"Dissemination of Education for Knowledge, Science and Culture"
-Shikshanmaharshi Dr. Bapuji Salunkhe

Vivekanand College, Kolhapur
(An Empowered Autonomous Institute)

Department of Computer Science

2024-25

Class Assignments

Assignment No. 1

Discuss contiguous memory allocation in Detail.

- Contiguous memory allocation -

1) Contiguous memory allocation is one of the methods of memory allocation in which when a process request for the memory a single contiguous section of memory blocks is asigned to the process according to its requirement.

2) All the available memory space remains together in place. It means freely available memory particitions are not expanded here and there across the whole memory space.

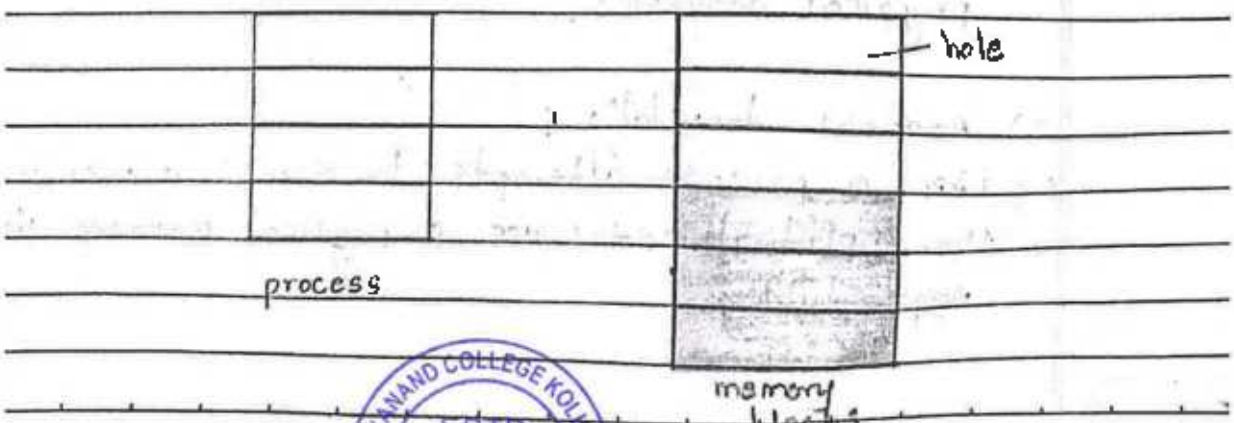
3) Both the operating system and user must reside in the main memory. The main memory is divided into two partions.

4) One portion is for os and other is for the user program

5) When any user process requests for the memory a single section of the contiguous memory blocks is given to that process according to it's need.

6) We can achive contiguous memory allocation dividing memory into the fixed size particions.

7) The free box of memory are known as holes. The set of holes is search to determine which hole is best to allocate.



04 Section	Q. No.													
	Marks													

प्र. क्र.
Q. No.

ii) Parameterized Constructor :-

A parameterized constructor takes additional argument to initialize instance variable. These parameter arguments allows for more flexibility when creating object.

It used when we want to set specific values for an objects attributes at the time of creation.

e.g

```
class Book :
    def __init__ ( self, Title, Author ) :
        self. Title = Title
        self. Author = Author

    def display_info ( self ) :
        print ( f" Title : { self. Title } " )
        print ( f" Author : { self. Author } " )

b = Book ( " Rich Dad Poor Dad ", " Robert T. kiosaki " )

b. display_info ( )
```

Display Current date and time.

$ date

Mon Sep 2 12:06:37 TST 2024

Display Cal for the year 2002

$ Cal 2002

2002

January

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

February

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28		

March

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

April

Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

May

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

June

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Assignment No. 8

Q. 1 Create a directory movie and ticket and create a text file Spiderman.txt in movie directory and do the following.

```
$ mkdir movie
$ mkdir ticket
$ ls
```

o/p: movie ticket

```
$ cd movie
$ touch spiderman.txt
$ ls
```

o/p spiderman.txt

Q. 2. 1] Copy the spiderman.txt from movie to ticket

⟹ `$ cp spiderman.txt tickets`

```
$ ls
```

o/p - ticket

2) rename tickets to tocken

⟹ `$ mv tickets tocken`

```
$ ls
```

o/p tocken

3) delete movie and tocken together.

Assignment No. 9

① explain various type of operator used in Shell Script

→ In shell script there are four types of operator are as follows:

① Arithmatic operators.

Operators	Example	description
+	$a + $b	perform addition
–	$a - $b	perform substract^n
/	$a / $b	perform division
*	$a * $b	perform multiplicat^n
%	$a % $b	perform remainder
^	$a ^ $b	perform powervalue

② Logical Operators

Operators	Examples	descriptions
!	! $b	logical NOT
-a	$a -a $b	logical AND
-o	$o -o $b	logical OR

③ Relational Operators

Operators	Examples	description
-eq	$a -eq $b	equal to (= =)
-ne	$a -ne $b	not equal (!=)
-lt	$a -lt $b	less than (<)
-le	$a -le $b	less than equal to (<=)
-gt	$a -gt $b	greater than (>)
-ge	$a -ge $b	greater than equal to (>=).

Assignment No. 1

1) Display Current date & time

⟹ $ date

o/p Mon Sep 14 12:10:37 IST 2024

2) Display the name of the user to log into the system

⟹ $ whoami

o/p vck91

3) Display calender 22 oct 2000.

⟹ $ cal oct 2000

o/p.

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

4) Print current working directory.

⟹ $ PWD

o/p /home/vck7/7781/05

5) Create two directory is river and mountain

⟹ $ ~~cd~~ ~~river~~

$ mkdir river

$ mkdir mountain

$ ls

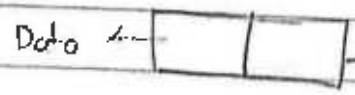
Assignment No 1.

- Data structure.

1) explain singly linked list and implement a singly linked list. Your program should support the following operations
- Traversing the list : print the data of each node in list.
- node searching for a node : Return True if the node is found and false otherwise.
- Prepending a node : Add a new node at the beginning of the list
- Removing a node by value : Delete the first occurence of a specific node.

→

1) linked list is a collection of similar elements
2) each elements points to the next element.
3) A singly linked list is a linear collection of nodes where each node contains two parts:
Data : The value stored in the node.
Next : A reference (or pointer) to the next node in the list.



Traversing the nodes :
To traverse the list, we start from the head node and keep moving to the next node until we reach end (when the next pointer is None).

Assignment No. 2

- The stack and Queue Adt

1) Explain stack and Implement it using a python list:

Implement a stack using a python list with operations: push(), pop(), peek(), is_empty(), and size().

→

A stack can be easily implemented using a list in python. The list allows us to use built-in methods such as append() (for pushing) and (pop) for (popping), which are O(1) operations.

```
class stack using list:
    def __init__(self):
        self.stack = []

    def push(self, item):
        self.stack.append(item)

    def pop(self):
        if not self.isEmpty():
            return self.stack.pop()
        else:
            raise IndexError("pop from
                empty stack")
    def peek(self):
        if not self.isEmpty():
            return self.stack[-1]
```

Assignment No. 1

Q.1] what is Object Oriented Programming?

Object-oriented programming (OOP) is a computer programming model that organizes software design around data.

- Object-oriented programming is about creating objects that contain both data and methods.
- for example, an object could represent a person with properties like a name, age and address & behaviors which such as walking, talking and running.

It could represent an email with properties like a recipient list, subject & body and behaviors like adding attachments & sending.

Q.2] list the features of Object Oriented programming.

features of OOP:

Six types of features of OOP.

1] Objects: In object oriented programming the object is an entity that has a state and behaviors associated with it. It may be real world object like keyboard, mouse,

Programming Question

1) write a lambda expression to return a dictionary with a word & its reversed word pair as a pair.

s = "welcome"
Expected Output = {"welcome": "emocle

⟹ num = int(input("Enter a number:"))
divisible_by_5_and_3 = lambda x: x%5 == 0 and x%3 == 0
print(divisible_by_5_and_3(num))

⟹ s = "welcome"
s = lambda a: {s: s[::-1]}
print(s(s))

Q.2 write a lambda expression to check if a number is divisible by both 5 and 3 (take user input).

⟹ num = int(input("Enter a number"))
divisible_by_5_and_3 = lambda x: x%5 == 0 and x%3 == 0
print(divisible_by_5_and_3(num))

Q.3 write a lambda expression to return a list element in upper case.
= ["apple", "mango", "orange", "berry", "coconut"]

Assignment No. 11

Q. 1. Explain the concept of class and objects. How do we specify attributes and behaviours within a class.

A class is a blueprint for creating objects. It defines what the objects will have (data) and what they can do (methods).

