

**Bangladesh Civil Service – 37<sup>th</sup>(ICT)**  
**Date : 2017 Time : 3 Hours. Full marks : 200**

**1.(a) Distinguish between TCP and UDP .**

SL	E-Commerce	M-commerce
Definition	It is a communications protocol, using which the data is transmitted between systems over the network. In this, the data is transmitted into the form of packets. It includes error-checking, guarantees the delivery and preserves the order of the data packets.	It is same as the TCP protocol except this doesn't guarantee the error-checking and data recovery. If you use this protocol, the data will be sent continuously, irrespective of the issues in the receiving end.
Design	TCP is a connection oriented protocol.	UDP is a connection less protocol.
Reliable	As TCP provides error checking support and also guarantees delivery of data to the destination router this make it more reliable as compared to UDP.	While on other hand UDP does provided only basic error checking support using checksum so the delivery of data to the destination cannot be guaranteed in UDP as compared to that in case of TCP.
Data transmission	In TCP the data is transmitted in a particular sequence which means that packets arrive in-order at the receiver.	On other hand there is no sequencing of data in UDP in order to implement ordering it has to be managed by the application layer.
Performance	TCP is slower and less efficient in performance as compared to UDP. Also TCP is heavy-weight as compared to UDP.	On other hand UDP is faster and more efficient than TCP.
Retransmission	Retransmission of data packets is possible in TCP in case packet get lost or need to resend.	On other hand retransmission of packets is not possible in UDP.

**(b)What is OSI model ? Draw the diagram of OSI model .**

**Open System Interconnection.** The Open Systems Interconnection (OSI) **model** describes seven **layers** that computer systems use to communicate over a network. It was the first standard **model** for network communications, adopted by all major computer and telecommunication companies in the early 1980s.



Address	Layer	PDU
	Application	
	Presentation	Data
	Session	
Port	Transport	Segment
IP	Network	Packet
MAC	Data Link	Frame
	Physical	Bits

### OSI model

(c) State the protocol of each layer .

Layer	Name	Protocols
Layer 7	Application	SMTP, HTTP, FTP, POP3, SNMP
Layer 6	Presentation	MPEG, ASCH, SSL, TLS
Layer 5	Session	NetBIOS, SAP
Layer 4	Transport	TCP, UDP
Layer 3	Network	IPV5, IPV6, ICMP, IPSEC, ARP, MPLS.
Layer 2	Data Link	RAPA, PPP, Frame Relay, ATM, Fiber Cable, etc.
Layer 1	Physical	RS232, 100BaseTX, ISDN, 11.

## 2.(a) Difference between E-Commerce and M-commerce

	E-Commerce	M-commerce
Devices used	Handheld devices such as smartphones and tablets	Computers, laptops, ...
Connection-dependency	Mandatory (*)	Mandatory (*)
Reachability	With push notifications in place, mobile commerce can reach a wider range of users even when they're on the go	Limited
Mobility	High: Users can make transactions everywhere as long as they are connected to the Internet.	Low: Users can make transactions on their computers and laptops with limited mobility
Ease of Use	User Experience on mobile devices is optimized for as little taps-on-screen per transaction as possible	Harder to navigate in the desktop websites of eCommerce stores.
Platform	Web stores and native apps	Web stores
	Caller's rate, mobile banking or user's credit card	Credit Cards
Security	Rely on the web security combined with built-in mobile security features	Rely mainly on the web security
Location-tracking capability	Yes	No

### (b) What are barriers of E-Commerce ?

- Search Engine Rankings
- Web Design
- Mobile Browser compatibility
- Slow Website Performance
- Shopping Cart

### (c) What is cryptography and public key infrastructure are important?

**PKI** is a **critical** part of the IT strategic backbone. **PKI** is **important** because the certificate-based technology helps organizations establish trusted signature, encryption, and identity between people, systems, and things.

### 3.(a) How does dynamic memory allocation help in managing data .

The process of allocating memory at runtime is known as dynamic memory allocation. Aside from being able to store simple structured data types, dynamic memory allocation can combine separately allocated structured blocks to form composite structures that expand and contract as needed.

