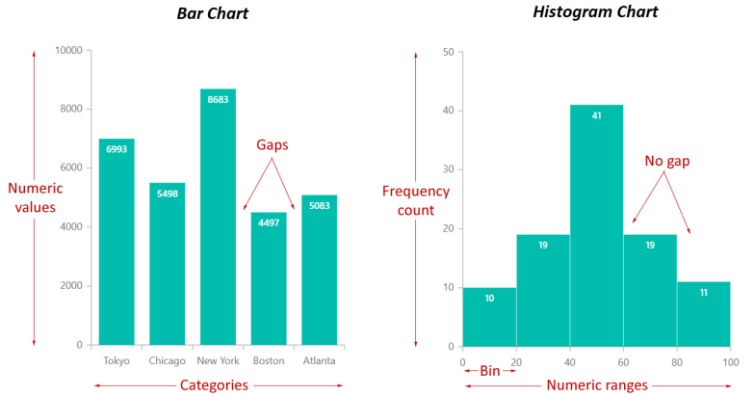
***Planar Graphs***

A planar graph is a graph that can be embedded in the plane such that no edges intersect except at their endpoints. In other words, it can be drawn on a flat surface without any edges crossing.




***Question 2: Histogram vs Bar Chart [3]***

| *Basis of Comparison* | *Histogram* | *Bar Chart* |
|---|---|---|
| *Type of Data* | *Represents continuous quantitative data divided into intervals (bins).* | *Represents categorical or discrete data.* |
| *X-axis Representation* | *Shows class intervals or ranges (e.g., 0–10, 10–20).* | *Shows distinct categories or groups (e.g., Apple, Mango).* |
| *Gaps Between Bars* | *No gaps between bars (data is continuous).* | *Gaps are present between bars (data is discrete).* |
| *Width of Bars* | *Width indicates the size of the interval.* | *Width is uniform and has no specific meaning.* |
| *Representation of Frequency* | *Area of each bar shows frequency.* | *Height of each bar shows frequency or value.* |
| *Purpose* | *Used to show frequency distribution of continuous data.* | *Used to compare values across different categories.* |


Bar Chart
Numeric values
10000
8000
6000
4000
2000
0
6993
5498
8683
4497
5083
Gaps
Tokyo
Chicago
New York
Boston
Atlanta
Categories
Histogram Chart
Frequency count
50
40
30
20
10
0
10
19
41
19
11
No gap
0
20
40
60
80
100
Bin
Numeric ranges