Example :

Think of a class as a blueprint for a building. The blueprint has details about the floors, rooms, doors and windows.

You can use this blueprint to build many different building.

- what does a class contain ?

Attribute : variables that store data relate to the class.

methods : Functions defined inside a class that describe the behaviours of the objects

constructor : The __init__ method is a special method that initalizes an objects attributes. Even if you don't explicitly define it, python provides a default constructor.

- Syntax to create a class :

```
Class className :
    def methodName (self) :
        # body of method.
```

Assignment

1) Explain the term data communication along with its components and organization standards.

⟹ Data communication -

In data communications, data generally are defined as information that is stored in digital form. Data communications is the process of transferring digital information between two or more points. Information is defined as the knowledge or intelligence. Data communications can be summarized as the transmission, reception, and processing of digital information. For data communications to occur, the communicating devices must be part of a communication system made up of a combination of hardware (physical equipment) and software (programs). The effectiveness of a data communications system depends on four fundamental characteristics : delivery, accuracy, timeliness, and jitter.

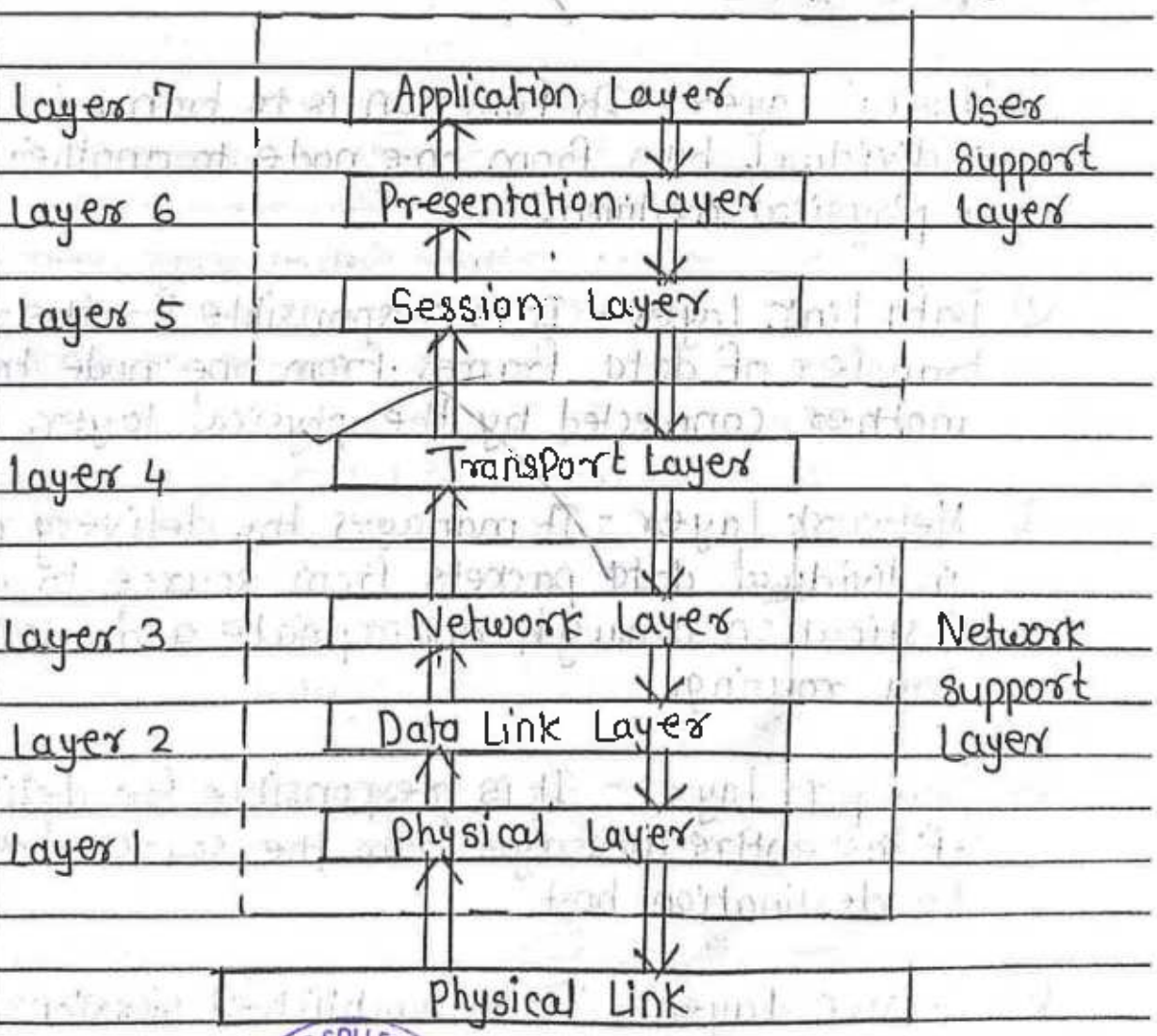
For A data communications system has five components

① Message -

The message is the information (data) to be communicated. Popular forms of information include text, numbers, pictures, audio, and video.

2) Explain OSI reference model

⇒ OSI or Open System Interconnection model was developed by International Standards Organization (ISO). It gives a layered networking framework that conceptualizes how communications should be done between heterogeneous systems. It has seven interconnected layers. The seven layers of the OSI Model are a physical layer, data link layer, network layer, transport layer, session layer, presentation layer and application layer, as known in the following diagram



users and offers services like dialog control and synchronization.

⑥ Presentation layer - It monitors syntax and semantics of transmitted information through translation, compression, and encryption.

⑦ Application layer - It provides high-level APIs (application program interface) to the users.

3) Explain TCP/IP reference model in detail.

Imp

⇒ TCP/IP Reference Model is a four-layered suite of communication protocols. It was developed by the DoD (Department of Defence) in the 1960's. It is named after the two main protocols that are used in the model, namely, TCP and IP. TCP stands for Transmission Control Protocol and IP stands for Internet Protocol.

The four layers in the TCP/IP protocol suite are -

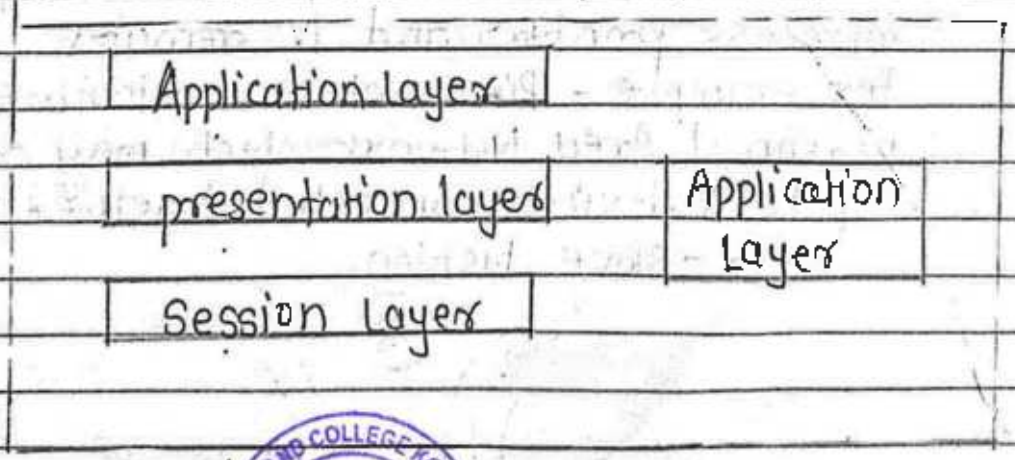
① Host-to-Network layer -
It is the lowest layer that is concerned with the physical transmission of data. TCP/IP does not specifically define any protocol here but supports all the standard protocols.

② Internet Layer - It defines the protocols for logical transmission of data over the network.

4) Differentiate between OSI and TCP/IP reference model.

- OSI model is a generic model that is based upon functionalities of each layer. TCP/IP model is a protocol-oriented Standard.
- OSI model distinguishes the three concepts, namely, services, interfaces, and protocols. TCP/IP does not have a clear distinction between these three.
- OSI model gives guidelines on how communication needs to be done, while TCP/IP protocols layout standards on which the Internet was developed so, TCP/IP is a more pratical model.
- In OSI, the model was developed first and then the protocols in each layer were developed. In the TCP/IP suite, the protocols were developed first and then the model was developed.
- The OSI has 7 seven layers while the TCP/IP has four layers.