In this data structure that allows you to store and remove data blocks of arbitrary size, in arbitrary order.

(b) Illustrate singly, doubly, circular linked list  
Solution:

### Singly Linked List

It is the most common. Each node has data and a pointer to the next node.



Node is represented as:

```
struct node {  
  
    int data;  
  
    struct node *next;  
  
}
```

A three-member singly linked list can be created as:

```
/* Initialize nodes */  
  
struct node *head;  
  
struct node *one = NULL;  
  
struct node *two = NULL;  
  
struct node *three = NULL;
```

```
/* Allocate memory */  
  
one = malloc(sizeof(struct node));  
  
two = malloc(sizeof(struct node));  
  
three = malloc(sizeof(struct node))  
  
/* Assign data values */  
  
one->data = 1;  
  
two->data = 2;  
  
three->data = 3;  
  
/* Connect nodes */  
  
one->next = two;  
  
two->next = three;  
  
three->next = NULL;  
  
/* Save address of first node in head */
```

head = one;**Doubly Linked List**

We add a pointer to the previous node in a doubly-linked list. Thus, we can go in either direction: forward or backward.



A node is represented as

```
struct node {

    int data;

    struct node *next;

    struct node *prev;

}
```

A three-member doubly linked list can be created as

```
/* Initialize nodes */

struct node *head;

struct node *one = NULL;

struct node *two = NULL;

struct node *three = NULL;

/* Allocate memory */

one = malloc(sizeof(struct node));

two = malloc(sizeof(struct node));
```

```
three = malloc(sizeof(struct node));

/* Assign data values */

one->data = 1;

two->data = 2;

three->data = 3;

/* Connect nodes */

one->next = two;

one->prev = NULL;

two->next = three;

two->prev = one;

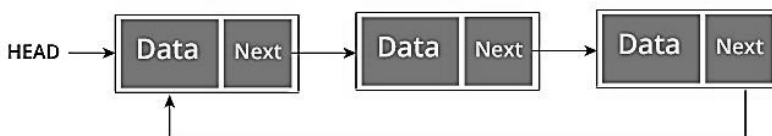
three->next = NULL;

three->prev = two;

/* Save address of first node in head */
```

head = one; Circular Linked List

A circular linked list is a variation of linked list in which the last element is linked to the first element. This forms a circular loop.



A circular linked list can be either singly linked or doubly linked.

- for singly linked list, next pointer of last item points to the first item
- In doubly linked list, prev pointer of first item points to last item as well.

A three-member circular singly linked list can be created as:

```
/* Initialize nodes */
struct node *head;
struct node *one = NULL;
struct node *two = NULL;
struct node *three = NULL;

/* Allocate memory */
one = malloc(sizeof(struct node));
two = malloc(sizeof(struct node));
three = malloc(sizeof(struct node));

/* Assign data values */
one->data = 1;
two->data = 2;
three->data = 3;

/* Connect nodes */
one->next = two;
two->next = three;
three->next = one;

/* Save address of first node in head */
head = one;
```

**4.(a)What is the difference between preemptive and non preemptive scheduling .**

**(b) What is aging ? Why it is used in OS ?**

**Solution:**

To avoid starvation, we use the concept of Aging. In Aging, after some fixed amount of time quantum, we increase the priority of the low priority processes. By doing so, as time passes, the lower priority process becomes a higher priority process.

For example, if a process P is having a priority number as 75 at 0 Ms. Then after every 5 ms(you can use any time quantum), we can decrease the priority number of the process P by 1(here also instead of 1, you can take any other number). So, after 5 MS, the priority of the process P will be 74. Again after 5 MS, we will decrease the priority number of process P by 1. So, after 10 MS, the priority of the process P will become 73 and this process will continue. After a certain period of time, the process P will become a high priority process when the priority number comes closer to 0 and the process P will get the CPU for its execution. In this way, the lower priority process also gets the CPU. No doubt the CPU is allocated after a very long time but since the priority of the

process is very low so, we are not that much concerned about the response time of the process. The only thing that we are taking care of is starvation

**5.(a) What is static IP ? (b) Why do we need static private ip address and advantages of it .**

**6(a) What is object oriented programming?**

Object Oriented programming (OOP) is a programming paradigm that relies on the concept of classes and objects. It is used to structure a software program into simple, reusable pieces of code blueprints (usually called classes), which are used to create individual instances of objects. There are many object-oriented programming languages including JavaScript, C++, Java, and Python.

