

**TASK 01:**

1. Java programming is a crucial tool for communication between devices over a network. It provides libraries and APIs for network programming, enabling developers to create applications that interact with remote resources, access web services, and collaborate seamlessly. Java networking enhances applications' capabilities by facilitating data exchange, remote method invocation, file transfer, and real-time communication. It also enables the development of distributed systems, enabling efficient handling of large-scale operations. Java networking also facilitates real-time communication applications like chat, video conferencing, and online gaming.
2. TCP is a connection-oriented network protocol that ensures reliable, ordered, and error-checked data packet delivery over a network. It guarantees data integrity and reliability, making it suitable for applications like file transfer, email, and web browsing. TCP is commonly used alongside HTTP, FTP, and SMTP.

UDP is a connectionless protocol that delivers data packets fast, unreliable, and unordered over a network. It offers low overhead and minimal latency, making it suitable for real-time applications like online gaming, video streaming, and VoIP. It can tolerate some packet loss.

1. Networking terminology includes IP addresses, port numbers, MAC addresses, protocols, and sockets. IP addresses are unique numerical identifiers assigned to devices, allowing communication between them. Port numbers are 16-bit integers used to identify specific processes or services on a device, allowing multiple applications to communicate over the same IP address. MAC addresses are unique identifiers assigned to network interface controllers (NICs) for physical network communication. Protocols are sets of rules and conventions governing communication between devices, defining format, sequencing, error checking, and data transmission control. A socket is an endpoint for communication between two devices over a network, consisting of an IP address and a port number, allowing data to be sent and received between client and server applications.

**Tast 02:**

1. Reference: in interlij
2. .

Socket: Represents an endpoint for communication between two devices over a network.

ServerSocket: Listens for incoming connection requests from clients and creates a new Socket for each client connection.

InetAddress: Represents an IP address and provides methods for resolving hostnames and retrieving IP addresses.

URL: Represents a Uniform Resource Locator (URL) and provides methods for accessing resources over the network.

URLConnection: Represents a connection to a URL resource and provides methods for reading from and writing to the resource.

1. Reference: in interlij

Task 03:

1. Java Socket Programming is a network-based mechanism that facilitates communication between processes on different devices, enabling seamless communication across distributed systems. It is crucial in building networked applications, including client-server architectures and distributed computing environments. In online multiplayer gaming, it ensures low-latency communication between players and the game server, ensuring smooth gameplay experiences.
2. Connection-Oriented Socket Programming (TCP) is a connection-oriented protocol that establishes a reliable and ordered communication channel between devices, ensuring data integrity and reliability. It is suitable for applications like file transfer, email communication, and web browsing. On the other hand, Connection-Less Socket Programming (UDP) is a connection-less protocol that provides fast, unreliable, and unordered data packet delivery, favored for real-time communication and speed, such as online gaming, video streaming, and VoIP.
3. Reference: in interlij