The following diagram shows the corresponding layers of OSI and TCP/IP models -



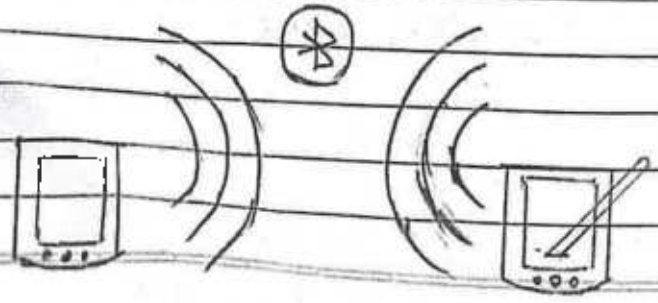
Transport layer	Transport layer
Network layer	Internet layer
Data link layer	
physical layer	Network Access Layer

Q5) Write short note on PAN, LAN, MAN and WAN.

Personal Area Network (PAN)

A personal Area Network (PAN) is smaller network which is very personal to a user. This may include Bluetooth enable enabled devices or infra-red enabled devices. PAN has connectivity range up to 10 meters PAN may include wireless computer keyboard and mouse, Bluetooth enabled headphones, wireless printers and TV remotes

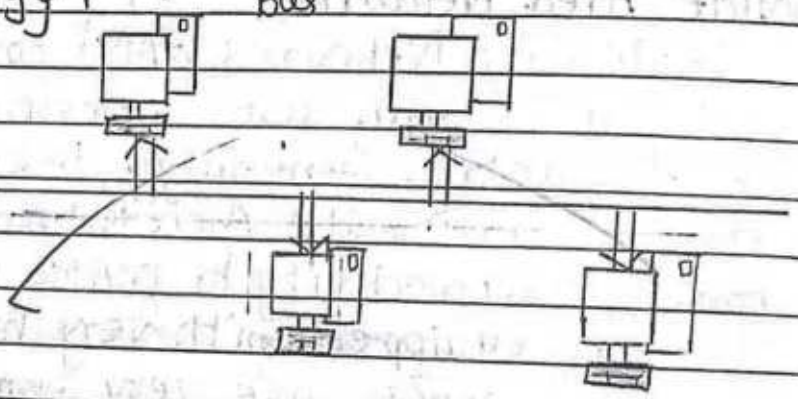
For example - Piconet is Bluetooth-enabled personal Area Network which may contain up to 8 devices connected together in a master-slave fashion.



Q6) Write a short note on Bus, Star, Ring, Tree and mesh topologies in brief.

⟹ Network Topologies. – Topology defines the structure of the network of how all the components are interconnected to each other.

1) Bus Topology :



- The bus topology is designed in such a way that all the stations are connected through a single cable known as a backbone cable.
- Each node is either connected to the backbone cable by drop cable or directly connected to the backbone cable
- When a node wants to send a message over the network, it puts a message over the network. All the stations available in the network will receive the message whether it has been addressed or not.
- The bus topology is mainly used in 802.3 (ethernet) and 802.4 standard networks.
- The configuration of a bus topology is quite simple as compared to other topologies.
- The backbone cable is considered as a

are a critical concern.

- Mesh topology is mainly used for wireless networks.
- Mesh topology can be formed by using the formula:
  Number of cables = (n * (n-1))/2;
  where n is the number of nodes that represents the network.

Q7) Write a short note on Twisted pair cable

⇒ Twisted Pair cable

(Twisted pair cabling comes in two varieties: shielded and unshielded. Unshielded twisted pair (UTP) is the most popular and is generally the best option for school networks.

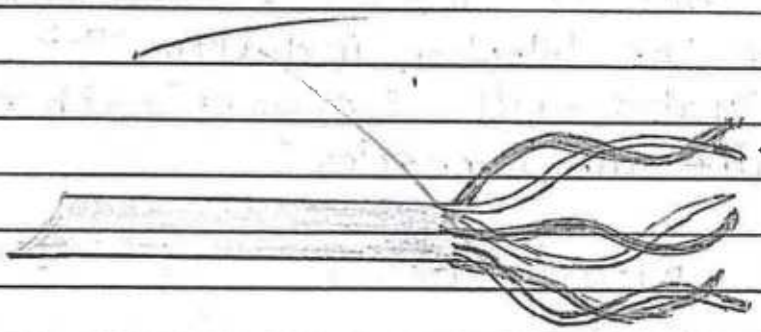


Fig: Unshielded twisted pair

- The quality of UTP may vary from telephone-grade wire to extremely high-speed cable.) The cable has four pairs of wires inside the jacket. Each pair is twisted with a different number of twists per inch to help eliminate interference from adjacent pairs and other

- Shielded Twisted Pair (STP) cable

Although UTP cable is the least expensive cable, it may be susceptible to radio and electrical frequency interference (it should not be too close to electric motors, fluorescent lights, etc). If you must place cable in environments with lots of potential interference or if you must place cable in extremely sensitive environments that may be susceptible to the electrical current in the UTP, shielded twisted pair may be the solution. Shielded cables can also help to extend the maximum distance of the cables.

Shielded twisted pair cable is available in three different configuration:

1. Each pair of wires is individually shielded with foil.
2. There is a foil or braid shield inside the jacket covering all wires (as a group).
3. There is a shield around each individual pair, as well as around the entire group of wires (referred to as double shield twisted pair).

Q8) Write a short note on Hub, Switch and Router, Bridge, Gateway.

⇒ Hub - Hub work in the physical layer of the OSI model. A hub is a device for connecting multiple Ethernet devices and making them

Q9) What is mean by DNS & Explain in brief.

Ans The domain name system (DNS) is a naming database in which internet domain names are located and translated into Internet Protocol (IP) addresses. The domain name system maps the name people use to locate a website to the IP address that a computer uses to locate that website.)

For example, if someone types "example.com" into a web browser, a server behind the scenes maps that name to the corresponding IP address. An IP address is similar in structure to ~~website~~. 203.0.113.72.

Web browsing and most other internet activities rely on DNS to quickly provide the information necessary to connect users to remote hosts. (DNS mapping is distributed throughout the internet in a hierarchy of authority) Access providers and enterprises as well as governments, universities and other organizations, typically have their own assigned ranges of IP addresses and an assigned domain name. They also typically run DNS servers to manage the mapping of those addresses. Most Uniform Resource Locators (URLs) are built around the domain name of the web server that takes client requests. DNS servers convert URLs and domain names into IP addresses that computers can understand and use. They translate what a user types into a

browses into something the machine can use to find a webpage. This process of translation and lookup is called DNS resolution.

What is Routing Algorithms. Explain its types.

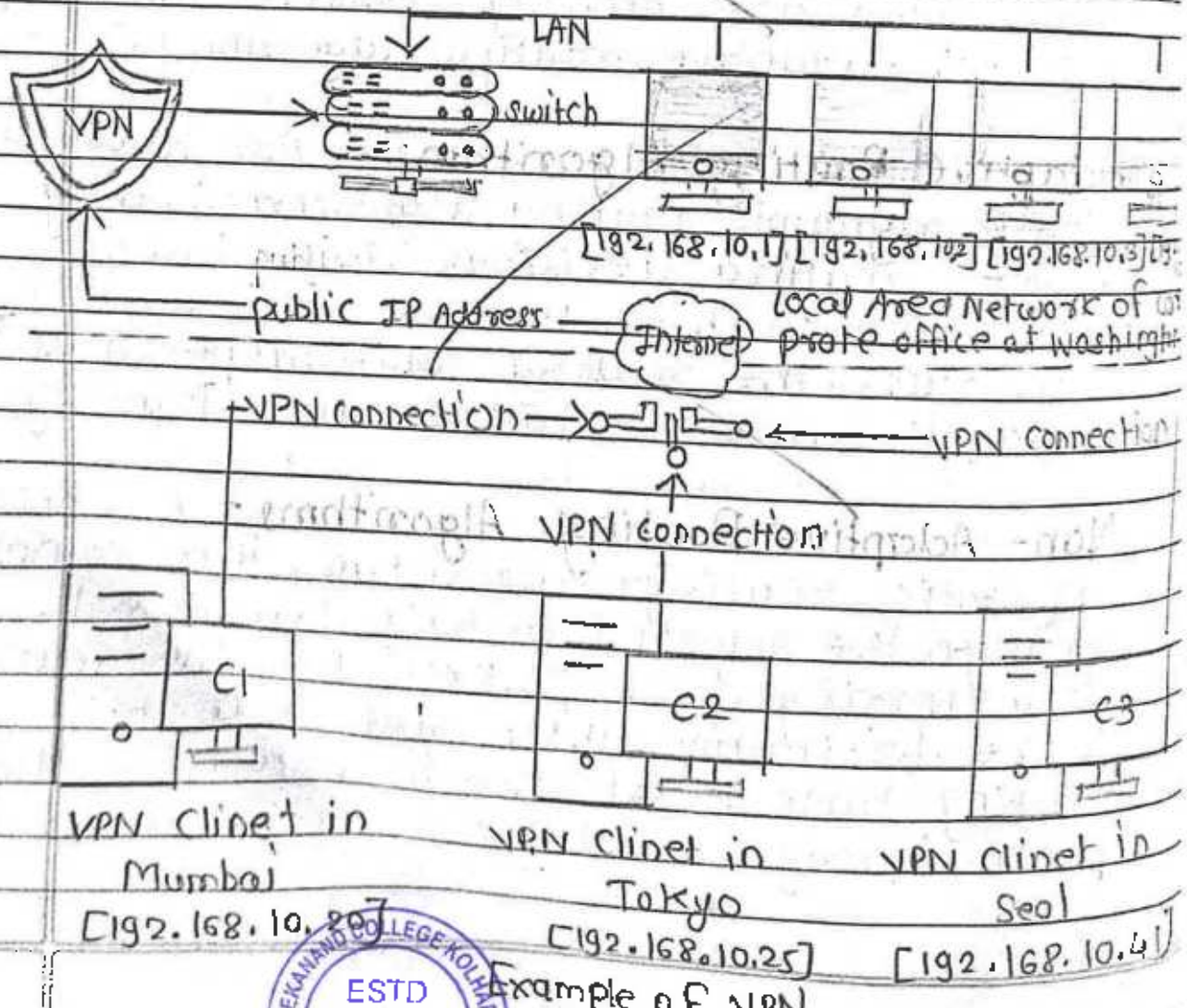
Routing Algorithms - A routing algorithm is procedure that lays down the route or path to transfer data packets from source to the destination. There can be many paths from the source to the destination, routing algorithm mathematically computes the best path, i.e. "least-cost path". Routing algorithms are broadly classified into adaptive and non-adaptive routing algorithms.