**(b) What are the main features of OOP?**

1. Encapsulation – Hide unnecessary details in classes and deliver a simple and clear interface for working.
2. Inheritance – explain how the class hierarchies develop code readability and support to the reuse of functionality.
3. Data Abstraction – to deal with objects since their important features and ignore all other details.
4. Polymorphism – explain how to work with different object in the same manner, which explain the specific implementation of some abstract behavior.

**(c) How can we solve problem using abstraction?**

**Solution:**

Abstraction is selecting data from a larger pool to show only the relevant details of the object to the user. Abstraction “shows” only the essential attributes and “hides” unnecessary information. It helps to reduce programming complexity and effort. It is one of the most important concepts of OOPs.

**7.(a) What are objects in C ++?**

An object is an abstract data type with the addition of polymorphism and inheritance. Rather than structure programs as code and data, an object-oriented system integrates the two using the concept of an "object". An object has state (data) and behavior (code). Objects can correspond to things found in the real world

**(b) How is a member function of a class ?**

Member functions are operators and functions that are declared as members of a class. Member functions do not include operators and functions declared with the friend specifier. These are called friends of a class. ... The definition of a member function is within the scope of its enclosing class

**8.(a) What is linear data structure ?**

A **Linear data structure** have **data** elements arranged in sequential manner and each member element is connected to its previous and next element. Such **data structures** are easy to implement as computer memory is also sequential. ... Examples of **linear data structures** are List, Queue, Stack, Array etc.

(b) Why we use stack ? Difference between array and linked list ?

(c) What is merge sort and how it works ?

**Solution:**

Merge sort is a sorting technique based on divide and conquer technique. With worst-case time complexity being  $O(n \log n)$ , it is one of the most respected algorithms.

Merge sort first divides the array into equal halves and then combines them in a sorted manner.

How Merge Sort Works?

To understand merge sort, we take an unsorted array as the following –



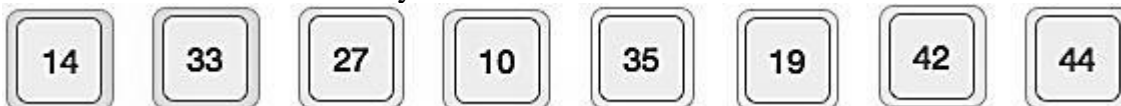
We know that merge sort first divides the whole array iteratively into equal halves unless the atomic values are achieved. We see here that an array of 8 items is divided into two arrays of size 4.



This does not change the sequence of appearance of items in the original. Now we divide these two arrays into halves.



We further divide these arrays and we achieve atomic value which can no more be divided.



Now, we combine them in exactly the same manner as they were broken down. Please note the color codes given to these lists.

We first compare the element for each list and then combine them into another list in a sorted manner. We see that 14 and 33 are in sorted positions. We compare 27 and 10 and in the target list of 2 values we put 10 first, followed by 27. We change the order of 19 and 35 whereas 42 and 44 are placed sequentially.



In the next iteration of the combining phase, we compare lists of two data values, and merge them into a list of found data values placing all in a sorted order.



After the final merging, the list should look like this –



Now we should learn some programming aspects of merge sorting.

Algorithm



---

Merge sort keeps on dividing the list into equal halves until it can no more be divided. By definition, if it is only one element in the list, it is sorted. Then, merge sort combines the smaller sorted lists keeping the new list sorted too.

**Step 1** – if it is only one element in the list it is already sorted, return.

**Step 2** – divide the list recursively into two halves until it can no more be divided.

**Step 3** – merge the smaller lists into new list in sorted order.