Adaptive Routing Algorithms - Also known as ~~Static~~ dynamic routing algorithms, they makes routing decisions dynamically depending on the network conditions. It constructs the routing table depending upon the network traffic and topology.

Non-Adaptive Routing Algorithms - Also known as static routing algorithms, they do not change the selected routing decisions for transferring data packets from the source to the destination. They construct a static routing table in advance to determine the path through which packets are to be sent.

Q.11) Explain virtual Private Networks

Virtual Private Networks (VPN) - is a technology that creates a safe and encrypted connection over a less secure network, such as the internet. A virtual Private Network is a way to extend a private network using a public network such as the internet. The name only suggests that it is a virtual "private network" i.e user can be part of a local network sitting at a remote location. It makes use of tunneling protocols to establish a secure connection.



Example of VPN

Page No.
Date

Q.12) Write a short note on Firewall, antivirus and content Filtering. Imp

⇒ Firewall - (A firewall is a network security system that manages and regulates the network traffic based on some protocols.) (A firewall establishes a barrier between a trusted internal network and the internet. (Firewall exist both as software that run on a hardware and as hardware appliances. Firewalls that are hardware-based also provide other functions like acting as a DHCP server for that network.

Most personal computers use software-based firewalls to secure data from threats from the internet. Many routers that pass data between networks contain firewall components and conversely, many (firewalls can perform basic routing functions.

Firewalls are commonly used in private networks or intranets to prevent unauthorized access from the internet.) Every message entering or leaving the intranet goes through the firewall to be examined for security measures.

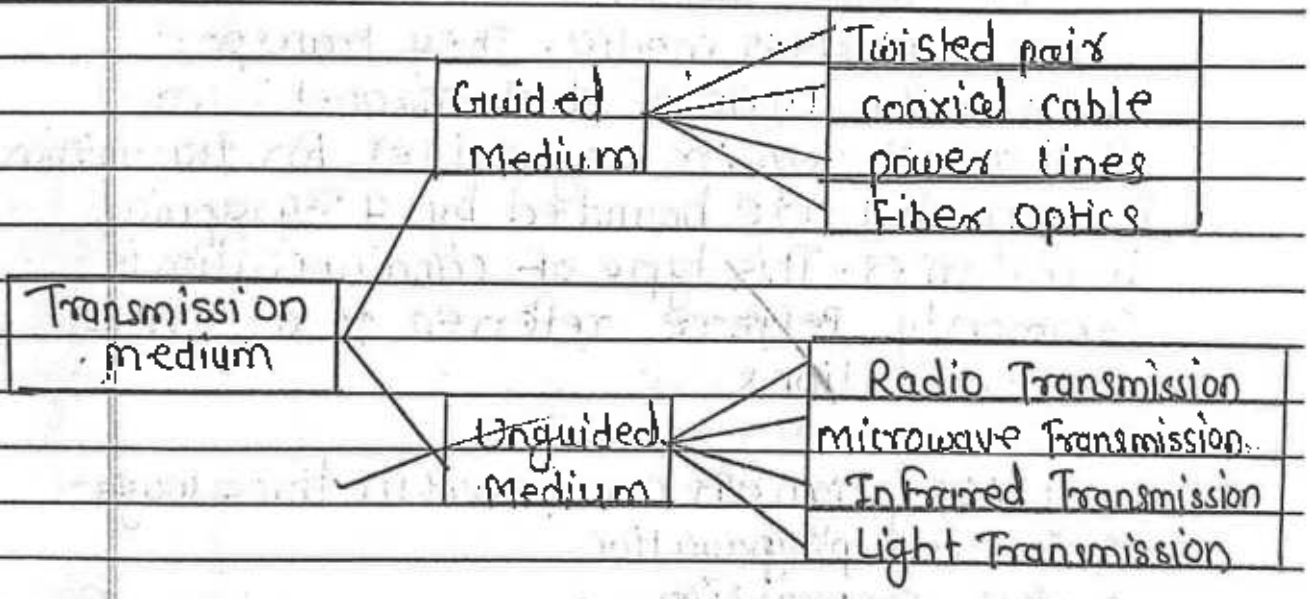
(An ideal firewall configuration consists of both hardware and software based devices. A firewall also helps in providing remote access to a private network through secure authentication certificates and logins.)

- E-mail filtering
- Screening of e-mail for spam
- other objectionable content.)

Q13) Describe the term Guided and Unguided transmission medium in brief

Transmission media are of two types-
- Guided Transmission Medium
- Unguided Transmission Medium

The following chart categorizes transmission media-



Guided Transmission Medium

Guided transmission media are also called bounded media or wired media. They comprise cables or wires through which data is transmitted. They are called guided

- Infrared transmission
- Light transmission

Q14) Explain different network protocols in detail

⇒ Network Protocols

There are various types of protocols that support a major and compassionate role in communicating with different devices across the network. These are:

1) Transmission Control Protocol (TCP)

(TCP is a popular communication protocol which is used for communicating over a network.) It divides any message into series of packets that are sent from source to destination and there it gets reassembled at the destination

2) Internet Protocol (IP) -

(IP is designed explicitly as addressing protocol. It is mostly used with TCP. The IP addresses in packets help in routing them through different nodes in a network until it reaches the destination system. (TCP/IP is the most popular protocol connecting the networks.

3) User Datagram Protocol (UDP)

UDP is substitute communication protocol to Transmission Control Protocol implemented

RLP (Resource Location Protocol)
L2TP (Layer Two Tunnelling Protocol)
PPTP (Point To point Tunnelling Protocol)
TFTP (Trivial file Transfer Protocol)
SNMP (simple Network Management Protocol)

Q15) Differentiate between Datagram approach and Virtual circuit.

Datagram Approach	Virtual Circuit
There is no dedicated transmission path.	There is also no dedicated transmission path.
There is a network responsible for individual packets.	There is a network responsible for packet sequence
The route is established for each table.	The route is established for the entire conversation
It is suitable for a few packets in data communication	It is suitable for longer data transmission.
There is no connection established	Connection is established
Datagram approaches are not dependable because the connectionless	It provides a dependable communication function to provide that each

Page No.
Date

Assignment - 1

Q1. What is System ? Explain in detail characteristics, Elements and Types of the System.

A System means "an organized relationship among functioning units or components".

A system exists because it is designed to achieve one or more objectives. We come into daily contact with the transportation system, the telephone system, the accounting System, the production system, and, for over two decades, the computer system. Similarly, we talk of the business system and of the organization as a system consisting of interrelated departments (subsystems) such as production, sales, personnel, and an information system.) None of these subsystems is of much use as a single, independent unit. When they are properly coordinated, however, the firm can function effectively and profitably.

Characteristics of System

① Organization -

Organization implies structure and order. It is the arrangement of components that helps to achieve objectives.

② Interaction

Interaction refers to the manner in which each component functions with other components of the system.

Page No.
Date

Assignment - 2

Q.1. What is Software Engineering & Explain Modelling with their different perspectives.

⟹ Software Engineering :

Definition -

"Software Engineering is the application of engineering to the design, development, implementation, testing and maintenance of a software in a systematic method".

Modelling -

(Software models are ways of expressing a software design.) Generally, some sort of abstract language or pictures are used to express the software design. Models are forms of description often adopted in software development. They are abstractions used to represent and communicate what is important, devoid of unnecessary detail, and to help developers deal with the complexity of the problem being investigated or the solution being developed.

Modelling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system. It is about representing a system using some kind of graphical notation, which is now almost always based on notations in the Unified Modeling Language (UML).

SCAN VI

Page No.
Date

Assignment - 1

1) Explain in detail OOAD concept. (object oriented Analysis and Design)

- Object -

An object is a real-world element in an object-oriented environment that may have a physical or a conceptual existence. Each object has -

- Identity that distinguishes it from other objects in the system.
- State that determines the characteristic the properties of an object as well as the values of the properties that the object holds.
- Behavior that represents externally visible activities performed by an object in terms of changes in its state.

Objects can be modelled according to the needs of the application. An object may have a physical existence, like a customer, a car, etc; or an intangible conceptual existence, like a project, a process, etc.

- Class

A class represents a collection of objects having same characteristic properties that exhibit common behavior. It gives the blueprint or description of the objects that can be created from it.

Assignment-3

Q. Explain the following Terms

1) Software Testing

(Software testing is a process of executing a program with objective of finding an errors. Software quality should be a primary concern in software development efforts. Software testing and evaluation are traditional methods of checking software quality. Software testing and evaluation are traditional methods of checking software quality. Software testing is successful only if all errors from the software are removed) A technique that finds maximum uncovered errors in a good testing technique. But no technique is available which will find all errors so we have to use a set of testing technique to find errors.

2) White Box Testing

(White Box Testing (WBT) is related with the structure (internal logic) of the program. to test the logic of program various test cases are designed which take care of the following.

- Every Statement in a program must be executed at least once.

Page No.
Date

Assignment - 2

Q1) Describe Mapping object Model to Database Schema.

⟹ • Mapping object Model to Database Schema:

(The data model describes the physical implementation of the database. This is the model of the internal database structure.) Persistent classes from the application model map to the data model. (The physical data model must map to the database. This is a simple one to one mapping, which is used to forward or reverse engineer the database structures.) The mapping between the application model and the database model is more complex. (As the data model can change because of normalization or de-normalization, the application model can change as well. Therefore, this mapping must be able to describe any and every possible relationship between the application model and the data model.)

Component to Database :

The database itself is the physical layer of data storage. It has no mapping into the application model.

Instead, the application has interfaces to the database. The database is associated to the application. The types of association is dependency.

Assignment - 2 (a)

Q) Explain with neat labelled diagram Different Perspective supported in UML.

Views :

To represent different perspective of software System different UML diagrams primarily based totally at the UML (Unified Modeling Language) are used. Each perspective focuses on some specific aspect and ignores the rest. Some can also additionally basic, why construct several models from different perspectives - Why not just construct one model that captures all perspectives ? The answer of the question is that

: If a single model is made to capture all the required perspectives, then it would be as complex as the original problem, and would be of little use. Once a System has been modelled from all the required perspectives, the constructed models can be refined to get the actual implementation of the System. It divides the whole software System into components and subcomponents.

The numerous perspectives which can be captured with the aid of using the UML diagrams are as follows :

- User's view
- Structural view
- Behavioral view
- Implementation view
- Environmental view

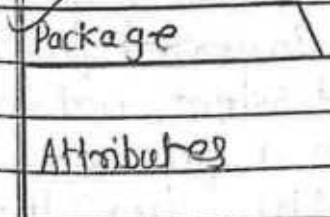
Assignment - 2 (b)

Q1) What is package diagram & Explain basic components of a package diagram

Package diagrams are structural diagrams used to show the organization and arrangement of various model elements in the form of packages. A package is a grouping of related UML elements, such as diagrams, documents, classes, or even other packages. Each element is nested within the package, which is depicted as a file folder within the diagram, then arranged hierarchically within the diagram. Package diagrams are most commonly used to provide a visual organization of the layered architecture within any UML classifier, such as a software system.

Basic components of a package diagram

The makeup of a package diagram is relatively simple. Each diagram includes only two symbols:

Symbol Image	Symbol Name	Description
	Package	For Group common elements based on data, behavior, or user interaction
	Dependency	Depicts the relationship betn one element (Package, named

Assignment No-2 (C)

Q1) What is Use case Diagram & Explain purpose of Use case Diagram.

(A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors and their relationships.) It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

- Purpose of Use case Diagrams

(The main purpose of a use case diagram is to portray the dynamic aspect of a system.) It accumulates the system's requirement, which includes both internal as well as external influences. It invokes perposon, use cases, & several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.

Following are the purposes of a use case diagram given below:

1) It gathers the system's needs.
2) It depicts the external view of the system.
3) It recognizes the internal as well as external factors that influence the system.
4) It represents the interaction between the actors.

ASSIGNMENT NO. : 01

Q1. What is Flask ? Explain features of Flask.

Ans. Flask is a web application framework written in python. It is used to build a website.

Web application Framework represents a collection of libraries and modules. It enables a web application developer to write applications without having to bother about low-level details such as protocols, thread management.

It is developed by Armin Ronacher. It is a tool that helps to create sites quicker. Flask makes development faster by offering code for all sorts of processes like database interaction or file activity.

- Features of Flask

- ~~of~~ ~~devlo~~ ~~devep~~ development server & debugger

Flask's development server and debugger are tools that help developers build web applications more effectively. The development server allows you to run and test your Flask applications locally before deploying them. Meanwhile, the debugger provides detailed error messages and

PAGE NO.
DATE :

ASSIGNMENT NO. : 02

*Q1. Explain decorator used in flask. What types of the parameters used with route decorator Implement a web application using decorator route.

Ans. In flask a decorator is a function that allows you to trap another function to add functionality or modify its behaviour. The most common decorator you encounter in flask is the @app.route() decorator. It is used to bind a URL to specific view function is executed. ~~@app.route() decorator.~~

- @app.route() Decorator

The @app.route() decorator takes several programmers

1) URL - The first and most common parameter is URL pattern you want to assign with a particular function.
e.g. @app.route('/') binds the root

2) methods - This specifies the HTTP method allowed for those route such as GET, POST etc.
e.g. @app.route('submit', methods ['POST'])
allows only POST requests to this URL.

PAGE NO.
DATE :

ASSIGNMENT NO. : 03

Q1. Explain request response cycle used in Flask.

Ans Request - Response ~~Cyl~~ Cycle in Flask

The request - response cycle in Flask is a fundamental mechanism through which web applications interact with clients.

It consist of ~~key~~ several key steps:

Client Request
The cycle begins when a client (such as a web browser or mobile app) sends an HTTP request to the Flask server. This request can involve various methods including GET, POST, PUT and DELETE, depending on the action the client intends to perform.

Routing
Flask utilizes a routing mechanism to map incoming requests to specific functions within the application. This is achieved using decorators like @app.route(). When a request matches a corresponding route, the function.

PAGE NO.
DATE :

ASSIGNMENT NO. : 04

Q1. What are Jinja templates used for? Explain with examples.

Ans Jinja templates are a templating engine used in Flask to seperate presentation logic from application logic. They allow you to render dynamic web pages with variable loops and conditional statements.)

Key features

Variable substitution
Control structures (if-else, for loops)
Functions and filters
Template inheritance

Example

```
# prog.py (Flask application)

from flask import Flask, render_template
app=Flask(__name__)

@app.route('/')
def index():
    title = "My Page"
    heading = "Welcome!"
```

PAGE NO.
DATE :

ASSIGNMENT NO. 05

Q1. How to create dynamic URLs in Flask ? Explain with example.

Ans ~~Flask provides~~

In Flask, dynamic URLs are created using angle brackets <> to capture variables.

Here's an example :

```
from Flask import Flask
app = Flask(__name__)

# Dynamic URL with string variable.
@app.route('/user/<username>')
def user(username):
    # show user profile for that user
    return 'User {{ username }}'

# Dynamic URL with integer variable
@app.route('/post/<int:post_id>')
def show_post(post_id):
    # show the post with given id
    return 'Post {{ post_id }}'
```

PAGE NO.
DATE:

ASSIGNMENT NO.:06

Q1. How do you create form in Flask? Explain with example.

Ans In Flask, we can create form using request method. The global request object to access incoming request data that will be submitted via the html ~~file~~ form.

Following is the example of creating form in flask —

```
from flask import Flask, render_Template, request
app = Flask(__name__)

@app.route('/')
def view_form():
    return render_Template("form.html")

@app.route('/data', methods=['POST'])
def data():
    if request.method == 'POST':
            data = request.form
            for key, value in data.items():
                return '<h1> {{ key }} </h1>
                        <h1> {{ value }} </h1>'
    else:
        return render_Template("form.html")
```

PAGE NO.
DATE :

ASSIGNMENT NO. : 07

Q1. What is make_response method ? Explain with example.

Ans. The make_response() method is a function in the Flask framework, a popular python web framework used to create web applications. It is used to create a Flask response object that can be customized with different attributes like headers, status codes, and response data.
By default, Flask automatically creates a response object when you returns a string or a dictionary from a route. However, if you need more control over the response you can use make_response().

Basic Syntax

```
from flask import Flask, make_response
response = make_response("Response body",
    status code)
```

Example

Simple example of make_response() to return a custom response with headers and status code.

PAGE NO.
DATE :

ASSIGNMENT NO. 08

Q1. We can inherit templates in Flask, Elaborate with example.

Ans Template inheritance allows you to build a base template that contains all the common elements of your site and defines blocks that child templates can override. This is particularly useful when you have multiple pages in your web application that share a common layout.

Base Template

This template contains the common structure such as the <head>, <header>, <footer>, etc. It also includes placeholders for content that will be replaced or added by child templates.

Child Templates

This template extends the base template and define content for the placeholder, known as blocks.

Example

Base Template (base.html)

PAGE NO.
DATE :

ASSIGNMENT NO. : 02

Q1. What is view function? Explain with exa

Ans A view function is Flask is a python function that is mapped to a specific URL routes in a Flask web application. It is responsible for handling requests and returns a response to the client.

Every view function is associated with a route and when a user accesses that rout the corresponding view function is executi

A view function is mapped a specific URL route using the @app.route() decorator.

It handles incoming requests (like GET, POST) a returns a response.

The response could be rendered HTML templat a string or any other valid response type

Ex.
```
from flask import Flask
app = Flask(__name__)

@app.route('/')  # URL route
def home():
    return 'Welcome!'  # Response
```

PAGE NO.
DATE :

ASSIGNMENT NO. : 10

Q. Write 50 objective questions in Flask with their options and answers.

1. What is Flask?
a) A web framework for Python
b) A python library for data analysis
c) A database management system.
d) A Javascript framework

Ans a) A web framework for Python

2. Flask framework is written in which of the following language?
a) Java
Ans. b) Python
c) Javascript
d) None of the above

Ans b) Python

3. Who developed the flask?
a) Guido van Rossum
b) Armin Ronacher
c) James Gosling
d) None of the above

Ans b) Armin Ronacher

PAGE NO.
DATE :

Theory Assignment no. : 01

Q.1. Write a note on history of Java Programming Language.

Ans. History of Java

- History of Java starts with green team.
- In 1990, Sun Microsystems funded an internal research project "Green" to develop a language for digital devices.
- James Gosling, Mike Sheridan and Patrick Naughton initiated this project in 1991 and resulted in a programming language 'Greentalk' which then renamed as 'Oak'.
- There was already a programming language called 'Oak' so green team again renamed it with 'Java' in 1995, the name taken from Java Coffee.
- Java was originally designed for small embedded systems in electronic application like set top boxes, TVs, VCRs, Toasters, Microwaves, Washing Machines, etc.
- So Java originally developed by James Gosling

and team at Sun Microsystems which is now a subsidiary (2010) of Oracle Corporation.

- Java is general purpose and object oriented programming language.
- The extension of java program file is .java
- First Java version was released in 1995 with name JDK Alpha and Beta.
- In very next year, 1996, JDK 1.0 released and since then number of Java versions have been released.
- JDK 14 i.e. Java SE 14 has released in March, 2020.

Q 2 Explain features of Java Programming language in details

Ans. Features of Java

1. Familiar, Simple
   - Java is very easy to learn and its syntax is simple, clean and easy to understand.
   - Java syntax is based on C and C++

PAGE NO.
DATE :

11. Distributed
- It is designed as distributed language for creating network applications.
- RMI and EJB are used to create distributed applications.
- Java applications can access remote objects or can communicate to many locations on the network.

12. Dynamic and Extensible
- Java is capable of loading classes on demand.
- Java programs support functions written in other languages like C and C++.
- It enables programmers to use efficient functions available from other languages.

Q.3. Explain Java Environment in detail.

Ans. Java Runtime Environment (JRE)

- Java Virtual Machine (JVM)
- JVM (= Java Virtual Machine) is an abstract machine that enables our computer to run a Java program.

PAGE NO.
DATE :

Q 4. Explain about Java Program life cycle.

Ans. The Java program lifecycle consists of several stages from writing code to running a program.

Following are the stages of Java Program lifecycle

1) Writing the code :

Java source code is written in .java files. Developers use (IDE) or a simple text editor for this purpose.

2) Compiling the code :

The java source code is compiled into byte code using the JAVA Compiler (javac). Byte code is executed by Java Virtual Machine (JVM).

3) Class loading :

The JVM loads the .class files into memory. This involves finding and loading the necessary classes and linking them to form a complete executable program. The class loader is responsible for this process.

4) Byte code Verification :

Before actual execution, the JVM verifies the byte code to ensure that it is not tempered. This step prevents the

Q.5. Explain Java statements, keywords, literals, blocks.

Ans. ■ Java Statements

A statement in java forms a complete command to be executed and can include one or more expressions.

Statements in Java can be one of the following:

- Comments
- Variable Declaration / Initialization
- Control statement (if, switch, while, for, etc.)
- Expressions
- Block of statements

■ Java Keywords

- Keywords are predefined, reserved words used in program that have special meanings to the compiler.
- We can not use these reserve words as variable or class or method name.
- Keywords in Java are case sensitive and all keywords are in lower case.

PAGE NO.
DATE :

- Java Blocks

A java block is group of statements (zero or more) that are enclosed within the curly opening and closing bracket {}.

e.g.
```
class class name {          // start of block
    // class body
}                           // end of block
```

Q 6. Write a note on comments in Java.

Ans. Java Comments

Java supports three types of comments

- Single Line Comment

e.g. // Single line comment

- Multiline Comment

e.g. /* Multiline Comment. These comment can span multiple lines. The compiler ignores all text up until */

- Documentation Comment

e.g. /** Javadoc comment */ The compiler ignores

this text too. However, the javadoc tool looks for these comments and interprets tags for documentation generation purposes. These are used to generate documentation or reference for java application / software. Javadoc tool recognizes these tags.

```
    @author XYZ
    @version 14.0.1
**/
```

Q7. Explain different tools used in Java Programming Language.

Ans. Java Tools

- javac - It is java compiler used to compile java program which converts it into byte code.
- java - It is java interpreter, also known as application launcher. Used to run java program which converts and executes java code.
- javadoc - It creates HTML help file from source code.
- jdb - It is java debugger used to debug java program.

PAGE NO.
DATE :

- appletviewer - It is applet launcher used to run applets.

Q 8. Explain different data types used in Java.

Ans. There are two groups of data types :

A] Primitive Data Types
B] Reference / Object / Non-Primitive Data Types

A] Primitive Data Types

There are eight primitive data types supported by java. Primitive data types are predefined by the language and named by a keyword.

i. Integral Data Types

There are 4 types based on integral values byte, short, int, long. All numeric types are signed. There are no unsigned types in java.

Type	size	Range	Default value
byte	8 bits	-128 to +127	0
short	16 bits	-32768 to +32767	0
int	32 bits	-2147483648 to +2147483647	0
long	64 bits	-9223372036854775808 to +9223372036854775807	0L

PAGE NO.
DATE :

- In Java array indexes start from zero.

e.g. int num[] = new num[10];

OR

String fruits[] = { "apple", "banana", "mango" };

Q9. Explain different categories of operators used in Java.

Ans In Java there are following types of operators

i) Arithmetic Operators
ii) Relational Operators
iii) Bitwise Operators
iv) Logical Operators
v) Assignment Operators
vi) Misc Operators

i) Arithmetic Operators

Arithmetic operators are used in the same way that they are used in algebra.

Operator	Description	Example
+	Addition - Add values on either side of operator.	A+B
-	Subtraction - Subtracts right hand operand from left hand operand	A-B

||	Logical OR : Returns true if any one of the value is true else returns false	Exp1 || Exp2	x<6 || x>4 True
!	Logical NOT: This is unary operator. The operator returns true if result is false and vice versa	! Exp	!(x<6) False !(x>5) True

Q10. Explain about variables, its conventions, its declaration and its initialization.

Ans. Variables

- Variables are a symbolic name given to a memory location.
- Variables may be of type object or simple variables.
- Like other compiled languages, variables must be declared before they can be used as Java is statically typed language.
- All variables have a type which is enforced by the compiler.
- Variables in Java are case sensitive.
- The general form of a variable declaration is :

  Type variable_name [= value];

e.g.
```
int total;
float interest = 6.0;
boolean isFinished;
String name;
```

datatype variable_name = value;

OR

datatype variable_name;
variable_name = value;

Example:

int age = 30;

OR

int rollno;
rollno = 30;

Q 11. Explain conditional statements in Java with examples.

Ans. Conditional or Decision Making statements

- Decision Making in programming is similar to decision making in real life.
- In programming also we face some situations where we want a certain block of code to be executed when some condition is fulfilled.
- A programming language uses control statements to control the flow of execution of program based on certain conditions.

Conditional statements in Java are as follows:

i. if statement - if statement is the most simple decision making statement. It is used to decide ~~what~~ whether a certain statement or block of

```
switch (a) {
    case 1 : System.out.println("Monday");
             break;
    case 2 : System.out.println("Tuesday");
             break;
    default : System.out.println("Not a week
              day number");
  }
  }
}
```

Q.12 Explain looping statements in Java with examples.

Ans. Looping statements

Looping in programming languages is a feature which facilitates the execution of a set of instructions / functions repeatedly while some condition evaluates to true.

Looping statements in Java are as follows:

- while statement,

A while loop is a control flow statement that allows code to be executed repeatedly based on a given Boolean condition. It is used to run a specific code until a certain condition is met. It is also called "Entry Control loop".

Syntax:

```
for (initialization ; condition ; increment / decrement)
    // statement/s or code to be executed
}
```

Example:

```
class Fordemo {
    public static void main (String args[]) {
    int x
    for (int i=0 ; i<5 ; i++) {
        System.out.println ("Value of i is :" +i
    }
    }
}
```

Q.13. Explain branching statements (break and continue) with examples.

Ans. In Java, branching statements 'break' and 'continue' are used to control the flow of loops like for, while and do-while.

- break statement

The break statement is used to terminate the loop or switch statement in which it is present.
When a break statement is encountered inside

e.g.
```
class Demo {
    public static void main(String args[]){
        for(int i=1; i<=10; i++){
            if(i==5){
                continue;
            }
            System.out.println(i);
        }
    }
}
```

// Output :

1
2
3
4
6
7
8
9
10

Q14. Explain about arrays in Java with example

Ans. Arrays are objects which store multiple values of the same type. It can hold primitive types as well as object references.

PAGE NO.
DATE :

Q 15. Explain about input reading from command line arguments with example.

Ans. Command line arguments allows user to pass input parameters to a program when they run it from the command line. These input can be used to alter the programs behavior without changing the code.

```
main method header
public static void main (String args[]){
}
```

Input values can be passed to main method from the command line ~~arg~~ and the code ~~Passed as arguments are~~

Passed arguments are stored as String, so there are conversions from String to Primitive types. —

Method	Purpose
Byte.parseByte	Returns the byte equivalent of a string
Short.parseShort	Returns the short equivalent of a string
Integer.parseInt	Returns the int equivalent of a string
Long.parseLong	Returns the long equivalent of a string
Float.parseFloat	Returns the float equivalent of a string
Double.parseDouble	Returns the double equivalent of a string

To pass command line arguments, use the following syntax :-

> java class_name arg1 arg2 ---

e.g.
```
class Demo {
    public static void main (String args[]) {
    int n = Integer.parseInt (args[0]);
    if (n > 0) {
        System.out.println (" Number is Positive ");
    }
    }
}
```

// Outputs : java Demo 3
Number is Positive

Q16. Explain about input reading from console using BufferedReader class with example.

Ans. System.in : It is predefined stream object which is used to read inputs in the form of byte oriented stream.

- InputStreamReader : To read the characters, we should wrap this stream into a character oriented stream using InputStreamReader. This let us read characters.

```
        if (n%2==0)
            System.out.println("Given no. is Even");
    }
}
```

Q 17. Explain about input reading from console using Scanner class with example.

Ans. In Java, input can also be given to a program by using a Scanner class.

It is in the java.util package.

The Java Scanner class provides nextXXX() methods to return the type of value such as nextInt(), nextByte(), nextShort(), next(), nextLine, nextDouble(), nextFloat(), nextBoolean(), etc.

- Object creation of Scanner class

```
import java.util.Scanner;
....
....
Scanner sc = new Scanner(System.in);
```

eg class Demo{
pub

Assignment: 1

Q.1. What is System? Explain in detail characteristics, Elements and Types of the System.

A System means "an organized relationship among functioning units or components"

A system exists because it is designed to achieve one or more objectives. We come into daily contact with the transportation system, the telephone system, the accounting system, the production system, and, for over two decades, the computer system. Similarly, we talk of the business system and of the organization as a system consisting of interrelated departments (subsystems) such as production, sales, personnel, and an information system.) None of these subsystems is of much use as a single, independent unit. When they are properly coordinated, however, the firm can function effectively and profitably.

Characteristics of System

① Organization -

Organization implies structure and order. It is the arrangement of components that helps to achieve objectives.

② Interaction

Interaction refers to the manner in which each component functions with other components of the system.

③ Interdependence -

Interdependence means that parts of the organization or Computer System depend on one another.

④ Integration -

Integration refers to the holism of Systems. Synthesis follows analysis to achieve the central objective of the organization.

⑤ Central objective -

Objectives may be real or Stated. Although a stated objective may be the real objective, it is not uncommon for an organization to state one objective and operate to achieve another.

Elements of a System

In most cases, Systems analysts operate in a dynamic environment where change is a way of life. The environment may be a business firm, a business application, or a computer System. To reconstruct a System, the following key elements must be considered.

① Outputs and inputs -

(A major objective of a System is to produce an output that has value to its user) Whatever the nature of the output (goods, services, or information), it must be in line with the expectations of the intended user. Inputs are the elements (material, human resource

after all, a user considers a system as providing a set of functionalities.

2) Structural view:
From the name structural view, this view defines the structure of the problem. The structure of the problem can be defined on the basis of objects or classes which is very important to understand the working of the system and for its implementation. This view is also used to capture the relationships among objects or classes. This view is refferred as the static model, reason behind it is that the structure of a system doesn't change with time.

3) Behavioural view:
This view defines how objects interact with each other, just to realize the system behaviour. It therefore constitutes is referred as dynamic model of the system, reason behind this is that it capture the time-dependent (dynamic) behaviour of the system.

4) Implementation view: This view capture the various important components of the system and their inter-dependencies. For example the implementation view might show the GUL part, middleware, and database part as the different parts and also would capture their inter-dependencies.

Assignment - 2 (b)

8) What is package diagram & Explain basic components of a package diagram

Package diagrams are structural diagrams used to show the organization and arrangement of various model elements in the form of packages. A package is a grouping of related UML elements, such as diagrams, documents, classes, or even other packages. Each element is nested within the package, which is depicted as a file folder within the diagram, then arranged hierarchically within the diagram. Package diagrams are most commonly used to provide a visual organization of the layered architecture within any UML classifier, such as a software system.

Basic components of a package diagram

The makeup of a package diagram is relatively simple. Each diagram includes only two symbols:

Symbol Image	Symbol Name	Description
Package Attributes	Package	Group common elements based on data, behavior, or user interaction
---------->	Dependency	Depicts the relationship between one element (Package, named

and information) that enter the System for processing. Output is the outcome of processing.)

② Processor (s):
The processor is the element of a system that involves the actual transformation of input into output.) It is the operational component of a system. Processors may modify the input totally or partically, depending on the specifications of the output. This means that as the output specifications change so does the processing. In some cases, input is also modified to enable the processor to handle the transformation.

③ Control
The control element guides the System. It is the decision-making subsystem that controls the pattern of activities governing input, processing, and output.)

④ Feedback -
Control in a dynamic system is achieved by feedback.

⑤ Environment.
The environment is the "suprasystem" within which an organization operates.

⑥ Boundaries and Interfaces
A System should be defined by its boundaries - the limits that identify its components, processes and interrelationship when it interfaces with another system.

Assignment - 2 (a)

Q) Explain with neat labelled diagram Different Perspective supported in UML.

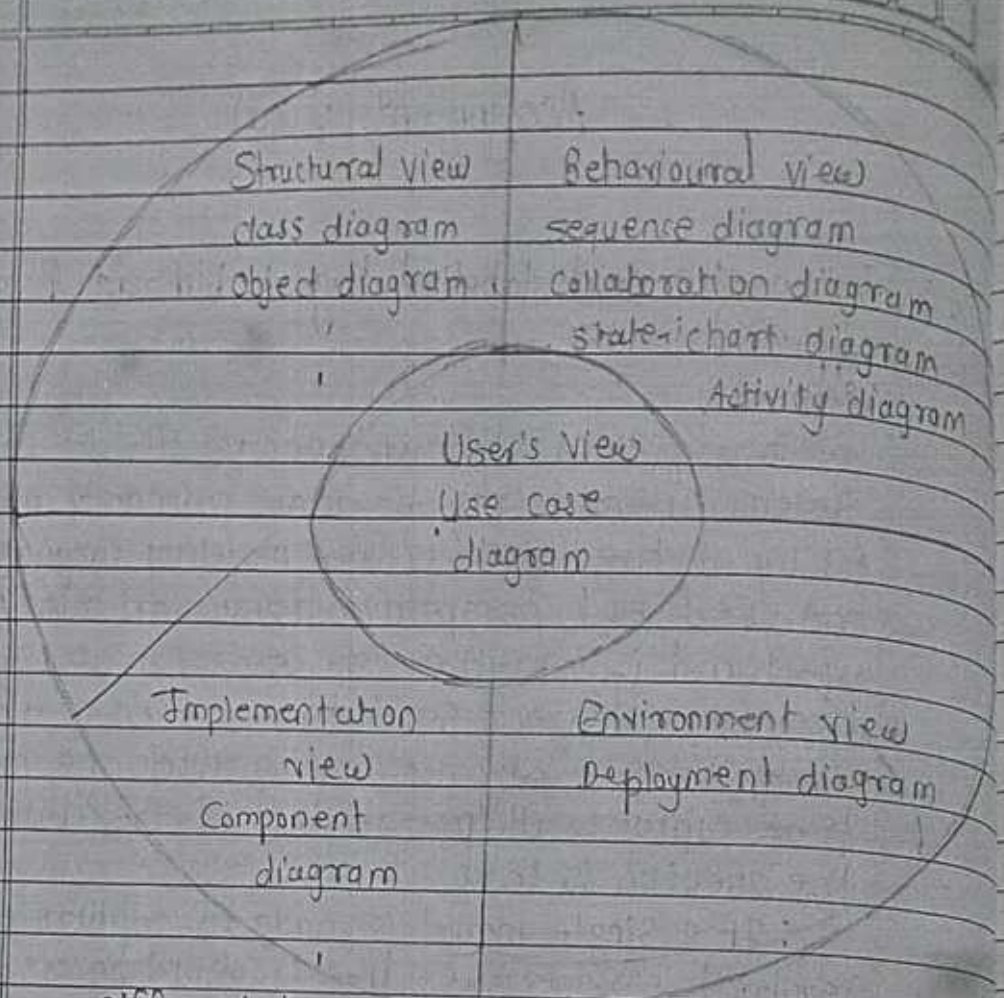
Ans Views :

To represent different perspective of software system different UML diagrams primarily based totally at the UML (Unified modeling language) are used. Each perspective focuses on some specific aspect and ignores the rest. Some can also additionally ask, why construct several models from different perspectives - Why not just construct one model that captures all perspectives ? The answer of the question is that

: If a single model is made to capture all the required perspectives, then it would be as complex as the original problem, and would be of little use. Once a system has been modelled from all the required perspectives, the constructed models can be refined to get the actual implementation of the system. It divides the whole software system into components and subcomponents.

The numerous perspectives which can be captured with the aid of using the UML diagrams are as follows :

- User's view
- Structural view
- Behavioral view
- Implementation view
- Environmental view



Different types of diagrams and perspective supported by UML

- These are explained as following below:

① Users view : Users view defines numerous functionalities of the device this is to be had to its user. This view captures the view of the system in terms of the functionalities offered by the system to its users. This view is just like the black box in which the various details of the like the system like interior structure, implementation etc is hidden from the user. It view is referred as central view because all other views have to conform to this view. It is indeed remarkable that even for object oriented development, we need a functional view. That is because

③ Interdependence -

Interdependence means that parts of the organization or Computer System depend on one another.

④ Integration -

Integration refers to the holism of Systems. Synthesis follows analysis to achieve the central objective of the organization.

⑤ Central objective -

Objectives may be real or stated. Although a stated objective may be the real objective, it is not uncommon for an organization to state one objective and operate to achieve another.

Elements of a System

In most cases, Systems analysts operate in a dynamic environment where change is a way of life. The environment may be a business firm, a business application, or a computer System. To reconstruct a System, the following key elements must be considered.

① Outputs and inputs -

(A major objective of a System is to produce an output that has value to its user) Whatever the nature of the output (goods, services, or information), it must be in line with the expectations of the intended user. Inputs are the elements (material, human resources,

Practical Session TY BCA

Q. Write a Java Program for :

1. Performing arithmetic operations on two numbers. (Consider all integral and floating point data types).

```
// Program to perform arithmetic operations

class ArithmeticOpt {
    public static void main(String args[]){
    int x=4;
    int y=9;
    double f1=4.5;
    double f2=6.4;

    System.out.println("Arithmetic Operations of
Integer values");
    System.out.println("Num 1 = " + x);
    System.out.println("Num 2 = " + y);
    System.out.println("Addition of "+x+" + "+y+
" = "+(x+y));
    System.out.println("Subtraction of "+ y/x +" - "+
xy+" = "+(y-x));
    System.out.println("Multiplication of "+x+
" * "+y+" = "+(x*y));
    System.out.println("Division of "+y+" / "+x+
" = "+(y/x));
```

```
        System.out.println("Modulus of " + x +
y + " = " + (x % y));

        System.out.println("Arithmetic Operat
of float values")
        System.out.println("Num-1 = " + f1);
        System.out.println("Num 2 = " + f2);
        System.out.println("Addition of " + f1 + "
f2 + " = " + (f1 + f2));
        System.out.println("Subtraction of " + f2
+ f1 + " = " + (f2 - f1));
        System.out.println("Multiplication of
" * " + f2 + " = " + (f1 * f2));
        System.out.println("Division of " + f2 + "
f1 + " = " + (f2 / f1));
        System.out.println("Modulus of " + f1 + " %
f2 + " = " + (f1 % f2));
    }
}
```

2. Checking given number is positive ~~or~~ or negative.

```
// Program to check given no is positive
   or negative

class PosNeg {
    public static void main(String args[]){
    int x = 8;
```

```
        if (x>0)
            System.out.println("The given no. "+x+
            " is Positive");
        else
            System.out.println("The given no. "+x+"
            is Negative");
        }
}
```

3. Checking given number is odd or even.

```
// Program to check given no. is even or odd

class EvenOdd{
    public static void main(String args[]){
    int n=3;
    if(n%2==0)
        System.out.println("Given no. "+n+" is
        Even");
    else
        System.out.println("Given no. "+n+" is
        Odd");
    }
}
```

4. Checking given year is leap or not.

```
// Program to check whether given year is
   leap or not.

class LeapYear{
    public static void main(String args[]){
    int y=2000;
    if(y%4==0 && y%100==00)
            System.out.println("Given year "+y+"
            is a Leap Year");
    else
            System.out.println("Given year "+y+
            is not a Leap Year");
    }
}
```

5. Checking large number from three number.

```
// Program to check maximum no. between 3
   no.s

class MaxNum{
    public static void main(String args[]){

    int n1=5;
    int n2=3;
    int n3=7;
```

```
System.out.println("Num 1 = " + n1);
System.out.println("Num 2 = " + n2);
System.out.println("Num 3 = " + n3);

if (n1>n2 && n1>n3)
        System.out.println("The given no. " +
        n1+" is Maximum");
else if(n2>n1 && n2>n3)
        System.out.println("The given no. " +
        n2+" is Maximum");
else if(n3>n1 && n3>n2)
        System.out.println("The given no. " +
        n3+" is Maximum");

else
        System.out.println("All numbers are equal");
    }
}
```

6. Checking given number is prime or not.

```
class Prime {
    public static void main(String arg[]){

    int n=5;
    int i=2;

    System.out.println("Number = " +n);
```

```
while (i<n){
        if(n%i=0){
                System.out.println("The
                no." +n+ " is not Prime
                break;
        }
        i=i+1;
}
if(i==n){
        System.out.println("The given n
        n+" is Prime Number");
}
}
}
```

7. Checking given number is carmstrong or not

```
// Program to check given no. is carmstrong
   or not

class ArmStrong{
    public static void main(String args[]){

        int a = 370;
        int sum=0;
        int n=a;
        int d;
```